Contribution to System-Level Design and Verification of Low-Power Digital Systems

Dominik Macko*
Institute of Computer Systems and Networks
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
dominik.macko@stuba.sk

Abstract

Power consumption is becoming the key aspect in modern digital systems design. Widely accepted method for applying power-reduction techniques is the adoption of a power-management strategy. The standard-based low-power design flow involves the application of the power-management techniques in a register-transfer level model. At this stage, the current systems are usually too complex, and thus the application of the power management is very complicated. Moreover, a verification of such an invasion into the system design takes too much time.

The dissertation thesis deals with a novel methodology for low-power systems design, based on an extension of the standard design flow to the system level of abstraction. The proposed extension utilizes high-level synthesis for automated transformation of the system specification into the widely-used model at the register-transfer level. During this process, the abstract power-management specification is transformed into the more-detailed standardized form in order to be supported by the existing tools for verification and analysis. The automation of this process prevents many human errors usually introduced by the manual specification. Besides, this process involves several verification steps in order to identify and correct some of the errors in the abstract specification. The experimental results showed that the proposed abstract power-management specification is approximately 19 times less complex than the standard specification at the register-transfer level. The resulted methodology makes the low-power systems design easier, prevents many human errors in the design, and simplifies the verification process.

*Recommended by thesis supervisor: Prof. Pavel Čičák. Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on August 21, 2015.

Macko, D. Contribution to System-Level Design and Verification of Low-Power Digital Systems. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 7, No. 2 (2015) 10-17

Categories and Subject Descriptors

B.5.1 [**Register-Transfer-Level Implementation**]: Design—*styles*; B.5.2 [**Register-Transfer-Level Implementation**]: Design Aids—*automatic synthesis, optimization, verification*; B.7.1 [**Integrated Circuits**]: Types and Design Styles; B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids

Keywords

digital system, design automation, design verification, high-level synthesis, low power, power analysis, power management, power reduction

1. Introduction

The power consumption has been a great concern for the mobile devices manufacturers for a long time. Recently, the finer process technologies (below 90 nm) along with the market forces make power the key factor constraining any electronic design. The increasing demand for high-performance portable SoCs (Systems on Chips) in communication and computing has shifted the focus from traditional constraints (area, performance, cost, and reliability) to the power consumption [3].

Since at smaller geometries the power density is rising, it has a negative impact on the temperature and implied reliability of the device (more faults - shorter lifetime). Therefore, even for the industry segments in which battery power has not traditionally been an issue, considerations of packaging, reliability, and cooling costs brings power to the forefront [1].

Many techniques have been developed to address the continuously aggressive power reduction requirements (see Table 1)[5]. Some of them were successfully integrated to the design flow (e.g. clock gating), but others were hard to adopt. To help the design teams to adopt advanced power-reduction techniques (e.g. multiple supply voltages), the low power standard for design and verification was developed [6]. This standard enables specification of the power-management low-level details early in the design flow (at the register-transfer level - RTL), thus making the verification of the power management more efficient. As the systems become more complicated, this abstraction level can no longer serve as the starting point.

Power management has to be considered from the start of the development process, namely at the system level. The right power-management strategy at the system level has

Table 1: Overview of Commonly Used Power-Reduction Techniques

Technique	Description
Multiple threshold voltages	Uses different threshold voltages in the circuit to reduce leakage but still satisfy timing constraints.
Multiple supply voltages	Different blocks are operated at different (fixed) supply voltages. Signals that cross voltage domain boundaries have to be level-shifted.
Gate sizing	Upsizing reduces dynamic power, downsizing reduces leakage power.
Logic restructuring	Moves high switching logic to the front and low switching logic to the back.
Clock gating	Disables clock tree part not in use.
Operand isolation	Prevents switching of inactive datapath element.
Voltage scaling	Different blocks are operated at variable supply voltages. The block voltage is dynamically adjusted based on performance requirements.
Frequency scaling	Frequency of the block is dynamically adjusted. Works alongside with voltage scaling.
Power gating	Turns off supply voltage to blocks not in use, significantly reducing the leakage. Block outputs float and need to be isolated when connected to active block.
Substrate biasing	Dynamically bias the substrate or the appropriate well in order to raise transistor voltage threshold in inactive mode, thereby reduce leakage.
Memory partitioning	The memory is split into several partitions. Not-used ones can be powered down.
Bus segmentation	The system bus is split into several segments. Only the required segments are charged upon an access.
Hardware acceleration	Uses dedicated elements to speed-up a task. These elements can be powered down when not needed.

the most beneficial impact on eventual power consumption. The used power-reduction techniques must be considered with the rest of the system requirements, because they impact all aspects of ASIC (Application-Specific Integrated Circuit) and SoC development [3]. Early verification can significantly improve productivity and find power-related design errors early when it is easier and cheaper to resolve them.

The dissertation is oriented towards the extension of the current low-power design flow in order to utilize the benefits of the system level (e.g. concise specification, faster verification) and the high-level synthesis process (e.g. faster implementation, more-accurate design analysis). The goal is to develop a new method for abstract power-management specification that would simplify the introduction of selected power-reduction techniques into the design. The next goal is to develop a new synthesis algorithm usable in the high-level synthesis process that would transform the proposed abstract power-management specification into a more-detailed standard-based specification. This transformation into the standardized form assures compatibility with existing design-automation tools that could be used for the design analysis and verification at the later design stages. The last goal is to utilize a combination of verification approaches to properly verify introduced power-management aspects with as little manual intervention to the design as possible.

This paper is organised as follows. The background information in Section 2 is followed by Section 3 that refers to the related work representing the state-of-the-art of the key problem area of this paper. Section 4 summarizes the goals of the dissertation thesis. In Section 5, the novel low-power design flow is proposed along with an introduction of the related proposed methods. Section 6 summarizes the experimental results provided in the dissertation thesis. The main achieved contributions are summarized in Section 7.

2. Design for Low Power

Power is the most critical issue in SoC design today [5]. In order the SoC power management to be efficient, it must be built into the design starting at the architecture stage and power-reduction techniques must be used at every stage of the design. Managing power becomes complicated, since there is a need to design for low-power as well as for performance and cost.

2.1 Power Basics

In CMOS (Complementary Metal-Oxide Semiconductor) technology, the power consists of two elements, the static and dynamic power [5]. The static power, also known as leakage power (P_L), is a function of the supply voltage (V_{dd}), the switching threshold voltage (V_{th}), and the transistor size (W/L).

$$P_L = F(V_{dd}, V_{th}, W/L) \tag{1}$$

The dynamic power is a sum of switching power (P_{SW} - dissipated when charging or discharging internal and net capacitances) and short-circuit power (P_{SC} - dissipated by short-circuit current during switching of the transistors). Formally, it is expressed by the following equation:

$$P_{SW} + P_{SC} = \alpha.f.C_{eff}.V_{dd}^2 + I_{SC}.V_{dd}.f, \tag{2}$$

where α is the switching activity, f is the switching frequency, C_{eff} is the effective capacitance, and I_{SC} is the short-circuit current. The total power is then formally represented by

$$P = P_{SW} + P_{SC} + P_L. \tag{3}$$

The dynamic power can be reduced by a reduction of switching activity, clock frequency, capacitance, and supply voltage. The leakage power is continuous due to the leakage current and therefore must be reduced by design techniques [5], such as those described in Table 1.

2.2 Power Management

The key UPF (Unified Power Format) standard [6] concept is to provide the means for dividing a system into

so-called power domains. A power domain is a collection of blocks that always operate at the same supply-voltage level (the same power state). It enables to power-down some unused power domains, while others are normally operating, or temporarily operating at different voltage level. This standard enables a designer to specify which blocks are grouped into a power domain, what voltage levels a power domain can operate at, what power-down condition for each power domain is, where isolation or level shifters should be used, and so on.

UPF assumes the control signals for power-management elements (e.g. power switches, isolation and retention cells) are generated in the functional design. For complex systems, this power-management logic is grouped into a separated system block, known as the power-management unit (PMU). It is responsible for determination of a suitable system power mode (a combination of power states of power domains) and handles the transition to that power mode. In Figure 1, the use of a PMU in a system architecture design is shown. In the figure, the *PMD* represents a power-mode determination module (usually consists of several monitors and a power-management policy algorithm), the *PSMs* stand for power-state machines handling the transitions between power states of individual power domains (the *PDs*), and the *Blocks* represent the other system blocks, such as a microprocessor, memory, and other IPs (Intellectual Properties).

Complete and accurate verification of a low-power design is indispensable in modern SoC design flows. For impact to be maximal, power-related decisions must be made at the architectural level, and the effects verified at each stage of the design [9].

After the formal specification of a power intent in UPF, the first verification step is to check functional, electrical, and structural correctness and completeness - this is usually done through formal verification. The next step is to verify the correct functionality of the system with low-power behaviour on top of normal function - usually achieved through power-aware simulation [5]. Additional power-management elements need to be checked whether they do not affect basic operation of the device. The design becomes more complicated when using multiple voltage domains or scaling voltage and frequency levels. Side-effects of the advanced power-reduction techniques are often the source of errors, and therefore the whole design must be verified for all specified operating modes [9]. Thus, complete power-aware verification should be hybrid, combining formal and informal approaches. Errors in the control signals (driven by the PMU) are detected

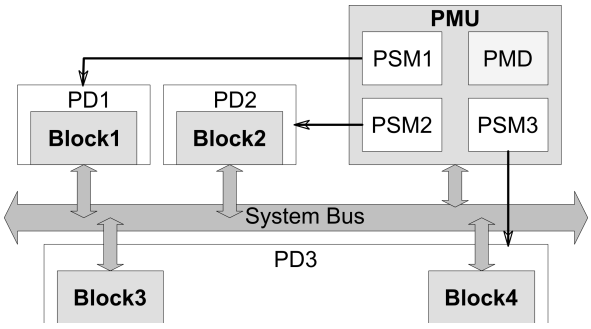

Figure 1: System architecture with PMU.

using low-power assertions. How well the power intent has been functionally verified is usually measured by using functional coverage in simulation. Formal tools are used for static analysis, validating the power intent, and for syntactical, structural, and functional checks throughout the low-power design flow [5].

2.3 System-Level Design

Limitations of power analysis at higher abstraction levels imply the fact that most designers consider power requirements later in the design flow - after synthesis or place-and-route process. However, the possibilities to reduce power at such a later stage are quite modest compared to the architectural context of the system abstraction level - early stage of the design flow. The tasks and processes made at an early stage (e.g. hardware/software partitioning, bus implementation, memory management, or hardware acceleration) have the greatest impact on eventual system power consumption [16]. The international technology roadmap for semiconductors [7] predicted few years ago that 80% of the power minimization would happen at the electronic system level (ESL).

Modern tools provide adequate modelling possibilities at the ESL, and thus, considering the increased simulation performance, enable the early design exploration. These tools usually depend on the design reuse that allows them to map the performance and power information from lower-level representations into the transaction-level (TLM) models. Synthesizable RTL descriptions are available for most parts of any new design, and the remaining parts can often be synthesized from their TLM representations (high-level synthesis). Power information obtained from the newly-synthesised blocks can be back-annotated into the TLM model to continue architecture exploration [16].

3. Related Work

The research area of ESL power management has been gaining attention for the past few years. Some works has already been published to extend UPF up to the system level. PwARCH [15] enables to explore different power architectures at the transaction level, augmenting a SystemC/TLM model with abstract UPF concepts intended for simulation. The problem is that power information has to be annotated to the ESL model (manual effort). Although this method enables the selection of the most power-efficient power architecture, it does not take into account other important parameters, such as area or performance. Similarly, in [8], the authors augmented the system-level functional model with a power-intent model (based on UPF concepts) and power data model (obtained from the lower-level power estimations and technology libraries). The method enables to automatically generate a UPF specification from the model, and thus use the power intent in the existing EDA verification tools. Disadvantage of the method is that the amount of information needed to model power intent is approximately the same as in the UPF specification.

Similar research has been done in [17, 10, 4]. These approaches are also based on the system-level power modelling (with the power-management modelling support) that enables faster simulation, and thus enables different power architectures exploration. The main difference from the previous methods is that the used power-management modelling is not based on the UPF standard. The RTL implementation has to be designed manually

and since the approaches do not use the standard concepts at the ESL, the verification cannot easily check the equivalency (error prone).

Another research [2] focuses on high-level power estimation based on statistical regression power models. It also proposes the use of ESL simulation traces at lower levels for power estimation. The work is also focused on system-level power reduction, specifically on the clock-gating technique. The need for clock gating is manually specified directly in the C programming code in a form of macro. Based on this specification, the high-level synthesizer inserts clock-gating cells into the RTL model. The disadvantage of this approach is that the designer needs to identify where exactly the clock gating should be used. It substantially complicates such high-level specification and does not involve other efficient power-management techniques.

4. Objectives

Based on the identified pros and cons of the existing methods, the following goals and tasks have been stated in the dissertation thesis.

- Identification of power-reduction techniques applicable at the system level of abstraction, their evaluation and selection of potential techniques for integration.
- Development of a method for specification of the selected power-reduction techniques at the system level.
- Integration of the proposed specification method into the selected system-level model.
- Development of a new power-management synthesis algorithm for transformation of the abstract power-management specification to the standardized RTL form.
- Development of a power-aware hybrid verification method based on the existing verification techniques.
- Evaluation of the proposed methods by an implementation of the developed synthesis algorithm.

5. A Novel Low-Power Design Flow

The key idea of the novel methodology [11] is illustrated in Figure 2. The extension of the UPF standard low-power design flow to the ESL keeps the current design flow steps intact, enabling to use the traditional design/verification methods and tools at lower levels. This methodology starts from a crude system specification at the ESL, where the main UPF concepts are integrated into the system functional specification in an abstract form. The specification participates in the abstraction refinement process.

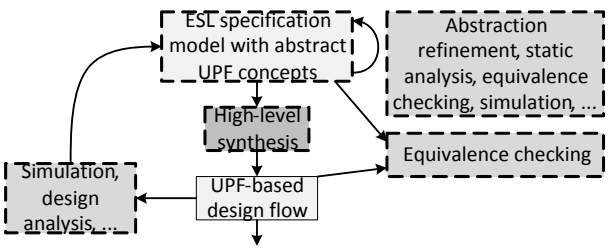

Figure 2: The low-power design flow extension.

When sufficiently refined, the high-level synthesis enables to automatically extract the specified power management and to generate its standard UPF representation along with the RTL functional model (typically in HDL). Then, the low-power design flow continues in a traditional way, as described in [6].

In the dissertation, the research was not focused on the functional high-level synthesis but on the high-level synthesis of the power-management specification. Each of the proposed methods utilized in the proposed methodology extending the UPF-based low-power design flow is briefly introduced in the following subsections.

5.1 Selection of Power-Reduction Techniques

The criteria for selection of power-reduction techniques have been defined as their suitability for the system-level of abstraction and their impact on eventual power consumption. Many techniques are highly coupled with an implementation technology and a circuit structure. Therefore, such techniques cannot be used at the system level. These techniques include multiple threshold voltages, gate sizing, and logic restructuring. Other techniques, such as memory partitioning, bus segmentation, and hardware acceleration, are rather architectural or microarchitectural choices. They are not specifically utilized in the proposed methodology but can be realized by the use of other techniques in a finer-grained manner (e.g. power gating). Substrate biasing is predicted to be replaced by power gating; therefore, it is not used in the proposed methodology. The other techniques, namely clock gating, operand isolation, multiple supply voltages, voltage and frequency scaling, and power gating, are utilized through power management at the system level in an abstract form.

5.2 Abstract Power-Management Specification

The proposed power-management specification abstracts from lower-level details, contained in UPF, such as power switches, isolation or retention cells, level shifters, or supply ports and nets. However, the basic concept of dividing a system into power domains has remained. The abstract power-management specification is based on these power domains, assignment of abstract power states to these domains (the states that can be reached by the blocks in the domain), assignment of instances to power domains (blocks that will belong to the domain), specification of allowed system power modes (allowed combinations of power states in power domains), and switching between the specified power modes.

The abstract power states represent performance levels (voltage-frequency pairs), in which the power domains and internal blocks can operate. These states include the following.

- *Normal* - the blocks operate at the basic supply voltage and at the basic operation frequency.
- *Hold* - the blocks stop their operation.
- *Diff_level#* - a set of states that are distinguished by an ordinal number #. The blocks operate at the performance level different from the basic one. Either voltage or frequency or both are different.
- *Off* - the blocks are powered down.
- *Off_ret* - the state of the blocks is retained, while they are powered down.

Each of these states represent some power-reduction techniques that will be used in the design. The *normal* state implies that no explicit power-reduction technique will be used while this state is active. *Hold* implies the architectural clock gating and operand isolation techniques will be applied during this state. *Diff_level* states enable multiple supply voltages to be used in the design and adoption of voltage and/or frequency scaling techniques. The *off* and *off_ret* states imply the power gating with or without state retention.

There is a dedicated system state variable representing the current power mode. This variable has to be firstly initialized to a default power mode, assigning some of the specified power modes to this variable. The switching between the individual power modes is modelled in the functional specification of the system by assigning to this state variable different power modes based on some system conditions.

5.3 Power-Management Integration

In order to put the abstract power-management specification into use, we have integrated it into two system-level specification models using two different integration techniques. The integration into HSSL (Hardware/Software Specification Language) model [11] was achieved by extending the language syntax. The integration into SystemC model [14] was achieved using the extension library.

The syntax extension offers a great opportunity for intuitive, rigorous and very concise specification. The existing constructs can be extended (or replaced) in order to support the power-management specification. Moreover, this kind of extension enables such a form of specification that is easily analysed (for a human being as well as for a computer). The greatest advantage of the extension library based integration is that it preserves the support in existing tools. When the language syntax is modified, the tools working with this language have to be updated. Another advantage is that this kind of extension is easily accomplished and the existing compilers can be used to perform static checks to reveal syntactical errors in the library. Designers are using the same modelling style as they are used to, when describing the system functionality in the given language. This can be also viewed as a disadvantage, because the specification has to follow the existing rules, and therefore it is more fractured (declaration in the declaration part, instantiation in the functional part). Moreover, several modelling options in the language have to be preserved (e.g. static and dynamic instantiation of the objects). This complicates the analysis of the power-management specification and provides an opportunity for a designer to introduce a human error to the specification.

5.4 Power-Management High-Level Synthesis

During the process of high-level synthesis, the power-intent is extracted from the abstract power-management specification, and the UPF standard specification is synthesized. The power-mode variable is changed to the enumerated type in order the functional high-level synthesizer to be able to synthesized it properly along with the functional specification. Beside the power intent in UPF, there is a functional description of the PMU generated, which drives the control signals of power-management elements specified in the UPF. Thus, the power-management synthesis consists of two processes - the power-intent specification synthesis and the power-management unit synthesis.

In order to perform the first synthesis process, we have proposed a synthesis algorithm that analyses abstract power management specification and then, based on the power management rules, it generates UPF that satisfies the abstract specification. However, in order to further simplify the low-power design, we have proposed another algorithm (algorithm with optimization) that can also automatically recover from some inconsistencies in abstract specification. It uses relations between power domains, given by the power modes and architectural dependencies of the system components, to determine whether the power-management specification is consistent with the functional specification. If it is not, it corrects the specification to be consistent (a designer is notified which parts have been modified).

For the PMU synthesis, we have used the information about the power intent, obtained from the UPF generation. Since all the required control signals for power-management elements are known in this method, control-signals sequences generation is pretty clear. These sequences are generated in such a way that power management rules are kept. For example, if some power domain is changing its state from *normal* to *off_ret*, the control signals must firstly isolate the domain boundary inputs and outputs, then the retention must be activated, and only afterwards the blocks in the domain can be powered down. These control sequences are achieved by generation of additional (intermediate) power modes of the system. The advantage is that this process is fully automated; thus, a designer does not need to keep these rules in mind and can focus on system functionality. Moreover, we have proposed a modification of the internal PMU architecture in order to be more power efficient [12]. It enables to power the PMU transition logic down, when a power-mode change is not needed. It is especially useful in a system that is "sleeping" (a power-saving mode) for a long time during its operation time.

5.5 Power-Management Verification

To verify the introduced power-management aspects, we have proposed a combination of formal and informal checks. These verification checks are intended to drive a designer to create a correct, complete, and consistent power management specification at the ESL. At such an early stage, these checks involve syntactic checks done by compilers, runtime checks that are part of the extension library, and the proposed static analysis - analysing the relations between power-management components and system architecture. After the high-level synthesis, we use the proposed equivalence-checking method to ensure that the power intent is preserved during this process. To verify the synthesized PMU, we generate assertions about control signals sequences. The assertions are also used for coverage measurement, driving a designer to verify the untested system functionality.

All of these verification steps help to reveal power-management errors at early stages of the design, when it is cheap and easy to correct them. The automation of these steps reduce verification overhead, which is very high in modern complex designs. Notifications, produced by the verification, help to identify the source of an error.

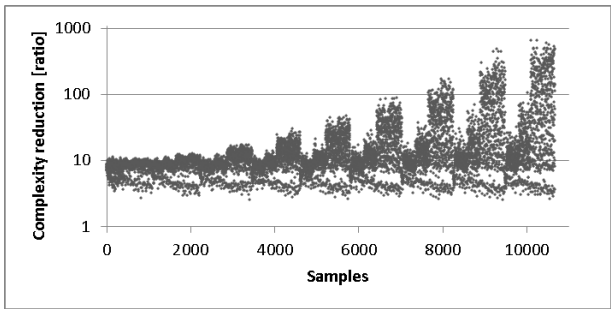

Figure 3: HSSL-integrated vs. synthesized UPF specification.

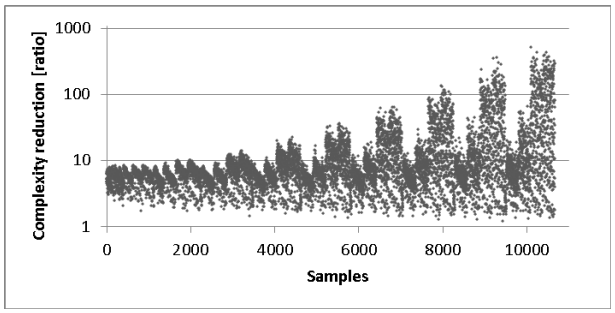

Figure 4: SystemC-integrated vs. synthesized UPF specification.

6. Experimental Results

For evaluation and validation of the proposed methodology, we have implemented the proposed methods into an experimental tool. We have generated over ten thousand samples of the ESL abstract power-management specification scaling the number of power domains, the number of power modes, the average number of power states in power domains, the average number of system blocks in power domains, and the number of inter-domain communications. The generated samples were described in both HSSL and SystemC extensions. The early verification steps ensured that the abstract specification was correctly generated and described. Afterwards, we have synthesized the ESL power-management specification using both proposed high-level synthesis algorithm (without and with optimization) and verified the synthesized UPF specification and PMU description.

6.1 Power-Management Simplification

To determine simplification of power-management specification, achieved using the proposed methodology, we have compared the abstract power-management specification to the standard UPF specification, generated by the synthesis process. As a comparison parameter, we have used the number of characters required for specification. Figure 3 and Figure 4 illustrate the results of this experiment. The results are provided in a logarithmic scale. The provided complexity reduction ratio represents the number of characters required for UPF specification divided by the number of characters required for ESL abstract power-management specification.

The results showed that the ESL power-management specification has approximately 19 times lower average complexity than the standard UPF specification with the same power intent. Simplified specification implies that a designer has fewer possibilities to introduce a human error into the specification. Thus, the specification and verification time is shortened. We have observed that the highest difference between ESL and UPF specifications was achieved in samples with higher number of abstract power modes. The reason is that a higher number of abstract power modes results in a higher number of intermediate power modes generated in UPF. When the two ESL power-management specifications (HSSL and SystemC) compared to each other, the result is that the HSSL-based specification is approximately by one third less complex than the SystemC-integrated one. Thus, the syntax-modification form of integration results in more concise specification than the extension library based integration, however, at a price of loosing compatibility with the existing EDA tools.

Table 2: Power-Management Validation Data

#	ESL	UPF	PMU	Assert	Coverage
1	313	1680	3300	3863	100 %
2	500	2706	4452	4749	100 %
3	642	2850	4619	3194	100 %
4	643	6035	35205	25912	98 %
5	751	4658	8658	9488	100 %
6	760	4854	7017	10340	100 %
7	813	5678	10094	13433	97.5 %
8	862	5557	63304	17779	100 %
9	953	8778	142992	52330	81.1 %
10	1051	9399	131478	45150	96.1 %
11	1090	11605	586303	100721	85.5 %
12	1275	7443	199866	48111	89 %
13	1324	9039	107068	43833	91.2 %
14	1402	13397	67691	50911	99.2 %
15	1939	12232	214122	91620	82.4 %

6.2 Synthesized Power-Management Validation

In order to validate the synthesized power management, we have selected 15 generated samples (various complexity) of abstract power management, synthesized them, and used industrial tool Modelsim SE 10.2c to verify the synthesized design. Using Modelsim's power-aware static analysis, we have validated the synthesized UPF specifications. No reported error has proven that the synthesized UPF specifications are syntactically correct and that no isolation or level-shifter strategy is missing in the specifications. To validate the PMU synthesis process, we have used Modelsim's simulation capabilities. It has successfully compiled and simulated all of the synthesized PMU samples. During this process, the generated assertions were used to check whether some control-signals sequence rule is violated. No reported error has proven that the control sequences are generated correctly. The generated assertions have also been successfully used to measure coverage of power modes and transitions between them.

Data provided in Table 2 include the number of characters for ESL specification, UPF specification, PMU description, and assertions specification. The last column provides directive coverage, achieved during 10 μs simulation time. This coverage is reported using the generated coverage assertions. There was no intention to exhaustively verify all PMUs, since it would require a significant

amount of time and effort (especially for complex PMUs). A goal was just to show that the synthesized PMUs are syntactically correct and ready for verification. To achieve 100 % coverage for more-complex PMUs, another stimuli-generation approach should be adopted (e.g. directed test generation). The summarized number of characters for generated data is really high, compared to the number of characters required for a designer to manually write in the ESL specification. For these samples, the proposed methods represent 132 times fewer characters to write manually in average in order to obtain the RTL power management ready for verification.

6.3 PMU Modification Effect

We have also performed the experiments evaluating the effect of the proposed PMU's power-state machine modification on its power consumption. Some results of these experiments have been published in [13]. A summarized result is that the modification can save up to 76 % of power for systems with rare power-mode switching. For the modification to be beneficial (area overhead is acceptable for gained power savings), the PMU must be at least 60 % of the operation time inactive, what most of the systems satisfy. For simple PMUs, the area overhead is significant (up to 50 %), but it can decrease under 1 % for more complex PMUs.

7. Conclusions

The dissertation thesis was oriented towards the utilization of the ESL abstraction level for design of low-power systems. We have proposed a novel methodology, consisting of several methods for design and verification processes, extending the current standard low-power design flow based on UPF. The main contribution lies in the simplified power-management specification, higher automation in the design flow, and early verification of introduced aspects.

The simplified specification is achieved by the proposed method for specification of low-level power-reduction techniques in an abstract form. Automation is contained in the power-management high-level synthesis process, generating UPF standard specification at RTL, the PMU controller, as well as the power-management assertions for verification. The proposed verification starts form the very beginning of the design process and continues until the RTL is reached and the existing methods and tools can be used. The verification process provides a proper guidance for a designer to get the power-management specification right - correct, complete, and consistent with functional specification. As a result the low-power systems design is simplified, enabling also the designers not familiar with power-reduction techniques to design for low power.

Further work in this area can be oriented towards a complete abstraction from power management, achieved by even more automation in the design. There is a need to automatically divide a system into power domains, assign to them suitable power states, and create a proper power-management strategy (specify allowed power modes and switch between them). The ESL simulation can help to pinpoint these aspects.

Acknowledgements. This work was partially supported by the Slovak Science Grant Agency (VEGA 1/1008/12 and VEGA 1/0616/14), the Slovak Research and Development Agency (SK-CZ-2013-0173), Slovak University of Technology in Bratislava ("ANSNS" and "ASRSS"), and COST Action IC 1103 MEDIAN.

References

[1] J. Ahuja. Towards a green electronic world: A collaborative approach. In *ISLPED '08 Proceedings of the 13th international symposium on Low power electronics and design*, pages 1–2. ACM, 2008. Keynote.

[2] S. Ahuja. *High Level Power Estimation and Reduction Techniques for Power Aware Hardware Design*. PhD thesis, Faculty of the Virginia Polytechnic Institute and State University, 2010.

[3] S. Bailey, G. Chidolue, and A. Crone. *Low Power Design and Verification Techniques*. Mentor Graphics, 2007. White paper.

[4] T. Bouhadiba, M. Moy, and F. Maraninchi. System-level modeling of energy in TLM for early validation of power and thermal management. In *DATE '13 Proceedings of the Conference on Design, Automation and Test in Europe*, pages 1609–1614, 2013.

[5] Cadence Design Systems. *A Practical Guide to Low Power Design: User Experience with CPF*. 2012. http://www.si2.org/?page=1061.

[6] IEEE Standard Association. *IEEE Standard for Design and Verification of Low Power Integrated Circuits*. IEEE, 2013. IEEE Std 1801-2013.

[7] ITRS. *International Technology Roadmap for Semiconductors: Design*. ITRS, 2011.

[8] J. Karmann and W. Ecker. The semantic of the power intent format UPF: Consistent power modeling from system level to implementation. In *2013 23rd International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pages 45–50. IEEE, 2013.

[9] N. Khan. Cosed-loop verification methodology for low-power SoC design. *Special Technology Report - Low Power Design*, pages 7–8, September 2008.

[10] H. Lebreton and P. Vivet. Power modeling in SystemC at transaction level, application to a dvfs architecture. In *IEEE Computer Society Annual Symposium on VLSI*, pages 463–466, 2008.

[11] D. Macko and K. Jelemenská. Managing digital-system power at the system level. In *IEEE Africon 2013 Sustainable Engineering for a Better Future*, pages 179–183. IEEE, 2013.

[12] D. Macko and K. Jelemenská. Self-managing power management unit. In *Proceedings of the 2014 IEEE 17th International Symposium on Design and Diagnostics of Electronic Circuits and Systems*, pages 159–162. IEEE, 2014.

[13] D. Macko, K. Jelemenská, and P. Čičák. Power-efficient power-management logic. In *2014 24th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pages 105–111. IEEE, 2014.

[14] D. Macko, K. Jelemenská, and P. Čičák. Power-management specification in SystemC. In *Proceedings of the 2015 IEEE 18th International Symposium on Design and Diagnostics of Electronic Circuits and Systems*, pages 259–262. IEEE, 2015.

[15] O. Mbarek, A. Pegatoquet, and M. Auguin. Using unified power format standard concepts for power-aware design and verification of systems-on-chip at transaction level. *IET Circuits, Devices and Systems*, 6(5):287–296, 2012.

[16] Mentor Graphics. Vista low power solutions. http://www.mentor.com/esl/vista/low-power.

[17] Y. Xu, R. Rosales, B. Wang, M. Streubühr, R. Hasholzner, C. Haubelt, and J. Teich. A very fast and quasi-accurate power-state-based system-level power modeling methodology. In *ARCS'12 Proceedings of the 25th international conference on Architecture of Computing Systems*, pages 37–49, 2012.

Selected Papers by the Author

D. Macko, K. Jelemenská, and P. Čičák. Power-management specification in SystemC. In *Proc. of the 2015 IEEE 18th Int. Symposium on Design and Diagnostics of Electronic Circuits and Systems – DDECS 2015*, pages 259–262, Belgrad, Serbia, 2015. IEEE.

D. Macko, K. Jelemenská, and P. Čičák. Power-efficient power-management logic. In *2014 24th Int. Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pages 105–111, Palma (Mallorca), Spain, 2014. IEEE.

D. Macko, K. Jelemenská. Self-managing power management unit. In *Proc. of the 2014 IEEE 17th Int. Symposium on Design and Diagnostics of Electronic Circuits and Systems – DDECS 2014*, pages 159–162, Warsaw, Poland, 2014. IEEE.

D. Macko, K. Jelemenská. Managing digital-system power at the system level. In *IEEE Africon 2013 Sustainable Engineering for a Better Future*, pages 179–183, Mauritius, 2013. IEEE.

D. Macko. Contribution to the low-power design. In *Počítačové Architektury & Diagnostika – PAD 2014*, pages 123–128, Malá Skála, Czech Republic, 2014. Technical University of Liberec.

D. Macko. System-level power management specification. In *Počítačové Architektury & Diagnostika – PAD 2013*, pages 87–92, Klášter Teplá, Czech Republic, 2013. University of West Bohemia.

D. Macko. Functional verification of the digital system design. In *Počítačové Architektury & Diagnostika – PAD 2012*, pages 61–66, Milovy, Czech Republic, 2012. Czech Technical University in Prague.

D. Macko, K. Jelemenská. Power-intent integration into the digital system specification model. In *Proc. of The Second Workshop on Manufacturable and Dependable Multicore Architectures at Nanoscale – MEDIAN 2013*, pages 49–52, Avignon, France, 2013.
http://www.median-project.eu/events/median2013/proceedings.