INTEGRATION, the VLSI journal xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

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

# Simplifying low-power SoC top-down design using the system-level abstraction and the increased automation

Dominik Macko*, Katarína Jelemenská, Pavel Čičák

*Slovak University of Technology in Bratislava, Ilkovičova 2, 84216, Bratislava, Slovakia*

ARTICLE INFO

*Keywords:*
High-level synthesis
Low power design
Power management
System-level specification
Verification

ABSTRACT

Since power is the key aspect in modern systems on chips, many power-reduction techniques are adopted in the design process, mostly applied through power management. Its standard specification lacks the abstraction required by complex designs and therefore becomes difficult and error-prone. In this work, higher abstraction is introduced into the power-management specification and it is integrated with the functional model of the system. It simplifies the specification approximately 16.8 times and enables the automatic generation and verification of the equivalent standard specification. The error-prone nature of the power-management specification is thus alleviated and the difficult verification process is relieved.

## 1. Introduction

The power consumption is a great concern for hardware systems developers mainly due to increasing power density in systems on chips (SoC). Therefore, power must be dealt-with during the whole design process, usually by utilizing a so-called power management. However, it complicates the already too complex design process even more, and therefore abstraction and automation must be used to cope with the complexity (also to increase the productivity). The electronic system level (ESL) abstraction slowly becomes the design starting point in the industry and several methods have been developed (e.g. Refs. [1,2]) to also raise the abstraction level for adoption of power management into the design. However, they are either too dependant on design reuse, use too much lower level details for specification, or do not provide sufficient automation.

This paper presents a new low power systems design methodology, which eliminates weaknesses and builds up on the strengths of the existing methods. It integrates the power-management specification directly into the functional model of the system at the ESL (in SystemC) and uses high-level synthesis to automatically obtain the standard power-managed register-transfer level (RTL) model of the same system. The simplified abstract specification makes the designer's input more efficient (approximately 16.8 times) and the automatically synthesized RTL model enables more accurate design analysis. This methodology is enhanced by automated verification processes, which drive a designer to the correct and complete specification.

This paper is organized as follows. The next section (Section 2) provides a deeper background and motivation for our research. In Section 3, the state-of-the-art in the area of ESL-based power management is described. Section 4 provides an overview of the proposed low-power SoC design methodology along with a description of the utilized new methods of abstract power-management specification and power-management high-level synthesis. Before the conclusion, the experimental results (Section 5), illustrating the usefulness of the methodology, and comparison to related works (Section 6) are presented.

## 2. Background

In order to reduce power consumption, one must understand what factors influence the power. The total power in CMOS (Complementary Metal-Oxide Semiconductor) technology is a function of switching activity, capacitance, voltage, and transistor physical properties [3]. Formally, it is expressed by

$$P = P_{SW} + P_{SC} + P_L \tag{1}$$

where $P$ is the total power, $P_{SW}$ is the switching power, $P_{SC}$ is the short-circuit power, and $P_L$ is the leakage power. The leakage power is also called the static power. The switching power together with the short-circuit power are referred to as the dynamic power ($P_D$). Components of the dynamic power are defined in the following equations.

$$P_D = P_{SW} + P_{SC} \tag{2}$$

$$P_{SW} = a.\, f.\, C_{eff}.\, V_{dd}^2 \tag{3}$$

$$P_{SC} = I_{SC}.\, V_{dd} \tag{4}$$

In these equations, $a$ represents the switching activity, $f$ is the switching

* Corresponding author.
*E-mail address:* dominik.macko@stuba.sk (D. Macko).

https://doi.org/10.1016/j.vlsi.2018.06.001
Received 17 January 2018; Received in revised form 28 May 2018; Accepted 4 June 2018
0167-9260/ © 2018 Elsevier B.V. All rights reserved.

frequency, $C_{eff}$ is the effective capacitance, $V_{dd}$ is the supply voltage, and $I_{SC}$ is the short-circuit current. Therefore, the dynamic power can be lowered by reducing switching activity and clock frequency (affecting performance), or by reducing capacitance and supply voltage. Leakage power is a function of the supply voltage, the switching threshold voltage, and the transistor size. It is dissipated continuously, because of the leakage current, and thus design techniques (such as enabling of powering-down the circuit when not used) must be used to reduce it [3]. All of these factors are substantially used to reduce power in modern SoCs. Based on which power-affecting factor is targeted, various power-reduction techniques have been developed. The following subsection summarizes the most popular techniques and their standard application in the design.

### 2.1. Power-reduction techniques application

The existing power-reduction techniques can be divided into three categories:

- *Circuit-optimization techniques* – This category contains techniques that change physical parameters of the designed circuit (e.g. structure and size), such as logic restructuring, transistor resizing, pin swapping, or multiple supply voltages.
- *Power-management techniques* – These techniques utilize a dynamic power management. It enables the device to temporarily switch the operating mode in order to save the energy. Some portions of the device can stop its operation, isolate the spreading of the signals, adjust the voltage or frequency, or even can be powered down. These techniques include clock gating, operand isolation, substrate biasing, voltage and frequency scaling, and power gating.
- *Architectural techniques* – This category is a hybrid one containing rather the architectural choices enabling other power-reduction techniques to be applied. These techniques include, for example, memory or bus segmentation and hardware acceleration.

Since the system power highly depends on the used implementation technology, it is coupled with the physical level of the design. However, hardware designs are too complex at such a low level, and therefore the adoption of advanced power-reduction techniques (working with multiple voltages) would be very difficult and error-prone. It would require full-chip functional verification, which would be unbearable (in terms of time). In order to deal with design complexity and to reduce the number of design re-spins the IEEE standard for design and verification of low-power integrated circuits [4] has been developed (commonly known as UPF – Unified Power Format). There is another widely used standard, known as CPF (Common Power Format) [5], which has similar capabilities to the UPF. However, since these standards are unifying in new versions of UPF, we focus only on this one. The UPF has offered a clear design flow utilizing the power-management techniques and enabled RTL (Register-Transfer Level) power-aware verification. It contains the constructs for specification of low-level power-related aspects during the design stage, when mostly the HDL (Hardware Description Language) modelling is used. In such a way, the whole design (the functional HDL model along with the UPF power management) can be verified.

The key UPF concept is to provide the means for dividing the system into power domains. The power domain is a collection of design blocks that always operate at the same supply voltage level. UPF enables the designer to specify which blocks are grouped into the power domain, what voltage levels the power domain can operate at, what the power-down condition for each power domain is, where the isolation cells and level shifters should be used, and so on.

However, the use of this standard is still rather complicated (error-prone and time-consuming) especially in modern SoC designs that are becoming more and more complex. To reflect the current trend of adopting more-abstract level above the RTL [6], the so-called electronic system level (ESL), the UPF has been updated to the version 3.0 [7]. It enables to specify power intent and verify it in context of IP blocks (Intellectual Properties) in the ESL model. It standardizes the power model, in which the designer can specify power consumption of an IP in various power states. The IP power characterization way is, however, outside the scope of this standard. For accurate power analysis, these data can be obtained from the previous implementation of the IP block, and therefore such a method is highly dependent on the design-reuse concept. Although an algorithmic model can be synthesized into IP block based model (with known power characterization) and UPF 3.0 then used for power management, it complicates the top-down design process and limits flexibility to only pre-designed IP blocks. Moreover, the specification of power management (e.g. power-supply networks, power-management elements, or power switching) does not correspond to this abstraction level (i.e. it uses too low-level details). The next subsection introduces the ESL-based design and problems of integrating UPF in such a design process.

### 2.2. System-level design

In the top-down design process utilizing the ESL, the order of abstraction levels is illustrated in Fig. 1. At the highest abstraction level, the most widely used modelling approaches are based on C or C++ languages for system description (SystemC [8] is the most popular along with its TLM extension – Transaction Level Modelling), because they can be used for both algorithmic and architectural modelling. The top-down transitions between the abstraction levels are achieved through the synthesis processes – high-level synthesis, logic synthesis, and place and route processes. The development process is nowadays usually accelerated by the use of various EDA (Electronic Design Automation) tools for synthesis or verification. The RTL and lower levels, along with the transitions between these levels, are well-supported by such tools in all aspects. Regarding the ESL, several EDA tools enable the system architecture definition in the SystemC/TLM form, or enable capturing the functionality in the C/C++/SystemC algorithmic manner. Some of them support even the high-level synthesis. The best-known of these tools are summarized in Table 1. These tools promote the easier use of the ESL in the design process. At first, the ESL was adopted in the industry mainly for verification purpose (e.g. virtual prototyping). Higher abstraction enables faster verification of the functionality. The verified ESL model further serves as a golden model for equivalence checking with the more complex RTL implementation model. However, the advances in high-level synthesis in the modern EDA tools enable the adoption of ESL as an implementation starting point. It means that the RTL model is automatically generated from the ESL model, according to some predefined constraints. It reduces the number of human errors in the design and shortens the verification time. Moreover, the automated high-level synthesis enables to get the results of more accurate design analysis at the RTL sooner, and thus it enables to find the right trade-off among multiple parameters (power, performance, area) much faster.

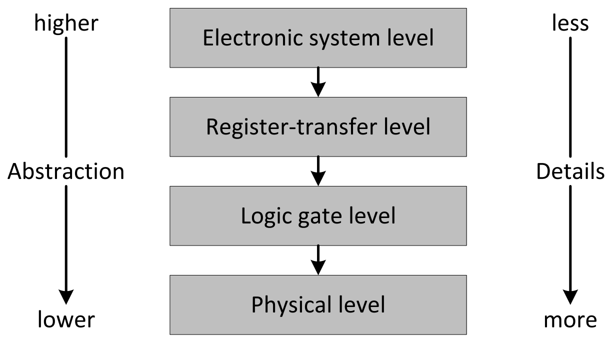

**Fig. 1.** Design process abstraction levels.