# Power Awareness in System-level Design of Embedded Systems

Fana Teffera

***Abstract*— This paper briefly discuss the current progress and gaps in power awareness in system-level design of embedded systems. We present the recent works and comment on the presented works.**

## I. Introduction

The increasing need for comprehensive early design abstraction due to the acceleration of complexity in embedded systems design and the need to produce high-quality system-level specifications demands incorporation of a high-level power consumption prediction model.

The power consumption of embedded systems is one of the non-functional requirements that will shape the design of the system. System feasibility depends on how well the system uses the allocated power resource, especially if it is battery-powered. The increased consumption of power will increase heat dissipation and limit battery life. This could be accelerated by possible mishaps in hardware or software design. [Hen+16]

Power dissipation is modeled as the sum of dynamic power and static power. The dynamic power value is the result of charging and discharging the capacitance of the gates and interconnect during current switching as well as the power while the input is in transitions. The switching power is dominant for synchronous designs that can be modeled as follows

$$P_{switch} = \frac{1}{2} s C V_{dd}^2 f \tag{1}$$

where s is switching activity, C is the capacitance, f is frequency and $V_{dd}$ is the supply voltage.

For static power calculations, sub-threshold and gate leakage is considered more dominant factor. The sub-threshold leakage is calculated as follows

$$P_{subthreshold} = I_{sub} \frac{W}{L} V_{dd} e^{\frac{V_{th}}{nV_T}} \tag{2}$$

where $I_{sub}$ and n are  are specific to the chosen technology, W and L are dependent on the device geometry, $V_{th}$ is the threshold and $V_T$ is the thermal constant.

Designing power consumption optimization at the last stage of the design process is expensive and time-consuming. On the other side, even if it seems attractive to abstract the system as much as possible, a good approximation for the process execution time should be made using simulation, possible feedback from hardware performance testing, and other feedback that will increase the usability of the model.

The complexity of the synthesis is also increased as we incorporate power awareness in our design. This complexity shouldn't significantly affect product delivery time. Also, a system-level approach that is not based on valid models will result in a poorly designed system that could be costly.

Adding the power consumption model early on the design phase is needed to consider the operating voltage, type of capacitance needed, switching activity, the threshold voltage, the size of the transistors, and the operating temperature. A significant complication could arise due to the relationship among these factors and careful simplification is required when designing a usable model for a practical system.

A consideration of the complicated relationship between the supply voltage and frequency adjustment coupled with introduction of time-bound scheduling deadlines is needed to produce a usable model. We could implement a system that will mitigate the complication that arise from the operating voltage and frequency, by having more than one operating voltage in simulations operation where non-critical operations will have a reduced power, segregation of processing elements to reduce the complexity in power distribution, and implementation of different task scheduling mechanism, i.e. Dynamic voltage and frequency scaling (DVFS) where scaling is done based on scheduled timing. Even without consideration of other practical scheduling needs, with ever increasing heterogeneity as well as complexity of the hardware designs, these kind of considerations are not an easy feat to accomplish. [DSJ06]

This paper explores recent works that tried to navigate this complication following different design approaches.

## II. Literature Review

System-level estimation of power consumption using SystemC is discussed in Giammarini et al. work. The proposed system provides a C++ energy model library that is implemented using the SystemC model. They showed a system that maps a specific data type with an energy model. The C++ library generates energy models that will be added to the kernel and will generate SystemC modules that will also generate energy model and add to the kernel. This energy model that is added to the kernel will be used when the simulations are initiated. The paper claims 15.8% mean relative error between the estimated power consumption and measured power consumption's. [GCO11]

Capturing power-intent at the system-level by power architecture exploration using SystemC/TLM model, which supplement unified power format(UPF) specification, is discussed by Mbarke et al. This approaches

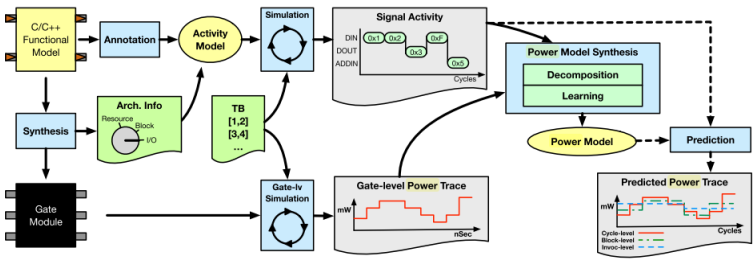


Fig. 1. Power Modeling Flow : [LG18]

uses abstract UPF specification, that characterize the system with power-gating and multi-voltage constraints as inputs to generate UPF files that can be synthesized to VHDL, to be placed and routed on the Transaction level. TLM will be transferred to RTL, synthesized, and placed on TL level implementation according to the netlist derived modeling. Early verification of the power modeling is done by using the UPF semantic at TL level. This deals with functional failures, power components placement, and component-to-component interactions. [MPA12]

Grüttner et al. [Grü+13] shows a virtual prototyping generation to evaluate different design approaches. This is limited power-focused approach that may not capture the reasonable complexity of embedded system design factors.

Macko et al. present a methodology with a list of requirements that propose the simplicity and unified approach of power reduction mechanism to increase the simplicity of system-level models. Their approach of power management in the early design process is the differentiation of the blocks to different power domains. By having this statically defined, they proposed that consistency error of the system domain is checked early in the design process. They claimed that the power management methodology could be verified early on in the design process using C++ compiler check of the specifications that are derived using the functional design specifications and feedback from run-time errors during the conditional checks. [MJČ18]

The above work claims the unique approach of basing the UPF design flow and integrating the current related tools, using power management specification (PMS) into the system to increase the effectiveness and reducing the learning curve of the design process. The coupling of power management with the application estimated power need is also proposed as a means to reduce the complexity of the design. The power management specification needs to match with UPF specification through equivalence check before being implemented. The UPF in this system has a power domain partitioning, power distribution network, and power management elements. The major drawback of this system is the lack of solid verification implementation for the proposed system. This result in the need to introduce other verification systems. [MJČ18]

Their work use clock gating, operand isolation, voltage scaling, multiple supply voltages, frequency scaling, and power gating techniques as power management techniques. Clock-gating works by controlling the IP block synchronization. Operand isolation works with by controlling the participation of datapath elements based on their activity level. They have multiple abstract power states that are *normal* for no power-reduction scheme, *diff_level#* for specified voltage number, *hold* for activated clock gates and operated an isolated system where the block stays powered without no operation, *off* is power gated system with a power loss, unlike the hold state. [MJČ18]

Some of the CAD-based approaches that use MBE are CAT (consumption Analysis Toolbox) and DIPLODOCUS/TTool. CAT uses Architecture Analysis and Design Language (AADL) system which characterize the hardware and software component on the functional level without dynamic power consideration. DIPLODOCUS/TTool is a Y chart based methodology that abstract a system by separating application and architecture modeling. Application modeling is based on a class that has channels, event, and request. The channel will abstract the data, event capture state change and request allocate tasks based on the behavior model. The overall organization and the reliance of formal semantics will increase the speed of the design as well as better translation awareness by based on the formal analysis. [Apv08; Sen+09]

Abdallah et al works show the DIPLODUCS implementation with a Model-Driven approach which adds an extension to the architecture modeling by considering the static and dynamic power usage from the CPU statistics. The voltage and frequency reading is attached to each operation which is counted using an easily configurable power policy. This method relies on prediction of the execution time by adding dynamic power and static power, extract the prediction error from the hardware performance, and adjust the frequency using the collected information. Their result shows that their DVFS modeling approach is able to generate a reasonable prediction that can be integrated into design decision. [Ben+14]

Works of Hendricks shows the model-based approach of system-level design that takes the power analysis in-

consideration. They use model checking means to analyze the power usage when it is implemented by the chosen language. [Hen+16]

As Gao et al. works shows, the usage of empirical models for the power estimation. The approach they proposed embedding the profile in the system with a trained model that takes different trained weights in the system. The model they used for the power estimation is to sum the trained weights and the performance counters multiples. They analyzed the system based on the operation type and the energy requirement for it while it was running on only CPU-only mode. [GSZ18]

Brini et al. works explore the power consumption of different commercial off-the-shelf based configuration using Simulink/StateFlow for ultra-low power wireless nodes. The average power consumption for the sensor node is calculated by adding the time scaled power consumption of the system based on the low-power, transition, and active state mode. They used the reference value from the manufacturer to base power consumption for micro-controller reference power state as well as boost power state. Since their analysis is based on off-the-shelf components, the final implementation of the system can be realized quicker than designing custom hardware. [BDN18]

Models that are abstracted from a gate-level simulation of data-dependent switching activity-based approach is explored by Lee et al. This approach uses a machine learning mechanism to derive an abstraction for the data-dependent power models that are extracted from the functional hardware models with different accuracy level. The power modeling flow this process is shown in Figure 1. [LG18]

As shown in Figure 1, the functional simulation code is translated to the activity model by characterizing the instruction and the I/O activity value. Then using the same input vector, the reference gate-level model is used to feed cycle based results and the high-level synthesis for signal activity tracing using the operation, blocking activity, and I/O operations call-back pattern. The decomposed model trained into the power model with detailed identification of the switching activity to later produce a predictive model which accounts the cycle data-dependency, blocking, and invocation-level power traces. [LG18]

The activity model generation from the XML formatted model in the above model is done by using per cycle trace function insertion. The activity model in this system is restricted based on the architectural information that can be extracted from the compiled individual resourced in the back-end synthesis state. The trace function sends a trace-buffer which return the used hardware resources for the individual resources, blocks and external IOs the state header. The trace-buffer commit the result, as long as there is no pipeline execution is returned, per block call and transfer it power model using the calculated Hamming Distance, which is the binary distance of the committed nearby vectors. If pipeline call is returned, the header of the corresponding mapped stage will be taken and the next tracing process moves to the next stage header.[LG18]

The training of the above system for white box, pre-known hardware configuration, is done using mean absolute error estimations, by measuring the prediction deviation from the result that is obtained using a commercial gate-level power estimator, and normalizing the difference by the arithmetic mean of the same gate-level estimation. The average error percentage is calculated using the same normalizing factor over the high-level based estimated power. The speed used in this system is the simulated cycle per simulation time form the gate-level simulations.[LG18]

In the grey-box model, instead of cycle-based feedback, block-level power estimation feedback with higher estimation speed than the white-box system is used. The linear correlation of pipelined inputs, the input, and near-to-input registers relationship, and the relationship between average power consumption of basic block with single block-level power using input and output history are assumed relationships that drive this model. The author mentioned that the model with a block-by-block estimation approach is 3X faster and within 2% of the average error of the earlier pre-known configuration based model. [LG18]

In the black-box model, due to the lack of significant architectural information, the external IO transnational activity is used to predict the power consumption of our system. The author assumes that this system still knows about the data port mapping, bit widths, and control signals feedback, and additional control registers due to the author's observation of documentation trend of vendors' engineers. [LG18]

The black-box IO tracing function is inserted between function calls. The author claim that their usage of decomposition based training set that is used for ensemble training, assuming the limit of effective diversity in the transaction activity to analyze the decomposed models features effect, and the assumed uniformity of input vectors, increase the relative effectiveness of the model. The author further assumes that the similarity of sum-of-square errors of individual models as well as the depth of the model training to simplify the system. Due to such simplification, the author claims a 6X speed in power estimation and an average error of 2% compared to known configuration model. [LG18]

Qamar et al. work shows the means of automatic power-intent generation using a specification that is written in a common power format (CPF). In their design, they use system-level model files which are written in SystemC to extract design information alongside the power-intent specifications. They generate a CPF file that will be written to RTL design using the common data structuring mode. The combination of behavior transformation of the system-level specification and low-power automatic power generation mode is proposed to show the range of power-intent that could be included

at the system-level. The system-level power analysis is done by reading the SystemC generated function partitioning and allocation which is based on the pragma directives that included manual power-intent specification. This is used later by attaching unique pragmas with each instance to collect the power-related information. This pragma-attached instance information will later use pragma-directed CPF power generation output. [Qam+17]

Other UPF based approaches By Karmann and Ecker [KE13] implements a power-intent based model coupling with the generated data model. Gagarski et al. [Gag+17] show an approach that uses a SystemC based design that generates the UPF specification.

Manually allocated ESL models based on power-state models, power requirement with missing RTL and UPF functionality, temperature-aware traffic based model, are proposed by Lebreton and Vivet [LV08] and Xu et al. [Xu+12] and Bouhadiba et al. [BMM13] respectively.

## III. Discussion

Virtual implementation of the system with separate functional and power intent functions will introduce an unnecessary overhead to the system designer and also will abstract out the other factors that will affect the power consumption of the overall system. This can be a useful tool in designing a well-known system which will behave similarly in the production system as the simulation run, but it is less effective if the power-awareness is affected by a complicated set of other practical real system derived factors. Since system-level modeling is done to ease the over customization of designing tools at the later stage, a less generalized system will also end up missing the fundamental concept of the process. Static analysis of power management is not easily implementable and verifiable in C++ modules for SystemC. This is a weakness in the systems that rely on this kind of approach to justify the early verification means. Mackos et al. used PMHLS to design the verification system. This proposed power management high-level synthesis (PMHLS) verification is dependent on a correct UPF annotation and correctness of the chosen commercial tools. Such static verification will also introduce additional revision and static optimization design loop starting in the later stage if the design clarity of the priority assignment is wrongly specified, a possible case for not well-known architecture with highly interacting components.

Hardware-level implementation of power-consumption restriction is rather complicated and might result in more error due to the need for a comprehensive functional verification system. Added complexity in our verification means, in addition to not matured software level verification, will introduce exaggerated loss of effectiveness in our modeling. [MJČ18]

In many the works we investigated, we noticed that the part which can be explicitly defined at the high-level synthesis than last minute RTL modification to meet previously not designed features is not adequately defined for not-well know architecture.

We also noticed the simplification of affecting factors in Lee et al. [LG18]work that might increase its adoption in real systems, considering the real systems nature of messy process interactions which lead the overall system to behave close to the black-box model.The above conclusion need to be taken with a consideration of the fact that machine learning based models lack consistency.

Unbiased interpretation of the result during the different sample periods and avoidance manual over-specification of systems need to avoided to increase the effectiveness of the models, as it is shown in Brini et al. and Qamar et al. [BDN18; Qam+17] works.

## IV. Conclusion

Many of the system that are presented here are based on a simplified, over-assumed, or semi-verified mechanisms. This shows the complexity of predicting the system performance from the system level, especially considering the hardware dependency on factors that are highly dependent on operating conditions. A significant work is needed to come up with systems that will satisfy both the academic and practitioners.

## References


[Apv08] Ludovic Apvrille. "TTool for DIPLODOCUS: an environment for design space exploration". In: *Proceedings of the 8th international conference on ...* (2008), pp. 214–217. ISSN: 01681702. DOI: 10.1016/j.virusres.2007.10.006. URL: http://dl.acm.org/citation.cfm?id=1416764 (cit. on p. 3).

[BDN18] Oussama Brini, Dominic Deslandes, and Frederic Nabki. "A Model-Based Approach for the Design of Ultra-Low Power Wireless Sensor Nodes". In: *2018 16th IEEE International New Circuits and Systems Conference (NEWCAS)* (2018), pp. 248–251 (cit. on pp. 4, 5).

[Ben+14] Feriel Ben Abdallah et al. "Early power-aware Design Space Exploration for embedded systems: MPEG-2 case study". In: *2014 International Symposium on System-on-Chip, SoC 2014* (2014). DOI: 10.1109/ISSOC.2014.6972450 (cit. on p. 3).

[BMM13] Tayeb Bouhadiba, Matthieu Moy, and Florence Maraninchi. "System-Level Modeling of Energy in TLM for Early Validation of Power and Thermal Management". In: *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013* (2013), pp. 1609–1614. ISSN: 15301591. DOI: 10.7873/DATE.2013.327. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6513773 (cit. on p. 5).

[DSJ06] R P Dick, L Shang, and N K Jha. "Power-aware architectural synthesis". In: *The VLSI Handbook* (2006), pp. 1–21. URL: http://books.google.com/books?hl=en&lr=&id=rMsqBgAAQBAJ&oi=fnd&pg=SA17-PA1&dq=%22systems+of+systems%22+AND+%22power+systems%22&ots=RyKOoT9OGF&sig=FtuhRJAOyJmVo9aJAUk1tLh3IsU (cit. on p. 2).

[Gag+17] Kirill Gagarski et al. "Power specification, simulation and verification of SystemC designs". In: *Proceedings of 2016 IEEE East-West Design and Test Symposium, EWDTS 2016* (2017), pp. 1–4. DOI: 10.1109/EWDTS.2016.7807731 (cit. on p. 5).

[GCO11] Marco Giammarini, Massimo Conti, and Simone Orcioni. "System-level energy estimation with powersim". In: *2011 18th IEEE International Conference on Electronics, Circuits, and Systems, ICECS 2011*. 2011. ISBN: 9781457718458. DOI: 10.1109/ICECS.2011.6122376 (cit. on p. 2).

[Grü+13] Kim Grüttner et al. "The COMPLEX reference framework for HW/SW co-design and power management supporting platform-based design-space exploration". In: *Microprocessors and Microsystems* 37.8 PARTC (2013), pp. 966–980. ISSN: 01419331. DOI: 10.1016/j.micpro.2013.09.001 (cit. on p. 3).

[GSZ18] Di Gao, Tianhao Shen, and Cheng Zhuo. "A design framework for processing-in-memory accelerator". In: *Proceedings of the 20th System Level Interconnect Prediction Workshop on - SLIP '18* (2018), pp. 1–6. DOI: 10.1145/3225209.3225213. URL: http://dl.acm.org/citation.cfm?doid=3225209.3225213 (cit. on p. 4).

[Hen+16] Martijn Hendriks et al. "A blueprint for system-level performance modeling of software-intensive embedded systems". In: *International Journal on Software Tools for Technology Transfer* 18.1 (2016), pp. 21–40. ISSN: 14332787. DOI: 10.1007/s10009-014-0340-3 (cit. on pp. 2, 4).

[KE13] Juergen Karmann and Wolfgang 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 2013* (2013), pp. 45–50. ISSN: 0001-8392. DOI: 10.1109/PATMOS.2013.6662154 (cit. on p. 5).

[LG18] Dongwook Lee and Andreas Gerstlauer. "Learning-Based, Fine-Grain Power Modeling of System-Level Hardware IPs". In: *ACM Transactions on Design Automation of Electronic Systems* 23.3 (2018), pp. 1–25. ISSN: 10844309. DOI: 10.1145/3177865. URL: http://dl.acm.org/citation.cfm?doid=3184476.3177865 (cit. on pp. 3–5).

[LV08] Hugo Lebreton and Pascal Vivet. "Power modeling in SystemC at Transaction Level, application to a DVFS architecture". In: *Proceedings - IEEE Computer Society Annual Symposium on VLSI: Trends in VLSI Technology and Design, ISVLSI 2008* (2008), pp. 463–466. DOI: 10.1109/ISVLSI.2008.71 (cit. on p. 5).

[MJČ18] Dominik Macko, Katarína Jelemenská, and Pavel Čičák. "Simplifying low-power SoC top-down design using the system-level abstraction and the increased automation". In: *Integration* 63.January (2018), pp. 101–114. ISSN: 01679260. DOI: 10.1016/j.vlsi.2018.06.001. URL: https://doi.org/10.1016/j.vlsi.2018.06.001 (cit. on pp. 3, 5).

[MPA12] 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". In: *IET Circuits, Devices & Systems* 6.5 (2012), p. 287. ISSN: 1751858X. DOI: 10.1049/iet-cds.2011.0352. URL: http://digital-library.theiet.org/content/journals/10.1049/iet-cds.2011.0352 (cit. on p. 3).

[Qam+17] Affaq Qamar et al. "LP-HLS: Automatic power-intent generation for high-level synthesis based hardware implementation flow". In: *Microprocessors and Microsystems* 50 (2017), pp. 26–38. ISSN: 01419331. DOI: 10.1016/j.micpro.2017.02.002. URL: https://doi.org/10.1016/j.micpro.2017.02.002 (cit. on p. 5).

[Sen+09] Eric Senn et al. "Power and energy estimations in model-based design". In: *Languages for embedded systems and their applications*. Springer, 2009, pp. 3–26 (cit. on p. 3).

[Xu+12] Yang Xu et al. "A Very Fast and Quasi-accurate Modeling Methodology". In: (2012), pp. 37–49 (cit. on p. 5).