Microprocessors and Microsystems 37 (2013) 319–332

Contents lists available at SciVerse ScienceDirect

Microprocessors and Microsystems

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

Enabling accurate modeling of power and energy consumption in an ARM-based System-on-Chip

Jose Nunez-Yanez [a,*], Geza Lore [b]

[a] University of Bristol, United Kingdom
[b] ARM Ltd., United Kingdom

ARTICLE INFO

Article history:
Available online 11 January 2013

Keywords:
Energy modeling
Power estimation
System-level modeling

ABSTRACT

Motivated by the importance of energy consumption in mobile electronics this work describes a methodology developed at ARM for power modeling and energy estimation in complex System-on-Chips (SoCs). The approach is based on developing statistical power models for the system components using regression analysis and extends previous work that has mainly focused on microprocessor cores. The power models are derived from post-layout power-estimation data, after exploring the high-level activity space of each component. The models are then used to conduct an energy analysis based on realistic use cases including web browser benchmarks and multimedia algorithms running on a dual-core processor under Linux. The obtained results show the effects of different hardware configurations on power and energy for a given application and that system level energy consumption analysis can help the design team to make informed architectural trade-offs during the design process.

1. Introduction

Energy efficiency is one of the primary design constraints for mobile devices that need to operate autonomously for as long as possible [1]. The current rate of battery life improvement of around 5% per year [2] means that the limited energy budget could delay the introduction of the future chips needed to support workloads whose complexity increases by one order of magnitude every 5 years [3]. Additionally, minimizing power consumption reduces the amount of heat dissipated requiring lower cost packaging, cooling solutions and increasing device reliability. Available studies show that a 10° increase in working temperature causes a 100% increase in failure rate [4].

The first action required to achieve the objective of minimizing energy or power consumption is to understand how the available energy budget is being used in the system [5]. This means that not only the main components such as the processor cores should be considered, but the whole system should be analyzed using real workloads as inputs to account for the dependency of power on the dynamic behavior of applications. These analysis results should be available before the device has been fabricated and should be accurate enough to guide the design team in the process of making architectural decisions leading to a solution superior in power and energy terms.

Fig 1 depicts the typical main components that can be found in a modern mobile phone. This paper focuses on the interaction of the application processor and the memory subsystem. These parts account for roughly 30–50% of the total device power budget in compute intensive applications such as media playing [5]. The power consumption of the LCD/backlight combination and radios are the two other main component drains in the system but they are considered to be beyond the scope of this work. With these constrains in mind the present work contributions are as follows:

1. The introduction of a power modeling methodology based on implementation data and regression analysis at the system level. The methodology enables the creation of accurate models that can then be stimulated from only sparse trace information collected during long application runs.
2. The analysis of power and energy consumption trends in a state-of-the-art multiprocessor architecture with realistic benchmarks including internet browsing and video coding running on a Linux operating system.

The rest of the paper is organized as follows. Section 2 positions this work in relation to existing work in the field of power modeling. Section 3 presents the power modeling methodology in detail. Section 4 presents the target system investigated in this work centered around a dual-core processor. Section 5 summarizes the power models developed for the different components. Section 6 showcases the application of the methodology to the estimations of power and energy consumption for a set of benchmarks and real

* Corresponding author.
E-mail address: j.l.nunez-yanez@bristol.ac.uk (J. Nunez-Yanez).

0141-9331/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.micpro.2012.12.004

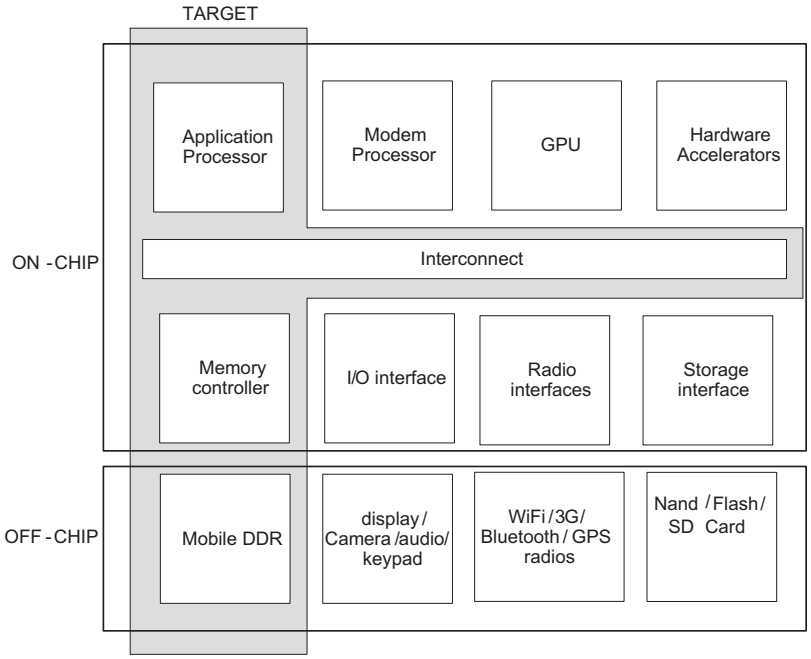

Fig. 1. High-level view of a mobile phone architecture.

applications. Finally Section 7 concludes this paper indicating its capabilities, limitations and stating future work.

2. Related work

This section makes a classification between research that considers power modeling at individual component level and system level which is the approach followed in this work.

2.1. Power modeling at the component level

This section reviews power modeling that considers independent components like the microprocessor core.

2.1.1. Microprocessor core

The component that has received most attention is naturally the CPU [6]. The standard approach consists of using existing cycle-level architectural simulators such as SimpleScalar [7] extended with tightly coupled power estimation capabilities. The power models added to the simulators are based on either analytical or empirical techniques [8]. Analytical techniques are useful for regular structures such as RAM-based structures (cache, register files, and buffers). These analytical models suffer from inaccuracies since they cannot properly capture node capacitance that depends heavily on design layout. They are complemented with empirical models based on power analysis of structures expected to be reused, with data extracted from recently designed processors. This methodology is used in power modeling research tools such as Wattch [9], SimplePower [10], and PowerTimer [11].

Wattch [9] is a power-performance simulator widely used within the academic community. The base performance simulator is SimpleScalar and can be used to investigate the effects of cache organization, pipelining, multi-instruction issue, etc. on power. The energy models used in Wattch are based on scaled power numbers obtained from published values or analytical equations for the regular structures. SimplePower [10] tries to improve accuracy by capturing power variations due to switching-activity in the processor logic blocks. Detailed simulation-based circuit energy is obtained for possible cycle-to-cycle transitions of the input pins of the different subunits and the values are stored in look-up tables. A similar approach is followed in PowerTimer [11] that includes a hierarchical suite of energy functions that are refined as the design and simulation model evolves. This allows a progressive improvement of the estimation accuracy when the circuit data becomes available.

These solutions share the common characteristic that they are focused on the processor and that the methodology is based on obtaining power data for each of the individual microarchitectural components present in the microprocessor. The power data is stored in look up tables in which a different entry exists for each possible input transition. These tables can grow very large and Wattch uses a simple fixed-activity model. Wattch only tracks the number of accesses to a specific component and utilizes an average capacity value to estimate the power consumed. The number of functional units considered in the processor can become very large as shown in [2] that models the power for a single 8-bit carry-select adder decomposed into its individual constituent gates. This approach is expected to be accurate for individual components but scaling it to full processors is complicated, slows down the simulation speed considerably due to the number of components that is necessary to trace and has limited accuracy because, for example, it neglects the layout of these units in relation with the other units and this can introduce errors. The advantage of this fine level decomposition is that it allows the study of the power effects of replacing subunits, for example, using a different addition technique. In contrast, our processor model is simpler although it is tuned to the Cortex-A9 processor [12] considered in this work and a different model should be developed if a different processor will be used. In essence our approach focuses on system architectural changes rather than microarchitectural changes in the processor.

Another challenge also present in the above described techniques is that the low level switching activity needed to address the look up tables will not be generally available in simulators such as SimpleScalar that make a number of simplifications to make the simulation numerically efficient [6]. The speed efficiency of simulators such as SimpleScalar is obtained by abstracting away the