Functional Level Power Analysis: An Efficient Approach for Modeling the Power Consumption of Complex Processors

Johann Laurent, Nathalie Julien, Eric Senn, Eric Martin
LESTER Laboratory
South Brittany University, France
firstname.name@univ-ubs.fr

Abstract

A high-level consumption estimation methodology and its associated tool, SoftExplorer, are presented. The estimation methodology uses a functional modeling of the processor combined with a parametric model to allow the designer to estimate the power consumption when the embedded software is executed on the target. SoftExplorer uses as input the assembly code generated by the compiler; its efficiency is compared to SimplePower's approach. Results for different processors (TI C62, C67, C55 and ARM7) and for several DSP applications provide an average error less than 5%.

1. Introduction

The Systems-On-Chip must respect critical constraints in time, area and consumption; therefore the designer need to validate his choices early in the flow, and has to characterize the system improvements through accurate estimates. As the software part is growing in the applications, its impact on the consumption is also becoming more important. Although many researches have developed methodologies to estimate the software consumption, few of these are associated with an estimation tool. Some tools use a fine-grain representation of the processor to obtain a cycle-accurate estimation: for example SimplePower [1] need a RTL representation of the architecture to allow the power characterization. This low-level modeling often implies that the time required to characterize a processor and to estimate an application becomes too important to give an efficient feedback in system design. Another method, the Instruction Level Power Analysis (ILPA) consists in estimating the cost of each instruction of the assembly code [2]; the estimation time is strongly reduced but elaborating the processor model can be very time consuming for complex processors. JouleTrack is the only known tool based on this approach [3]. We propose here to increase the abstraction level, by combining a functional level model of the processor requiring only a coarse-grain knowledge on its architecture and a parametrical model of the code; the aim is to obtain a good tradeoff between the estimation accuracy and the model complexity.

2. Functional Level Power Analysis

To estimate the power/energy consumption of an application executed on a processor, we first need to realize the power characterization of the target. The Functional Level Power Analysis (FLPA) methodology can be started from only a simple block diagram of the architecture. As sketched on Fig. 1, the first step consists in dividing the processor architecture into different functional blocks and sub-blocks, to cluster the components that are concurrently activated when a code is running.

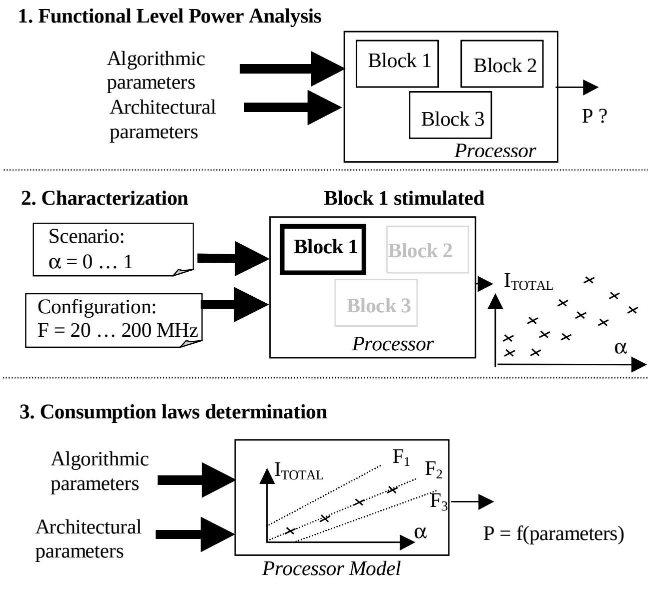

Figure 1. Processor Modeling Methodology

Then, the relevant consumption parameters are selected as the significant links between these blocks. There are two types of parameter: algorithmic parameter values depend on the executed algorithm (typically the cache miss rate) and architectural parameter values depend on the processor configuration settled by the designer (typically the clock frequency).

The second step is the characterization of the processor power consumption when the parameters vary. These variations are obtained by using some elementary assembly programs (called scenario) elaborated to stimulate each block or sub-block separately. Characterization can be performed either by measurements or by simulation; for our part, the supply current is measured on evaluation boards. Finally, a curve fitting of the graphical representation allows determining the consumption laws by regression. More information about our methode can be found in [4].
Table 1 presents the SoftExplorer power estimation results in Watt and the associated error between the physical measures and the estimates. For more details, this tool will be in demonstration at the DATE’04 University Booth.

Table 1. SoftExplorer power estimation results

		PROCESSORS			
		C55	**C62**	**C67**	**ARM7**
DWT	P (W)	0.39	3.83	0.9	0.22
	Error	*+2.3%*	*+2%*	*+6%*	*+7%*
EFR	P (W)	0.43	2.62	0.95	0.22
	Error	*+2.1%*	*-0.6%*	*-1%*	*+3%*
MPEG-1	P (W)	0.4	5.59	0.93	0.22
	Error	*-1.6%*	*-4%*	*2.4%*	*+8%*
Maximum Error		*2.3%*	*-4%*	*6%*	*+8%*

3. SoftExplorer/SimplePower Comparison

We will compare the efficiency of our method with the SimplePower’s approach on the SPEC-95 benchmarks (provided with SimplePower) and on other signal-processing applications.

SimplePower has been executed on a Ultra Sparc III+ (900MHz, 1Go of RAM) and SoftExplorer on a PC (Athlon 1GHz, 256Mo of RAM); a first computer benchmarking has been achieved to confirm that the workstation is always faster than the PC for every type of application.

Afterwards, power estimation has been performed on SPEC-95 benchmarks and typical signal and image processing applications (DSP) with both SimplePower and SoftExplorer. As the modeled processors are different (a modified MIPS-IV for SimplePower and TI C62, C67, C55 and ARM7 for SoftExplorer), it is not relevant to compare the estimates. Nevertheless, another indicator of the tool efficiency is the estimation time, corresponding to the total time necessary to compile, to profile and to estimate the benchmark consumption; results are proposed in Table 2.

SoftExplorer realizes a static trace of the application whereas SimplePower realizes a dynamic trace, increasing the estimation time when the execution time is high. Furthermore, it provides to the designer a more detailed information, cycle by cycle, allowing fine-grain code optimizations. Therefore, SoftExplorer is a power estimation tool intended to the system design assistance; it realizes the suitable tradeoff between the estimation accuracy and time in order to ensure a rapid and reliable feedback to the designer.

Table 2. SimplePower/SoftExplorer estimation time

		SimplePower	SoftExplorer
SPEC-95	Bubble	35s	4s
	Matmult	1s	2s
	Perm	67s	4s
	Quick	6s	3s
DSP	FFT 1024	17s	3s
	LMS 1024	24728s	4s
	DWT 512x512	142267s	10s
	MPEG-1	10s	8s

4. Conclusion

A high-level estimation methodology and the associated tool SoftExplorer have been presented. The processor is modeled through a functional analysis and the software model is parametric: algorithmic parameters, depending on the code execution, are determined by the estimation process. The average error of the SoftExplorer results against the physical measurements is about 2.4% (except for the ARM7 with 5%). The accuracy and the rapidity of the tool make it convenient to give metrics in the Co-Design step, to choose both the target and the suitable algorithm and to return to the designer a reliable indication about algorithm optimizations. Future works will add other processor models to our library and further comparisons will be performed with existing tools and methods.

5. References

[1] W. Ye, N. Vijaykrishnan, M. Kandemir, M.J. Irwin “The Design and Use of SimplePower: A Cycle Accurate Energy Estimation Tool,” in *Proc. Design Automation Conf.*, June 2000, pp. 340-345.

[2] V. Tiwari, S. Malik, A. Wolfe “Power analysis of embedded software: afirst step towards software power minimization.” IEEE Trans. on VLSI Systems, vol.2, N°4, Dec 1994, pp. 591-593.

[3] A. Sinha, A. P. Chandrakasan "JouleTrack - A Web Based Tool for Software Energy Profiling", in *Proc. DAC*, June 2001, p220

[4] N. Julien, J. Laurent, E. Senn, E. Martin “Power Consumption Modeling and Characterization of the TI C6201”, in IEEE Micro Sept/Oct 2003, p40.