

System-Level Power Estimation and Optimization — Challenges and Perspectives

Jan M. Rabaey

Department of EECS, University of California at Berkeley

Abstract

Energy considerations are at the heart of important paradigm shifts in next-generation designs, especially in systems-on-a-chip era. Voltage might very well become a variable design parameter. Hybrid architectures mixing a variety of computational models are bound to be integrated on a single die. Exploiting the opportunities offered by these architectural innovations requires a well-thought out design methodology, combining high-level prediction and analysis tools with partitioning, optimization and mapping techniques. The paper presents a plausible composition of such a design environment.

1. Introduction

Systems-on-a-chip are becoming a reality even today, combining a wide range of complex functionalities on a single die. Integrated circuits that merge core processors, DSPs, embedded memory, and custom modules have been reported by a number of companies. It is by no means a wild projection to assume that a future generation design will combine all the functionality of a mobile multimedia terminal, including the traditional computational functions and operating system, the extensions for full multimedia support including graphics, video and high quality audio, and wired and wireless communication support. In short, such a design will mix a wide variety of architecture and circuit styles, ranging from RF and analog to high-performance and custom digital.

Such an integration complexity may seem daunting to a designer and might make all our nightmares regarding performance, timing and power come true. On the other hand, the high level of integration combined with its myriad of design choices might be a blessing as well and can effectively help us to address some of the most compelling energy or power-dissipation problems facing us today. Exploiting these opportunities requires a system-oriented design methodology, which is certainly not in place today.

The purpose of this paper is to highlight some of the opportunities and challenges offered by system-on-a-chip integration levels, and to outline a design methodology that takes energy into account from the early phases of the design process.

Permission to make digital/hard copy of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

©1997 ACM 0-89791-903-3/97/08..\$3.50

2. Opportunities for energy minimization

Most of the literature of the last decade has focused on power dissipation, it is really minimization of the energy dissipation in the presence of performance constraints that we are interested in. For real-time fixed-rate applications such as DSP, energy and power metrics are freely interchangeable as the rate is a fixed design constraint. In multi-task computation, on the other hand, both energy and energy-delay metrics are meaningful depending upon the prime constraints of the intended design. In the remainder of the text, we will focus mainly on the energy metric, although energy-delay minimization is often considered as well.

Assuming that energy consumption will be dominated by capacitive switching for some time to come, the parameters the designer can work to reduce the energy budget include switching capacitance and supply and signal voltages. While traditionally the latter were fixed over a complete design, it is fair to state that in future systems-on-a-chip voltage can be considered as a parameter that can vary depending upon the location on the die and dynamically over time. This has been explored by a number of researchers over recent years and the potential benefits of varying supply voltages are too large to be ignored. This is illustrated in Fig. 1., which plots the normalized energy consumption of a computational task

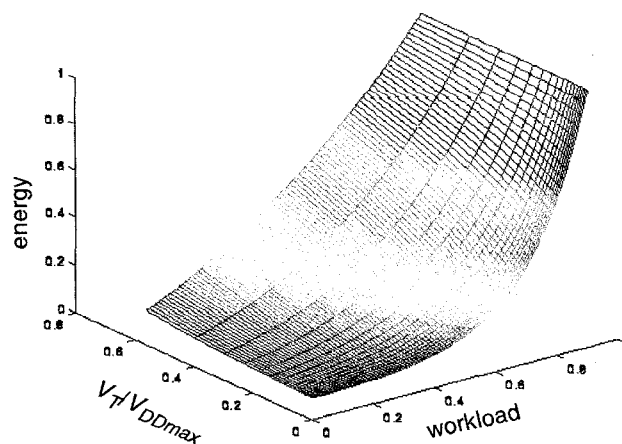


Fig. 1. Energy dissipation as a function of workload and ratio between threshold voltage and maximum supply voltage.

as a function of the workload ratio and of V_T/V_{DDmax} . A process is defined to have a workload of 1 if it consumes all the computational power of a processor running at its maximum supply voltage. Under-utilizing a hardware resource without scaling the supply voltage yields only a linear improvement in energy consumption, provided that the resource does not consume any energy when not in use. On the other hand, scaling the supply voltage in concert with the required workload (i.e. performance) produces a cubic reduction in dissipation! The graph further indicates that this cubic effect is only achieved if the design process maintains a reasonable ratio between V_{DDmax} and V_T , as voltage scaling becomes rather ineffective for small ratios.

Matching the desired supply voltage to a task can be accomplished in different ways. For a hardware module with a fixed functionality and performance requirement, the preferred voltage can be set statically (e.g. by choosing from a number of discrete voltages available on the die). Computational resources that are subject to varying computational requirements have to be enclosed in a dynamic voltage loop that regulates the voltage (and the clock) based on the dialed performance level [Chand96].

Based on the above discussion, we may assume that an energy minimization flow treats supply voltage as an open variable, to be fixed in concert with architectural optimizations for switching capacitance.

Reducing the latter typically comes down to a single, relatively obvious task: **avoid waste**. When analyzing many integrated circuit implementations, it becomes rapidly obvious that this is not that trivial: only a small fraction of the energy is typically spend on the real purpose of the design, i.e. computation. The rest is wasted in “overhead” functions such as clock distribution, instruction fetching and decoding, busing, caching, etc. Energy-efficient design should strive to make this overhead as small as possible. This is most easily achievable in ASIC implementations, where the hardware architecture can be directly matched to the application at hand. Yet, a large majority of the applications require programmability, which automatically translates in a lower energy efficiency. The goal of the architecture optimization process is then to keep this overhead to a minimum, which can be accomplished by sticking to a number of guidelines:

- Match architecture and computation to a maximum extent
- Preserve locality and regularity inherent in the algorithm
- Exploit signal statistics and data correlations
- Energy (and performance) should only be delivered on demand, i.e. an unused hardware module should consume no energy whatsoever.

Taking these considerations into account can lead to programmable architectures that consume dramatically less power than the traditional load-store engines. Reconfigurable architectures that program by restructuring the

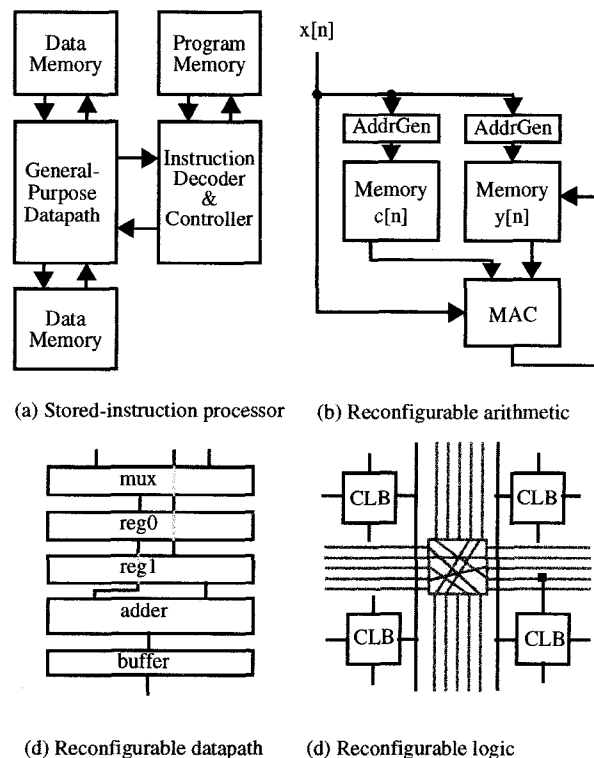


Fig. 2. Granularity of reconfiguration

interconnections between modules are especially attractive as they adhere to many of the above guidelines, although this is only true under the condition that the an adequate match is obtained between computational and architectural granularity. Fig. 2. enumerates a number of reconfiguration (or programming) approaches at different granularity levels, each of which tends to excel at one particular class of applications.

The impact of choosing the correct architecture for a given computational structure can be quite extreme and can change the energy consumption by orders of magnitude as illustrated in the examples of Table 1.

Table 1. Impact of architectural choice on energy dissipation for two important DSP functions.

Dot-vector product	ARM 6	Superscalar DSP	Reconfigurable Arithm.
	17 MIPS/W	266 MIPS/W	6 GIPS/W
CDMA correlator (per symbol)	ARM 6	Xilinx 4003	ASIC
	2765 nJ	394 nJ	1.2 nJ

It is rare that a complete application matches perfectly to a single computational paradigm. It is therefore to be expected that future systems-on-a-chip will combine various models on the same die in a hybrid configuration. While this multi-processor approach helps to boost performance, it is most probably energy minimization that will be the major trust behind this paradigm shift.

Hybrid architectures — while attractive — have the disadvantage to be more complex and harder to program than traditional general purpose processors. Overcoming this

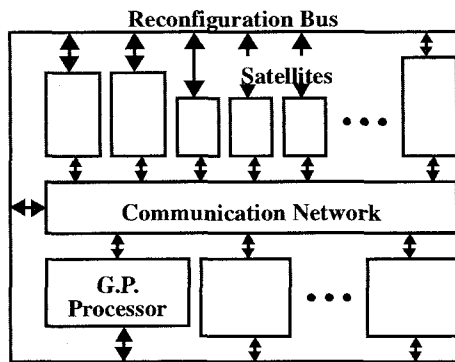


Fig. 3. Multi-granularity architecture template.

important disadvantage requires a structured approach that allows for maximum reuse of both hardware and software components. The Pleiades project [Pleiades97], currently under way at UC Berkeley, is an attempt to present such an approach. The reusable template of Fig. 3. can be used to implement a domain-specific processor instance, that can then be programmed to implement a variety of algorithms within a given domain of interest. All instances of the architecture template share a common set of control and communication primitives. The type and the number of processing elements may vary; they depend upon the properties and the computational requirements of the particular domain of interest.

The architecture is centered around a reconfigurable communication network. Connected to the network are an array of heterogeneous, autonomous processing elements, called satellite processors. These could fall into any of the reconfigurable classes: a general microprocessor core (most of the time only one of these is sufficient), a dedicated functional module such as a multiply-accumulator or a DCT unit, an embedded memory, a reconfigurable datapath, or an embedded PGA.

3. Power analysis and optimization methodology

Exploiting the opportunities offered by the architecture optimizations presented above requires a well-thought out design methodology and its accompanying tools. Fig. 4. plots the design flow to be followed when mapping a given application onto a hybrid architecture of the style of Fig. 3. Although the presented flow was developed for this particular architectural template, it is rather general in nature and is in fact applicable to a variety of hybrid hardware-software co-designs. The following phases can be discerned:

- classification of the various components of an application along computational properties and requirements
- trade-off analysis for the dominant kernels based on performance, energy and area estimates over implementation alternatives. Observe that it requires improvements in the range of orders of magnitude before it makes sense to move a kernel from the general purpose architectures to a more dedicated accelerator.

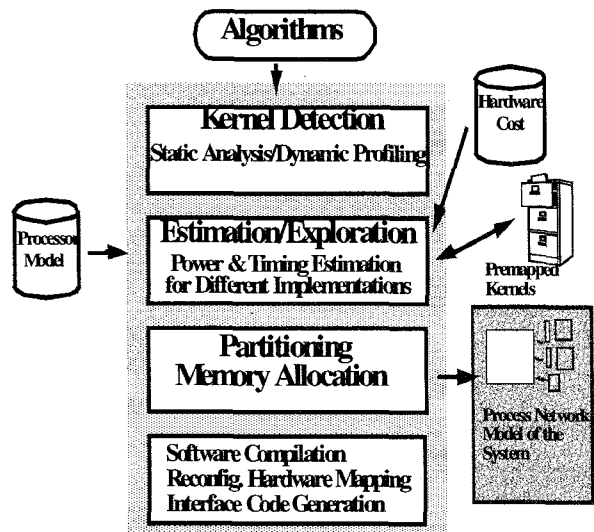


Fig. 4. Proposed design flow for low-power hybrid architectures

- design partitioning
- mapping the kernels onto the chosen architecture model and generation of the communication and reconfiguration interfaces.

The enabling of such a design methodology requires a clear understanding of the underlying concepts and semantics of the various architectural choices, and the interrelationship between the architectural and algorithmic properties. Only when such is achieved will we be able to produce modeling techniques that will help to produce the fast and accurate estimations of performance, energy consumption and cost that are a crucial element to the success of any design environment for hybrid architectures. The conference presentation will present an overview of the existing modeling approaches, addressing both their strengths and their weaknesses.

4. Summary

Systems-on-a-chip will likely be composed of a collection of hybrid architectural elements ranging from microprocessors, DSPs, vector processors, specialized accelerators, ASIC components and programmable logic. The correct partitioning of the applications over these components can have a dramatic impact on the energy dissipation of the design. Design methodologies that enable an intelligent trade-off analysis are essential in that perspective.

5. References

[Chand96] A. Chandrakasan, "Data-driven signal processing: An approach for Energy Efficient Computation", Proc. ISLPED 96, pp. 347-352, Monterey.
 [Pleiades97] The Pleiades Project Home Page, <http://infopad.eecs.berkeley.edu/research/reconfigurable>