Adopting High-level Synthesis Approach to Accelerate Power Management Design

Katarina Jelemenska, Dominik Macko
Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava, Slovakia
katarina.jelemenska@stuba.sk, dominik.macko@stuba.sk,

***Abstract*— High-level synthesis (HLS) is rapidly gaining its position in hardware design. With nowadays designs complexity and continuously growing pressure to cut down the time-to-market it is now inevitable to raise the hardware design from Register-Transfer Level (RTL) to the higher-level of abstraction, commonly known as Electronic System Level (ESL). The HLS naturally brings an increase in design productivity and by adopting current techniques like IP reuse and formal verification the design correctness could be improved as well. However, the available studies show that the recent HLS tools still have a lot of limitations. Not only the design quality could be improved, concerning for example performance or energy-efficiency, but also various design techniques, currently applied at RTL and lower levels, should be supported at the ESL. The low power design techniques belong to the group. Application of these techniques is especially important in fault-tolerant systems, where an incorporated overhead results in highly increased power consumption. Although low power design is supported by standard RTL specification it is still significantly challenging, as well as highly error prone work, to apply these techniques to the HLS synthesized design. We present an approach to ESL power intent specification, together with the proposed HLS methods for generating an equivalent standard RTL specification of power management. The approach substantially reduces the power management specification and provides for rapid RTL-precise power estimation, offering thus the fast exploration of various power architectures. What is more, our current research aims for automated ESL power intent generation that could make the adoption of low power design techniques fully transparent.**

Keywords—design automation; power intent specification; power-management design; high-level synthesis; rapid power-management exploration.

I. Introduction

The answer to the ever growing electronic design complexity is the current trend to shift the functional design starting point to the electronic system level (ESL). At the higher abstraction level the functional specification can be developed much faster and it is easier to manage. The electronic system level is nowadays supported by various high-level synthesis (HLS) tools that can be used to transform the ESL functional specification to its lower-level equivalent, typically an RTL (Register-Transfer Level) model.

Although the functional design process is successfully moving to the ESL, the available studies still reveal a lot of limitations in the current HLS tools [1-3]. Typically, the synthetized design can be further optimized, concerning for example performance or energy-efficiency. There are still a lot of design techniques that are not supported at the ESL and have to be applied to the synthetized design further in the development process, at the RTL and lower levels.

The emergence of finer process CMOS technologies (below 90 nm) intensified the problem of sustaining temperature and reliability of devices, while their power density is rising. As a result, the power became the key constraint in every electronic design and various techniques have been developed to reduce the power. Application of the power-reduction techniques is especially important in the systems, where higher reliability is required (e.g. fault-tolerant systems). The replication of hardware components, often utilized in these systems, results in power consumption replication, which is ever increased by added comparison/voting logic. The software redundancy also represents some overhead, resulting in a power consumption increase. On the other hand, the rising power consumption decreases the system reliability.

The dynamic power management became the most widely spread approach to apply the power-reduction techniques. Therefore, the power management should also be specified at the ESL and supported by the HLS process. Otherwise, the manual modification of synthesized functional design is required at RTL, which is both error prone and complex and results in the increased verification effort.

The ESL power-management specification should be easy to use and modify, enabling efficient power-management strategies exploration. However, for this purpose, the fast and fully automated power-management HLS represents another precondition. In addition, the HLS verification can not be excluded, including at least a method for equivalence checking between the higher- and lower-level models.

The paper is organized as follows. In Section II, the related work is discussed. The proposed low-power design flow is introduced in Section III. In Section IV, the developed power-intent specification method is briefly explained, followed by the proposed HLS methods description in Section V. In Section VI, the methods evaluation is provided. Finally, the results are summarized and further work outlined in the concluding section.

This is an accepted version of the published paper:
K. Jelemenska and D. Macko, "Adopting high-level synthesis approach to accelerate power management design," in 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2018, pp. 124-130.
doi: 10.1109/ICRITO.2018.8748268
URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8748268

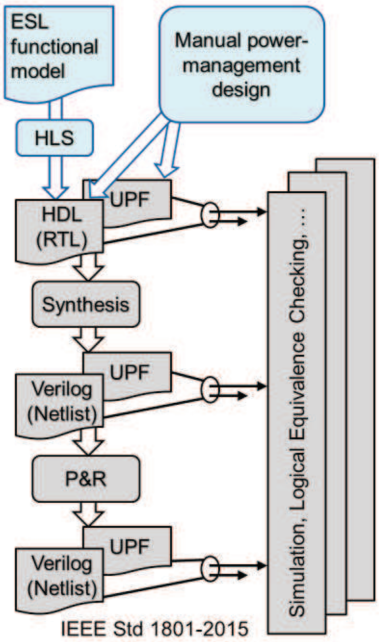


Fig. 1 Current low-power systems design flow

II. Related Work

At RTL and lower abstraction levels the power-reduction techniques are supported by dedicated languages like UPF (Unified Power Format) [4] and CPF (Common Power Format) [5]. Although UPF standard has been updated recently to support ESL [6], it still does not support a high-level abstract power-management specification (i.e. a lot of low-level details, like power switches or isolation cells, is required in UPF specification). For this reason, it is currently a common practice to design the UPF/CPF power-management specification manually at RTL to go together with the HLS synthesized functional specification through the further design flow. The situation is illustrated in Fig.1, where the typical use of UPF in low-power design flow is outlined. In this way, the power-management specification increases the design complexity and prolongs the design and verification time.

What is more, the UPF/CPF specification is separated from the functional model. It remains separated throughout the whole design flow and it participates in the verification process of the functional model. This separation keeps the RTL models manageable to a certain extent, however, the combined specification would provide the overall design picture and would be therefore more convenient at ESL.

Several approaches have been published recently, intended to support power-intent specification at the ESL. For example, the authors in [7] propose a framework that enables the ESL power-intent specification, modeling, and power estimation. Unfortunately, the specification method used does not have analogy in UPF/CPF standard, therefore its synthesis to the RTL and the following verification would be difficult to complete. The UPF power-management specification approach, abstracted to the ESL was used in [8]. This makes the verification of power-intent equivalency easier, however, the separation of power management and functional specification prevents an overall perspective of the system power management. The authors in [9] describe generic power-management module at the ESL and adopt HLS approach to generate low-lever power intent. Nevertheless, among all the power-reduction techniques, only power shut-off is supported in this case. Similarly to [8], our approach is also based on UPF concepts abstracted to the ESL. In our earlier work [10] we integrated the abstracted power-management constructs directly into the syntax of HSSL (Hardware-Software Specification Language). The proprietary language though suffers from lack of HLS support, so additional compilation of functional model was needed to utilize the available HLS tools.

There are other works devoted to the system-level power management modeling [11-13]. Some of them use too much low–level details to support an efficient exploration of various power-management strategies. In the others, on the contrary, the abstraction level is too high, preventing thus the proper power and area overhead estimation of the introduced power management.

Several other research teams, like [14-15] concentrate their effort on the system-level power consumption monitoring and estimation. These approaches are based on SystemC extension libraries, but they do not support power management, which can substantially influence the power consumption when introduced to the system at the lower levels. Other approaches [12], [16-19], supporting power management exploration, require components power characteristics, achieved from the lower-level power estimations. Also, the overhead of the power-management unit (controlling the power management) is not considered. However, some methods, like [11], [13] and [20], do support power-management unit modelling, although in a manual manner.

To summarize, the available extensions of the standard power-management specification to the ESL have several drawbacks. Some methods suffer from insufficient abstraction, which imply a need to specify details that are improper for the ESL (e.g. power-supply networks, level shifters, etc.). Other methods enable to abstract from unnecessary details, but their problem is lack of automation. Therefore, they count on the manual introduction of power management in the UPF. A lot of approaches are based on system-level power consumption estimation, relying on the prior components power profiling, which is time consuming. Finally, in most of the solutions, the verification process is not fully supported.

III. The Proposed Low-Power Design Flow

Our idea was to identify the power-reduction techniques that are suitable to be used at the ESL and to propose a convenient method for system-level power-management specification. The power-management specification should be integrated into the ESL functional specification in order to provide an overall view of the system architecture and its power management. An HLS approach should then be adopted in order to connect the proposed system-level specification to the standard low-power design flow starting at the RTL.

As a result, the novel low-power systems design methodology was proposed, which extends the standard UPF-based design flow to the system level of abstraction (ESL). The

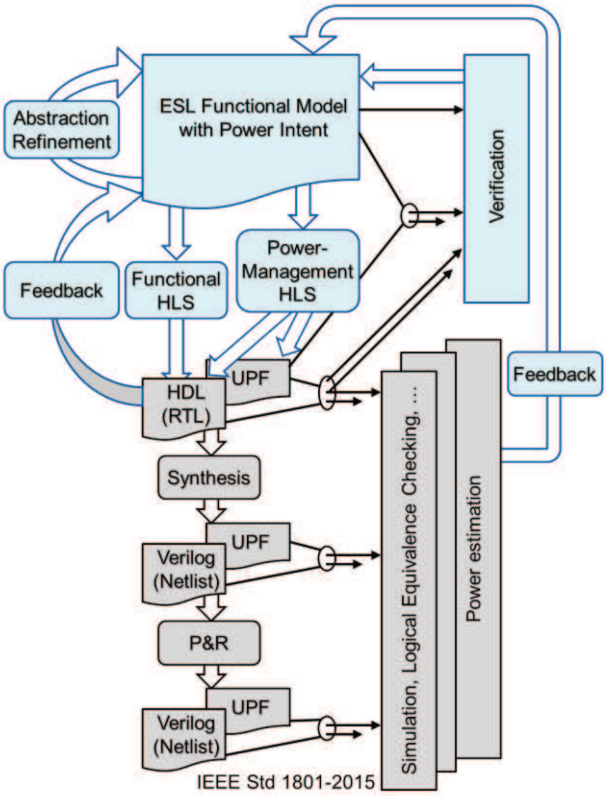


Fig. 2 The proposed ESL extension of low-power design flow.

situation is illustrated in Fig. 2. At the ESL, multiple UPF concepts are incorporated directly into the functional specification model in an abstract form. Such a model then goes through abstraction-refinement process, during which more details are added to the specification. This process is accompanied by multiple verification steps. When the system-level model is sufficiently detailed, it can be transformed to the RTL models. For this purpose, the novel HLS method was developed for synthesis of abstract power-management specification into the standard UPF form and the inevitable extensions of a functional RTL model. The functional RTL model itself will be generated by means of available HLS tools.

To ensure the correctness of the synthetized power management, the verification method was proposed as well, utilizing the current formal and informal techniques. After this transformation, the equivalence checking ensures that the RTL model corresponds to the specification. Currently, there are several RTL tools available that can be used to verify and analyze the model, and thus to obtain trustworthy information about power consumption, performance, or area. Based on this information, it is then possible to modify the specification in order to obtain several design alternatives. In this way, a suitable tradeoff among various parameters can be achieved. Starting from RTL, the proposed design methodology relies on standard UPF-based design flow. It means that the existing methods and trusted tools for low-power design can be used at later design stages.

TABLE I. COMPARISON OF ESL AND UPF SPECIFICATION

Power state	**UPF components required**	**Power-reduction technique used**
normal	main supply net	no
off	power switch isolation cells (inputs and outputs)	power-gating
off_ret	power switch, isolation cells (inputs and outputs) state retention	power-gating with state retention
hold	isolation cells (inputs) isolation control	clock-gating operand-isolation
diff_level#	additional supply net power switch level shifter	voltage and frequency scaling multiple fixed supply voltages

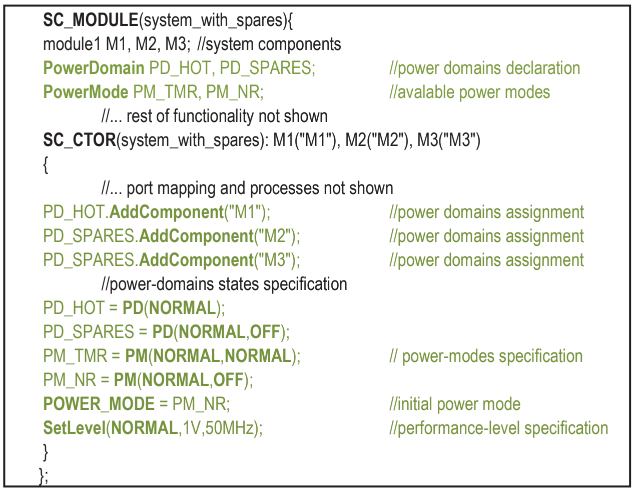
IV. ESL Power Intent Specification

In our approach, multiple UPF concepts are incorporated directly into the functional specification model in an abstract form. Similarly to UPF we also define power domains that will be assigned a cluster of system components (blocks), operating always in the same power states. For each power domain, several power states can be defined. The overall power situation in the system model is then represented by power mode. Each power mode value stands for a specific combination of power states of the individual power domains. The system power management at the ESL is thus as easy as an assignment of a different value to the power mode variable.

More specifically, five power states are supported: *normal, off, off_ret, hold,* and *diff_level#*. The *diff_level#* actually represents a group of states, each defined by a different voltage-frequency pair (called also performance level) and assigned a unique number in place of the # sign. In fact, any power state is dependent on supply voltage and operation frequency. Other details, like power-management elements, power switches etc., are abstracted from the ESL specification and will be introduced automatically during the HLS process. The supported power states are summarized in Table I, together with the required UPF components that have to be generated during the HLS process and the power-reduction techniques applied in the respective states.

The *off* and *off_ret* power states both represent the situation when the power supply of the domain is switched-off (i.e the power-gating technique is applied). The only difference is that the retention must be set for the domain in case of the *off_ret* power state. In the *hold* power state the domain remains powered, but isolated from the rest o the system (the clock-gating and operand-isolation techniques are used).

The proposed system-level power-management specification method was implemented in the form of the extension library with the predefined classes and macros, implemented in the C++ language [21]. Thus the compatibility with the available tools, such as compilers, has been retained. The library, called PMS (Power Management Specification), is dedicated to SystemC specification methodology extension. The example in Fig.3 illustrates the PMS constructs (in green color) incorporation into a SystemC model. The SystemC was



```
SC_MODULE(system_with_spares){
module1 M1, M2, M3;  //system components
PowerDomain PD_HOT, PD_SPARES;              //power domains declaration
PowerMode PM_TMR, PM_NR;                    //avalable power modes
        //... rest of functionality not shown
SC_CTOR(system_with_spares): M1("M1"), M2("M2"), M3("M3")
{
        //... port mapping and processes not shown
PD_HOT.AddComponent("M1");                  //power domains assignment
PD_SPARES.AddComponent("M2");               //power domains assignment
PD_SPARES.AddComponent("M3");               //power domains assignment
        //power-domains states specification
PD_HOT = PD(NORMAL);
PD_SPARES = PD(NORMAL,OFF);
PM_TMR = PM(NORMAL,NORMAL);                 // power-modes specification
PM_NR = PM(NORMAL,OFF);
POWER_MODE = PM_NR;                         //initial power mode
SetLevel(NORMAL,1V,50MHz);                  //performance-level specification
}
};
```

Fig. 3 SystemC/PMS source-code fragment.

an obvious choice, since it is currently the most widely used ESL model, enabling both the algorithmic and architectural modelling. It enables model execution without a need to implement any simulation tool. In this way, an abstracted UPF specification can be integrated into the functional SystemC model quite easily and it is denoted by expression SystemC/PMS.

The proposed abstraction from low-level details enables a designer to focus on the design functionality. Nevertheless, in this approach, the power management is not modelled at the ESL, it is only specified. That means, no ESL power consumption estimation is provided. Therefore, a designer has to rely on the RTL model power analysis, which, on the other hand, is more precise, compared to the currently available ESL power-consumption-estimation approaches. However, this fact makes the fast, reliable, and automated synthesis of UPF power management inevitable.

V. The Proposed HLS Methods for ESL Specification Synthesis

In order to provide a designer with a possibility of fast design-space exploration, which will be based on the design alternatives comparison at the RTL, the UPF concepts synthesis has to be automated and as fast as possible. Several methods and algorithms have been developed that will perform the necessary synthesis steps.

A. Power-management high-level synthesis

The high-level synthesis process is separated into two parts: functional HLS and power-management HLS. For the functional HLS, a currently available HLS tool can be used that supports SystemC specification. This process is commonly used in the industry today. It transforms a functional specification into a functional description in some hardware description language. Remember that in the functional specification, there is the power-mode switching specified. Thus, the functional RTL model contains the power-management policy algorithm.

However, to perform automated transformation of abstract power management into an equivalent UPF specification, the novel power-management HLS method had to be proposed, followed by the tool development. Based on the abstract power-management specification, the UPF specification is generated along with the functional description of the corresponding power-management unit. This process is also enhanced by the optional and automated specification optimization, which utilizes the analysis of the specified aspects to resolve some kinds of specification inconsistency and to remove redundant constructs.

In order to generate the standard UPF specification at the RTL, several issues have to be solved. First, the PMS constructs have to be recognized and analyzed in the SystemC/PMS specification. However, because of the high level of abstraction not all the information required for UPF commands generation are included in the PMS specification. Instead, the low-level components (supply nets, power switches, isolation cells etc.) have to be specified based on the SystemC model itself. Therefore, several passes through the model representation will be necessary. The proposed algorithms for standard UPF synthesis have been implemented into the PMS2UPF tool [22].

B. Power-management unit synthesis

As it was explained in the previous chapter, in the ESL power-intent specification, the overall state of the system is represented by the power mode (variable POWER_MODE in the PMS library), which is first assigned an initial value. By the initial power mode the initial power state of each power domain is determined. Instead of changing the power state of individual power domains, the system power mode is changed. Typically, only a limited number of power modes is defined, representing the UPF power-state table items. The switching among power modes is specified in the functional part of the ESL specification.

At RTL, the generated UPF low-level power-management components (power switches, isolation cells, retention cells, and level shifters of respective power domains) need to be controlled according to the specified power intent. For this purpose a power management unit (PMU), in the form of an HDL model, is generated and integrated into the RTL functional model. The PMU consists of two parts: Transition logic – determining the target power mode and power-state machine – generating the actual control-signals sequences for power-management components.

At this level, the target power mode is encoded by power-management components control signals. In order to correctly reach the determined target power mode, the control signals have to be generated in the precise time. This is the task of power-state machine. For example, to set *off* or *off_ret* power state in a power domain, first the clock signal has to be stopped, then the domain has to be isolated. For *off_ret* power state, the isolation is followed by retention of registers' state, and finally, the power is switched off.

Into the power-state machine, a simple management of its own power consumption has been integrated, which enables to power down the transition logic when the power mode does not need to be switched. It can significantly reduce its leakage power.

C. Power-management verification

The verification in the proposed ESL extension of low-power design flow (see Fig. 2) plays the crucial part of the whole methodology. The proposed verification approach consists of multiple verification steps, which begin at the early design stages at the system level. In this way, a designer is guided to create the correct and consistent specification of power management from the very beginning. The approach is based on combination of current formal and informal techniques. The internal structure of verification is illustrated in Fig. 4, where the three key verification methods are depicted.

First, the power-management static analysis is used to verify the consistency and completeness of the developed ESL specification. This is the unique method, dedicated to the proposed ESL power-intent specification. This verification step simplifies the design process by supporting the power management refinement at the early design stages. In case an issue is detected, a designer is notified about the source of the problem. The static analysis enables also to carry out the optimization during the HLS process.

Second, the power-management HLS is verified by means of dedicated method, based on equivalence checking technique. In this verification step, the equivalence between synthetized power management and the original power intent is verified. It guaranties that the power intent is preserved after the HLS process. The structural equivalence checking is also performed.

Finally, the assertion-based verification is used to functionally verify the synthesized power-management unit. Assertions are automatically generated during the HLS process and they are used to monitor the correct control-signals sequences in the power-management unit. They are also used to check the coverage of power modes and power states during the functional verification of the RTL model.

In addition to the key verification methods, the combination of syntax checking, along with the basic conditional checks during model execution, ensures that the specification is syntactically and semantically correct. The synthesized UPF is expected to be verified by the available power-aware verification tool at the RTL.

VI. The Proposed Methods Evaluation

The performed evaluations of the proposed approach show the substantial reduction of power-management specification complexity. For comparison of ESL and RTL power-management specifications complexities, the number of characters required for describing the specification has been selected. Another parameter, such as the number of statements, could not be used; because the comparison would be misleading (UPF uses long inline commands). The complexity reduction is not achieved by a shorter description of some statements, but by the utilized abstraction (i.e. the proposed ESL specification is abstracting from low-level details, such as power switches, isolation, or supply network). Using the pseudorandom approach, over ten thousand samples of abstract power-management specification have been generated. In order to generate specifications with various complexities, several parameters have been scaled. For example, the number of power domains in the system or the number of system power modes. The specification samples were then synthesized into the UPF form, using the developed synthesis algorithms – with and without optimization.

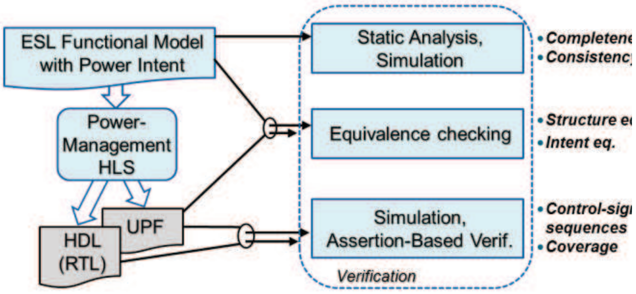


Fig. 4 The key power-management verification steps.

The results of the experimental comparison are illustrated in Fig. 5 and Fig. 6. The vertical axis in the figures uses a logarithmic scale. The results indicate that the synthesized power-management specification in UPF is about 16.8 times more complex (in average) than the proposed SystemC/PMS specification method. In case the developed synthesis algorithm with optimization is used, the UPF specification is about 14.3 times more complex in average than the SystemC/PMS specification. Thus, the proposed abstract ESL power-management specification is significantly simplified compared to the standard specification at the RTL. Moreover, the optimization reduces the UPF complexity approximately by 15% in average, which means that fewer UPF constructs will be used but with the preserved power intent (unnecessary elements are not synthesized). The proposed power-management abstraction offers an easy way of power-management strategy modification. Providing thus,

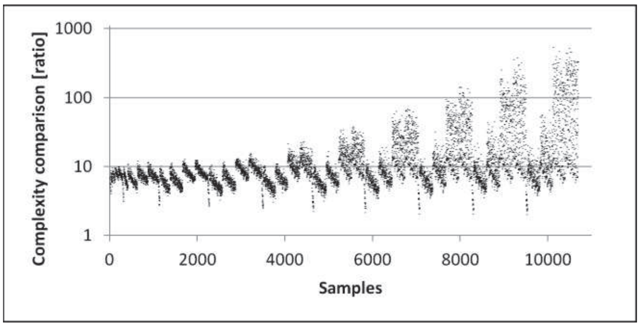


Fig. 5 Synthesized unoptimized UPF to SystemC/PMS comparison.

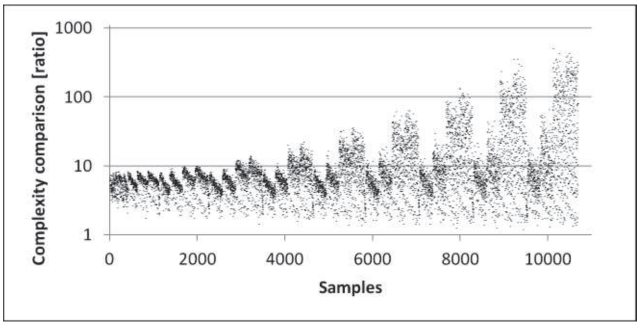


Fig. 6 Synthesized optimized UPF to SystemC/PMS comparison.

accompanied with automated and fully verified HLS methods, the rapid RTL-precise power estimation.

The synthesized UPF specifications have been verified by Modelsim SE 10.2c for syntactical and semantical correctness. The synthesized power-management units in VHDL have also been verified using the assertion-based verification during power-aware functional simulation in Modelsim. Moreover, the synthesized power management has passed the developed equivalence checking, thus the equivalence between the ESL and RTL power intent has been ensured.

VII. Conclusions and Further Work

The paper presents an approach that enables to shift the low-power system design to the ESL. The approach is based on dynamic power management that can be specified in an abstract, simple, and well-arranged manner. Supported by the proposed HLS methods and tools, the equivalent RTL power-management specification can be automatically generated and verified. In this way the fast and RTL-precise exploration of various power architectures is made possible.

Our current research is targeted to automated ESL power intent generation. This would make the adoption of low power design techniques fully transparent. It includes automated partitioning of the system into power domains, automated assignment of suitable power states to these power domains, automated synchronization of communication between clock domains, automated selection of usable power modes, and automated switching among power modes according the current requirements. Our first results indicate that such automation is possible and beneficial, especially for designers not familiar with power-reduction techniques. However, we must further deal with some limitations (e.g. dependency on simulation results) and refine/finish some aspects mentioned above for the method to be fully usable.

Acknowledgment

This work was supported by the Slovak Research and Development Agency (APVV-15-0789), and the Ministry of Education, Science, Research and Sport of the Slovak Republic within the Research and Development Operational Programme for the project “University Science Park of STU Bratislava”, ITMS 26240220084, co-funded by the European Regional Development Fund.

References

[1] R. Nane et al., „A Survey and Evaluation of FPGA High-Level Synthesis Tools,“ in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 35, No. 10, 2016, pp. 1591-1604.

[2] K. Campbell, W. Zuo, D. Chen, “New Advances of High-Level Synthesis for Efficient and Reliable Hardware Design,” in Integration, the VLSI Journal, Vol. 58, 2017, pp 189-214.

[3] T. Hill, , “11 Myths About High-Level-Synthesis Techniques for Programming FPGAs,” [Online]. Available: http://www.electronicdesign.com/embedded-revolution/11-myths-about-high-level-synthesis-techniques-programming-fpgas, 2017.

[4] IEEE Standard for Design and Verification of Low Power Integrated Circuits, IEEE Standard 1801-2013, May 2013.

[5] Cadence Design Systems, A Practical Guide to Low Power Design: User Experience with CPF, 2009.

[6] IEEE Standard for Design and Verification of Low-Power, Energy-Aware Electronic Systems, IEEE Standard 1801-2015, Dec. 2015.

[7] 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, 2008, pp. 463-466.

[8] 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 & Systems, vol. 6, no. 5, pp. 287-296, 2012.

[9] A.Rogers, “Designing a simple system-on-a-chip in under 60 minutes with the mu0 microprocessor and Xilinx tools,” 2003. [Online]. Tut.. Av: http://www.ece.uah.edu/~lacasa/tutorials/mu0/mu0tutorial.html

[10] D. Macko and K. Jelemenská, “Managing digital-system power at the system level,” in IEEE Africon 2013 Sustainable Engineering for a Better Future, 2013, pp. 179-183.

[11] H. Affes et al, “An ESL framework for low power architecture design space exploration,” in 2016 IEEE 27th International Conference on Application-specific Systems, Architectures and Processors (ASAP). IEEE, 2016, pp. 227–228.

[12] K. Gagarski, M. Petrov, M. Moiseev, and I. Klotchkov, “Power specification, simulation and verification of SystemC designs,” in 2016 IEEE East-West Design & Test Symposium (EWDTS). IEEE, 2016, pp. 1–4.

[13] A. Qamar, F. B. Muslim, J. Iqbal, and L. Lavagno, “LP-HLS: Automatic power-intent generation for high-level synthesis based hardware implementation flow,” Microprocessors and Microsystems, vol. 50, pp. 26–38, 2017.

[14] F. Klein, R. Azevedo, L. Santos, and G. Araujo, “SystemC-based power evaluation with PowerSC,” in Electronic System Level Design: An Open-Source Approach, S. Rigo, R. Azevedo and L. Santos, Eds. Springer, 2011, pp. 129-144.

[15] “PKtool: Power estimation in SystemC,” [Online]. Available: http://pktool.sourceforge.net/

[16] D.Greaves, M.Yasin, “TLM POWER3: Power estimation methodology for SystemC TLM 2.0,” in Models, Methods, and Tools for Complex Chip Design, LNEE, vol. 265, J. Haase Ed. Springer, 2014, pp. 53-68. [10] D. Greaves and M. Yasin, “TLM POWER3: Power estimation methodology for SystemC TLM 2.0,” in Models, Methods, and Tools for Complex Chip Design: Selected Contributions from FDL 2012, ser. Lecture Notes in Electrical Engineering, J. Haase, Ed. Springer International Publishing, 2014, vol. 265, pp. 53–68.

[17] 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. San Jose, CA: EDA Consortium, 2013, pp. 1609–1614.

[18] 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). IEEE, 2013, pp. 45–50.

[19] F. Mischkalla and W. Mueller, “Advanced SoC virtual prototyping for system-level power planning and validation,” in 2014 24th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS). IEEE, 2014, pp. 112–119.

[20] Y. Xu, et al, “A very fast and quasi-accurate power-state-based systemlevel power modeling methodology,” in ARCS’12 Proc. of the 25th Int. Conf. on Arch. of Comp. Sys., Berlin, 2012, pp. 37–49.

[21] IEEE Standard for Standard SystemC Language Reference Manual, IEEE Standard 1666-2011, Jan. 2012.

[22] M. Siro, D. Macko, and K. Jelemenská, "PMS2UPF: An automated transition from ESL to RTL power-intent specification," 2017 IEEE 20th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), Dresden, 2017, pp. 140-144.