An Efficient Power Estimation Methodology for Complex RISC Processor-based Platforms

Santhosh Kumar Rethinagiri, Rabie Ben Atitallah, Jean-Luc Dekeyser, Smail Niar, Eric Senn

An Efficient Power Estimation Methodology for Complex RISC Processor-based Platforms

Santhosh Kumar
Rethinagiri
Inria Lille - Nord Europe
59650 Villeneuve d'Ascq -
France
santhosh-
kumar.rethinagiri@inria.fr

Rabie Ben Atitallah
Université de Valenciennes et
du Hainaut Cambrésis,
Valenciennes
59313 Valenciennes - France
rabie.benatitallah@univ-
valenciennes.fr

Jean-Luc Dekeyser
Laboratoire d'Informatique
Fondamentale de Lille (LIFL)
Université de Lille 1
59655 Villeneuve d'Ascq
Cédex - France
dekeyser@lifl.fr

Smail Niar
Université de Valenciennes et
du Hainaut Cambrésis,
Valenciennes
59313 Valenciennes - France
smail.niar@univ-
valenciennes.fr

Eric Senn
Université de Bretagne Sud
Centre de Recherche
56321 LORIENT Cédex -
France
eric.senn@univ-ubs.fr

ABSTRACT

In this contribution, we propose an efficient power estimation methodology for complex RISC processor-based platforms. In this methodology, the Functional Level Power Analysis (FLPA) is used to set up generic power models for the different parts of the system. Then, a simulation framework based on virtual platform is developed to evaluate accurately the activities used in the related power models. The combination of the two parts above leads to a heterogeneous power estimation that gives a better trade-off between accuracy and speed. The usefulness and effectiveness of our proposed methodology is validated through ARM9 and ARM CortexA8 processor designed respectively around the OMAP5912 and OMAP3530 boards. This efficiency and the accuracy of our proposed methodology is evaluated by using a variety of basic programs to complete media benchmarks. Estimated power values are compared to real board measurements for the both ARM940T and ARM CortexA8 architectures. Our obtained power estimation results provide less than 3% of error for ARM940T processor, 3.5% for ARM CortexA8 processor-based system and 1x faster compared to the state-of-the-art power estimation tools.

Categories and Subject Descriptors

I.6 [**Simulation and modeling**]: Model validation, analysis

General Terms

System Level Power Estimation

Keywords

Power Estimation, Virtual Platform, SystemC, TLM, FLPA, ASIC

1. INTRODUCTION

Today's embedded industries focus more on manufacturing RISC processor-based platform as they are cost and power effective. On the other side, modern embedded applications are becoming more and more sophisticated and resource demanding. Examples of the concerned applications are numerous such as software defined radio, GPS, mobile applications, etc. The computation requirements of such systems are very important in order to meet real-time constraints and high quality of services. At the same time, the recent advances in silicon technologies offer a tremendous number of transistors integrated on a single chip. For this reason, embedded hardware designers are directed more and more towards complex RISC architectures, which may contain several pipeline slots, hierarchical memory system (L1 and L2 cache level), and specific execution units such as NEON architecture for ARM CortexA8 processor as a promising solution to deal with the potential parallelism inherent from modern applications. Recently, the ITRS [8] and HiPEAC [1] roadmaps promote *power defines performance* and *power is the wall*. In fact, power consumption is becoming a critical pre-design metric in complex embedded systems. An efficient and fast design space exploration (DSE) of such systems needs a set of tools capable of estimating performance and power at higher abstraction level in the design flow. Today, virtual platform power estimation is considered as an important hypothesis to cope with the critical design constraints. However, the development of virtual platform tools for power estimation and optimization

---
[1]http://www.hipeac.net/system/files/hipeacvision.pdf

is in the face of extremely challenging requirements such as the seamless power-aware design methodology.

At the virtual platform level, the power estimation process is centred around two correlated aspects: *the power model granularity* and *the system abstraction level*. The first aspect concerns the granularity of the relevant activities on which the power model relies. It covers a large spectrum that starts from the fine-grain level such as the logic gate switching and stretches out to the coarse-grain level like the hardware component events. In general, fine-grain power estimation yields to a more correlated model with data and to handle technological parameters, which is tedious for virtual platform designers. On the other hand, coarse-grain power models depend on micro-architectural activities that cannot be determined easily. The second aspect involves the abstraction level on which the system is described. It starts from the usual Register Transfer Level (RTL) and extends up to the algorithmic level. In general, going from low to high design level corresponds to more abstract description and then coarser activity granularity. The power evaluation time increases as we go down through the design flow and the accuracy depends on the extraction of each relevant activity and the characterization methodology to evaluate the related power cost. In order to have an efficient power estimation methodology, we should find a better trade-off between these two aspects.

To answer the above challenges, we propose an efficient power estimation methodology for consumption estimation of complex RISC processor-based systems. The idea here is to develop a power estimation virtual platform, which combines Functional Level Power Analysis (FLPA) for hardware power modeling and a system-level simulation technique for rapid prototyping and fast power estimation. The functional power estimation part is coupled with a OVPSim [2] simulator in order to obtain the needed functional-unit activities for the power models, which allows us to reach a superior bargain between accuracy and speed.

This paper is organized as follows. After Section 2 which, presents the related works, Section 3 exposes the proposed power estimation methodology. In Section 4, the power modeling methodology is applied to 2 complex RISC processors designed around OMAP5912 and OMAP3530 boards. To evaluate our methodology in terms of accuracy and speed, experimental results are presented in Section 5.
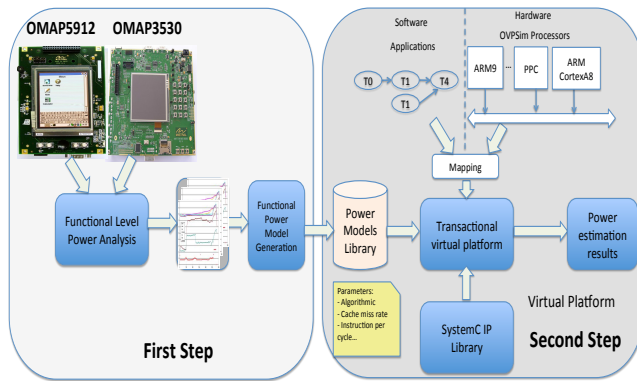

Figure 1: Hybrid power estimation methodology

2. RELATED WORKS

In order to make a better trade-off between power estimation time and accuracy, several studies have proposed evaluating system power consumption at higher abstraction levels. Almost of these tools use a micro-architectural simulators to evaluate system performance and with the help of analytic power models to estimate consumption for each component of the platform. Wattch [3], SimplePower [17] and Skyeye [4] are example of available tools. In general, these tools rely on Cycle-Accurate (CA) simulation technique. Usually, to move from the RTL to the CA level, hardware implementation details are hidden from the processing part of the system, while preserving system behavior at the clock cycle level. The power consumption of the main internal units is estimated using power macro-models, produced from lower-level characterizations. The contributions of the unit activities are calculated and added together during the execution of the program on the cycle-accurate micro-architectural simulator. Though using CA simulators has allowed accurate power estimation, evaluation and simulation time are very significant for the off-the-shelf processor.

In an attempt to reduce simulation time, recent efforts have been done to build up fast simulators using *Transaction Level Modeling* (TLM) [2]. SystemC [14] and its TLM 2.0 kit have become a de facto standard for the system-level description of Systems-on-Chip (SoC) by the means of offering different coding styles. Nevertheless, power estimation at the TLM level is still under research and is not well established. In [12] and [13], a methodology is presented to generate consumption models for peripheral devices at the TLM level. Relevant activities are identified at different levels and granularities. The characterization phase is however done at the gate level: from where they deduce the activity and power consumption for the higher level. Using this approach for recent processors and systems is not realistic. Dhawada et al. [5] proposed a power estimation methodology for a monoprocessor PowerPC and CoreConnect-based system at the TLM level. Their power modeling methodology is based on a fine-grain activity characterization at the gate level, which needs a huge amount of development time. Due to a high correlation with data, a power estimation inaccuracy of 11% is achieved. Compared to the previous works, our proposed methodology for power estimation also partially uses SystemC/TLM simulation with coarse grain power models. Today, the Open Virtual Platform by Imperas Inc. [1] uses the same level of simulation but also tackles the simulation speed problem by proposing the OVPSim simulator which is very fast since processors are not ISS but use code morphing and just-in-time (JIT) compilation. This technique will be also used in our framework.

For the functional level, Tiwari et al. [16] have introduced the concept of Instruction Level Power Analysis (ILPA). They associate a power consumption model with instructions or instruction pairs. The power consumed by a program running on the processor can be estimated using an Instruction Set Simulator (ISS) to extract instruction traces, and then adding up the total cost of the instructions. This approach suffers from the high number of experiments required to obtain the power model. In addition, significant effort is required to obtain the ISS of the target processor. To overcome this drawback the *Functional Level Power Analysis* (FLPA) was proposed [10] [9], which relies on the identification of a set of functional blocks that influence the

power consumption of the target component. The model is represented by a set of analytical functions or a table of consumption values which depend on functional and architectural parameters. Once the model is build, the estimation process consists of extracting the appropriate parameter values from the design, which will be injected into the model to compute the power consumption. Based on this methodology, the tool SoftExplorer [6] has been developed and included in the recent toolbox CAT [15]. It includes a library of power models for simple to complex processors. Only a static analysis of the code, or a rapid profiling is necessary to determine the input parameters for the power models. However, when complex hardware or software components are involved, some parameters may be difficult to determine with precision. This lack of precision may have a non-negligible impact on the final estimation accuracy. In order to refine the value of sensible parameters with a reasonable delay, we propose to couple the OVPSim simulator with the functional level power models which offers us the reasonable trade-off between estimation speed and accuracy.

3. THE HYBRID POWER ESTIMATION METHODOLOGY

This section exposes our proposed power estimation methodology that is divided into two steps as shown in Fig. 1. The **first step** concerns the power model elaboration for the system hardware components. In our framework, the FLPA methodology is used to develop generic power models for different target platforms. The main advantage of this methodology is to obtain power models which rely on the functional parameters of the system with a reduced number of experiments. As explained in the previous section, FLPA comes with few consumption laws, which are associated to the consumption activity values of the main functional blocks of the system. The generated power models have been adapted to the system level, as the required activities can be obtained from the virtual platform. For a given platform, the generation of power models is done at once. To do so, the first part is to divide the architecture of the corresponding processor into different functional blocks and then to cluster the components that are concurrently activated when the code is running.

There are two types of parameters: *algorithmic parameters* that depend on the executed algorithm typically the cache miss or instruction per cycle rates and *architectural parameters* that depend on the component configuration set by the designer typically the clock frequency. For instance, Table 1 presents the common set of parameters of our generic power model. These sets of parameters are defined for a general class of RISC processors. Additional parameters can be identified for specific processors based-architecture such as Superscaler. The next step is the characterization of the embedded system power consumption when the parameters vary. These variations are obtained by using some elementary assembly programs (called scenario) or built in test vectors elaborated to stimulate each block separately. In our work, characterization is performed by measurements on real boards. Finally, a curve fitting of the graphical representation will allow us to determine the power consumption models by regression. The analytical form or a table of values expresses the obtained power models. This power modeling approach was proven to be fast and precise.

The **second step** of the methodology defines the architecture of our power estimator that includes the *functional power estimator* and *fast Just In Time (JIT) compilation simulator* as shown in Fig.1. The functional power estimator evaluates the consumption of the target system with the help of the elaborated power models from the first step. It takes into account the architectural parameters (e.g. the frequency, the processor cache configuration, etc.) and the application mapping. It also requires the different activity values on which the power models rely. In order to collect accurately the needed activity values, the functional power estimator communicates with a fast JIT OVPsim at the TLM level. The combination of the two components above described at different abstraction levels (functional and TLM) leads to a hybrid power estimation that gives a better trade-off between accuracy and speed.

The vital function of the this power estimation methodology is to offer a detailed power analysis by the means of a complete simulation of the application. This process is initiated by the functional power estimator through *mapping interface* (Fig. 2). In this way, the mapping information is transmitted to the fast OVPSim simulator. Our simulator consists of several hardware components which are instantiated from the Open Virtual Platform (OVP) [7] library in order to build a virtual prototype of the target system. We highlight that processors are described using JIT simulator that works by coding morphing concept and is instruction accurate.
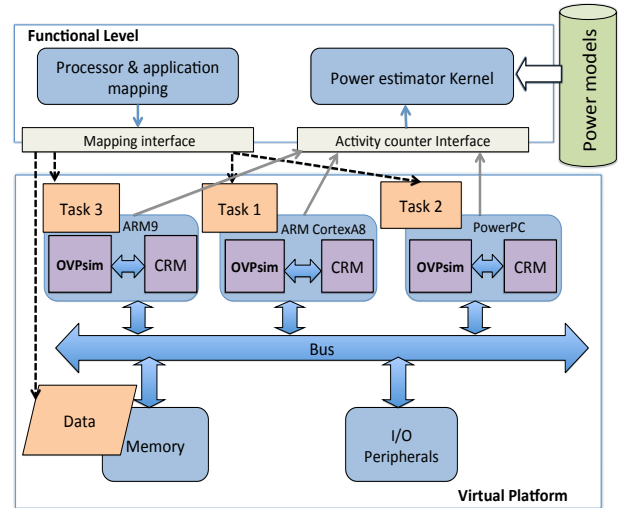

Figure 2: Power estimation methodology functioning

In the power estimation step, the simulator collects the activities that are influenced by the application and the input data. At the end of the simulation, the values of the activities are transmitted to the power consumption models or power estimator kernal using *the activity counter interface* in order to calculate the global power consumption as illustrated in Fig. 2. As we have stated before, the following section will discuss the **first step**; the elaboration of the power model for the OMAP5912 and OMAP 3530 platform by using FLPA methodology.

Table 1: Generic power model parameters

	Name	Description
Algorithmic	τ	External memory access rate
	γ	Cache miss rate
	IPC	Instruction per cycle rate
Architectural	$F_{processor}$	Frequency of the processor
	F_{bus}	Frequency of the bus

Table 2: Generic power models for different processors

Processor	Power models
ARM9	P(mW)=1.03 $F_{Processor}$ + 0.6 (γ) + 5.3
ARM CortexA8	P(mW)=0.79 $F_{Processor}$ + 18.65 IPC +0.26 $(\gamma 1 + \gamma 2)$ + 10.13
PowerPC	P(mW)=4.1 (γ) + 6.3 F_{bus} + 1599

4. POWER MODEL ELABORATION

In order to prove the usefulness and the effectiveness of the proposed power estimation methodology, we used an ARM9 architecture implemented into the OMAP5912 and an ARM CortexA8 -based architectures implemented into the OMAP 3530 platform. The OMAP5912 contains an ARM926EJ-S processor (16KB instruction cache and 8KB data cache). The OMAP3530 contains an ARM Cortex A8 processor (16KB, 2-way set associative instruction and data caches and 256KB L2 cache). Each processor has access to the off-chip memory (SDRAM) via the processor bus interconnect. As explained above, we used the FLPA methodology to generate a power model for each target system. As a first step, we divided the architecture into different functional blocks such as the core clock system, the memory system, and the functional unit for ARM CortexA8 processor as shown in the Fig. 3. A parameter is denoted for each functional block such as $\gamma 1$ and $\gamma 2$ respectively for L1 cache miss rate and L2 cache miss rate. The second step is the characterization of the power model by varying the parameters. These variations are obtained by using some elementary assembly programs (called scenario) or built in test vectors elaborated to stimulate each block separately. In our work, characterization is performed by measurements on real boards. Finally, a curve fitting of the graphical representation will allow us to determine the power consumption laws by regression. The analytical form or a table of values expresses the obtained power laws. This power modeling approach was proven to be fast and precise.
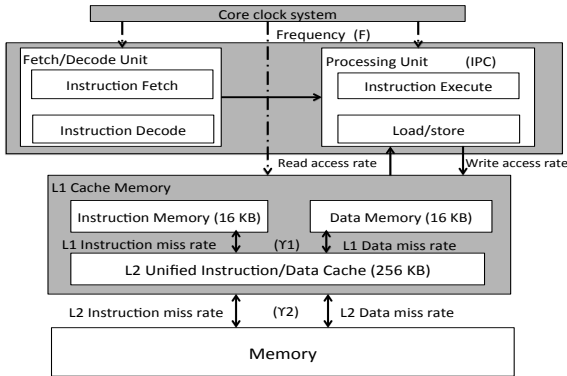

Figure 3: Main functional blocks of ARM CortexA8 processor

Table 2 shows the power consumption models for the ARM9, ARM CortexA8, and PowerPC405 by using FLPA methodology. The input parameters on which the power models rely are the frequency of the processor ($F_{processor}$(MHz)), Instruction Per Cycle (IPC), and the c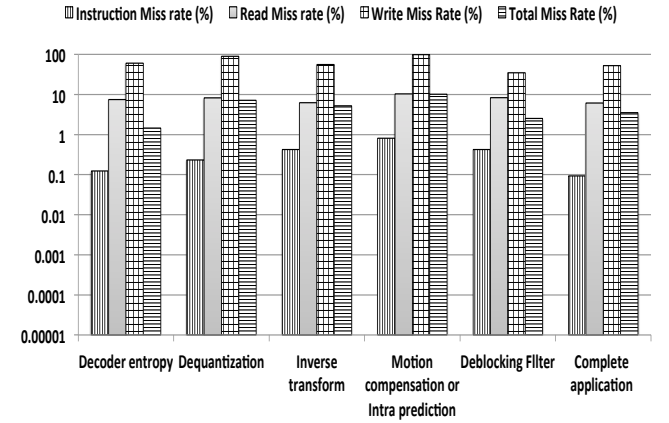ache miss rate ($0 < \gamma < 100$ (%)). The system designer chooses the frequency of the processor and the bus while the cache miss rate and the IPC are considered as an activity of the processor, which could be extracted from the simulation environment.
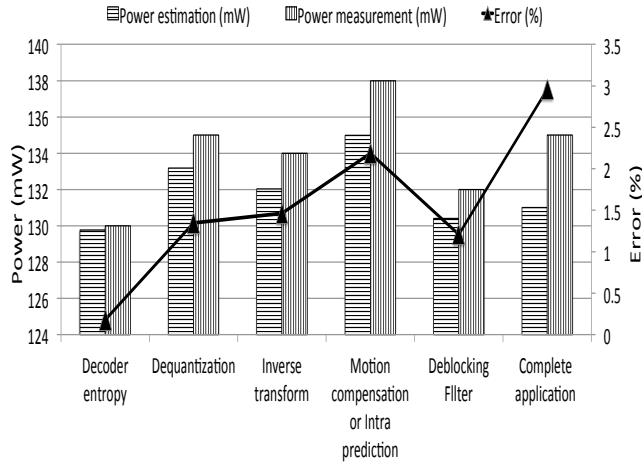
5. SYSTEM LEVEL POWER ESTIMATION RESULTS

5.1 Power estimation accuracy

Figure 4: H264 application cache miss rates for ARM9 processor

For the **second step** of our power estimation methodology, a virtual platform prototype of an ARM9 and an ARM CortexA8 based architecture has been developed. This prototype uses different virtual hardware models, a cache ratio monitor (CRM) provided with the virtual platform for cache miss rate, and the JIT for the target processor as shown in the Fig. 2. Furthermore, the cache parameters and the bus latencies are set to emulate the real platform behaviour. From the CRM, we are able to determine the occurrences of the main activities. For the ARM9 processor the following counters are used for different cache miss rates: read data miss, write data miss and read instruction miss. As a main application, we used the H.264/AVC baseline profile decoder that supports intra and inter-coding, and entropy coding with Context-Adaptive Variable-Length Coding (CAVLC) as a benchmark for ARM9 processor. The H264 decoder application consists of 5 main tasks: decoder entropy, dequantization, inverse transform, motion compensation or intra prediction and deblocking filter.

Fig. 4 shows the detailed results of the activities fetched by the fast JIT-SystemC simulator for each task of the H264 application for ARM9 processor. From these results several

remarks can be drawn. First, we can notice that instruction cache miss rates and read data miss rates are very low when compared with write data miss rates. This is due to the reduced task kernel and data pattern sizes that are very low compared to the cache size (16 KB), which decreases the access to the external memory and thus having a minimal effect on the dynamic power consumption. Second, the data write miss rates have a high impact on the total power consumption of the system. This is because of the algorithm's structure, which does not favour the reuse of data output arrays and the usage of cache policy. Therefore, the statistics collected in Fig. 4 could help in tuning the application structure for a better optimization of the system power consumption. In a similar fashion, we extracted the activities for the ARM CortexA8 processor.
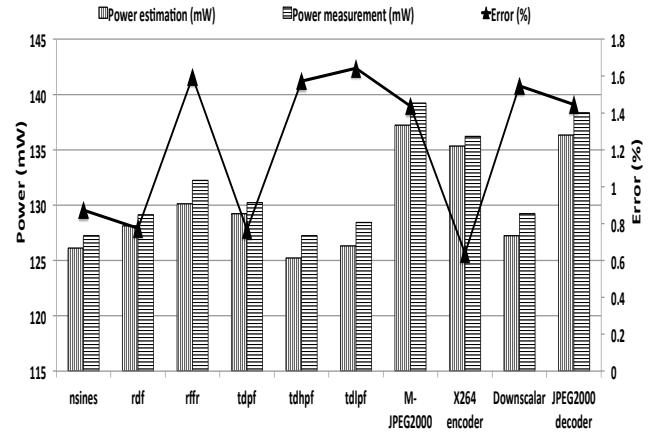

Figure 6: Power estimation accuracy vs real board measurement (ARM9 at 120 MHz)

Figure 7: Power estimation accuracy vs real board measurement (ARM CortexA8 at 500 MHz)

Figure 5: Power estimation accuracy for the H264 application (ARM9 at 120 MHz)
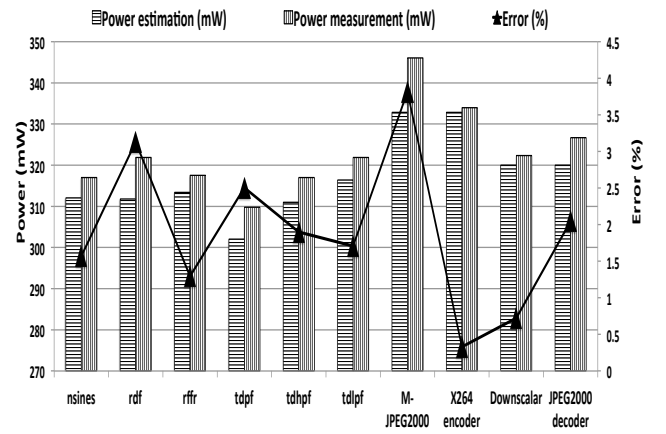
In the next step, we estimated the total power consumption of each task using the power models shown in Table 2. Fig. 5 illustrates the results and shows the comparison between the proposed methodology and the real board measurements. First, our power estimator has a negligible average error of 2%. This study offers a detailed power analysis for each task in order to help designers to detect peaks of consumption and thus to propose efficient mapping or optimization techniques. In order to evaluate the accuracy of our methodology, we carried out power estimation on several image & signal processing benchmarks. Fig. 6 and Fig. 7 illustrates the power results by showing the estimation accuracy between the proposed power estimation methodology and the real board measurements. Our proposed methodology has an negligible average error of 1.24% and 2.4% respectively for ARM9 and ARM Cortex A8 processor, which is considered as a high accuracy level when compared to SoftExplorer's [11] average error of 8% for RISC processors.

5.2 Power estimation speed

In this section, we will compare the efficiency of the proposed methodology in term of estimation speed with the SimplePower(Cycle-Accurate), TLM with ISS based simulation and SoftExplorer (functional level simulator) ap-

proaches as introduced in Section 2. This comparison is for the quantification of our proposed methodology to the state-of-art power estimation tools used in current industrial and academic practices. SoftExplorer, TLM with based simulation, and our proposed methodology are executed on a PC (Intel, 1.8 GHz, 2 Go of RAM), whereas SimplePower on a Workstation (Ultra Sparc T2+, 1.6 GHz, 2 Go of RAM). In order to compare the result, computer benchmarking has been done to confirm that the workstation is always faster compared to the PC for all kind of applications. Power estimation has been carried out with a set of image & signal processing applications and also with SPEC 2008 [3] benchmarks.

From Fig. 8, we can notice that SoftExplorer and TLM with ISS based simulation have an average estimation time of 5 seconds, which is faster when compared to the SimplePower's average estimation time of 20 seconds. Our proposed methodology has a average estimation time of 2.45 seconds, which is faster compared to the other tools. Our methodology works by running the application on the virtual

---

[3]http://http://www.spec.org/benchmarks.htmlpower/

Figure 8: Comparison of estimation time for our proposed methodology, SimplePower, TLM with ISS based simulation and SoftExplorer tools

platform thereby collecting the dynamic activities. Simple-Power uses cycle accurate specifications to collect the necessary power data, whereas SoftExplorer realizes a static profiling of the code, which results in reduced execution time and thus resulting in a low power consumption estimation time. Static profiling of the C code is not sufficient to determine the average execution time and the global energy consumption, for this reason we need to run the application on the virtual platform in order to collect the activities accurately and efficiently. Experimental results prove that our proposed methodology is efficient, fast and accurate.

6. CONCLUSIONS

This paper presents a efficient system level power estimation methodology for RISC processor-based embedded system. Indeed, power/energy constraints are considered as a major challenge when the system runs on batteries. Thus, designers must take these constraints into account as early as possible in the design flow. First, a power modeling methodology has been defined to address the global system consumption that includes core clock system, memory, etc. Secondly, the functional power modeling part is coupled with a fast virtual platform to obtain the needed micro-architectural activities for the power models, which allows us to reach accurate estimates. With such proposed methodology, the designer can explore several implementation choices on different processors. The future works of this project will focus on more complex heterogeneous platforms. Furthermore, in order to obtain more accurate power estimations, some power model refinements must be realized.

7. REFERENCES

[1] B. Bailey. System level virtual prototyping becomes a reality with OVP donation from imperas. Technical report, EDA, June 2008.

[2] G. Beltrame, L. Fossati, and D. Sciuto. ReSP: A Nonintrusive Transaction-Level Reflective MPSoC Simulation Platform for Design Space Exploration. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 28(12):1857–1869, Dec. 2009.

[3] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. In *Proc. International Symposium on Computer Architecture ISCA'00*, pages 83–94, 2000.

[4] Y. Chen. *The Analysis and Practice on Open Source Embedded System Software–Based on SkyEye and ARM Developing Platform*. Beihang University Press, 2004.

[5] N. Dhanwada, R. A. Bergamaschi, W. W. Dungan, I. Nair, P. Gramann, W. E. Dougherty, and I.-C. Lin. Transaction-level modeling for architectural and power analysis of powerpc and coreconnect-based systems.

[6] S. Dhouib, J.-P. Diguet, D. Blouin, and J. Laurent. Energy and power consumption estimation for embedded applications and operating systems. *Journal of Low Power Electronics (JOLPE)*, 5(3), 2009.

[7] Imperas Inc. OVP World home page. http://www.ovpworld.org/.

[8] ITRS. Design, 2010 edition. http://public.itrs.net/, 2010.

[9] J. Laurent, N. Julien, and E. Martin. High level energy estimation for DSP systems. In *Proc. Int. Workshop on Power And Timing Modeling, Optimization and Simulation PATMOS'01*, pages 311–316, 2001.

[10] J. Laurent, N. Julien, and E. Martin. Functional level power analysis: An efficient approach for modeling the power consumption of complex processors. In *Proceedings of the Design, Automation and Test in Europe Conference*, Munich, 2004.

[11] J. Laurent, N. Julien, and E. Martin. Softexplorer: estimation, characterization and optimization of the power and energy consumption at the algorithmic level. In *Fourteenth International Workshop on Power and Timing Modeling (PATMOS 2004)*, pages 15–17, Santorini, Greece, September 2004.

[12] I. Lee, H. Kim, P. Yang, S. Yoo, E. Chung, K.Choi, J.Kong, and S.Eo. Powervip: Soc power estimation framework at transaction level. In *Proc. ASP-DAC*, 2006.

[13] N.Dhanwada, I. Lin, and V.Narayanan. A power estimation methodology for systemc transaction level models. In *International conference on Hardware/software codesign and system synthesis*, 2005.

[14] Open SystemC Initiative. Systemc, 2008. World Wide Web document, URL: http://www.systemc.org/.

[15] J. D. S. Douhib. Model driven high-level power estimation of embedded operating systems communication and synchronization services. In *Proceedings of the 6th IEEE International Conference on Embedded Software and Systems*, China, May 25-27 2009.

[16] V. Tiwari, S. Malik, and A. Wolfe. Power analysis of embedded software: A first step towards software power minimization. In *Transactions on VLSI Systems*, 1994.

[17] W. Ye, N. Vijaykrishnam, M. Kandemir, and M. Irwin. The design and use of simplepower: a cycle accurate energy estimation tool. In *Proc. Design Automation Conference DAC'00*, June 2000.