Modeling Power Dynamics for an Embedded DSP Processor Core

An Empirical Model

C.H.Gebotys and R.Muresan

Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L3G1

Abstract: A new model for dynamic current analysis and simulation is presented for power and energy analysis of a complex VLIW DSP processor core, targeting secure wireless communications. Unlike other research, an instruction level RC based model, whose input parameters can be extracted from the DSP core's assembly level program, is introduced for power simulation. Experimental results utilizing several benchmark cryptographic applications show that the model can accurately simulate power variation of a program. For the first time results are verified with real power traces of a DSP processor core in a VLSI chip running cryptographic applications with an average error in energy estimation of 7%. This research is important for analyzing the impact of software on power and the design of embedded cryptographic VLSI systems that are safe from power attacks.

Key words: Power, DSP, Current, Model, Energy

1. INTRODUCTION

Power dissipation has a large impact on cost, reliability and security of an embedded system. Previous literature has discussed the first two issues at length, however only recently with the widespread use of wireless communications has interest in security of embedded system designs increased. Typically embedded systems involve the design of low power, low cost, high performance algorithms supporting wireless communication

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35597-9_40](https://doi.org/10.1007/978-0-387-35597-9_40)

M. Robert et al. (eds.), *SOC Design Methodologies*

© IFIP International Federation for Information Processing 2002

systems with audio and video types of data. The encryption of audio and video data before transmission is very important. Not only must encryption algorithms be high performance and low power, but more importantly they must be secure or safe from power attacks. For example, power attacks involve utilizing power traces (or dynamic power measurements over time) of a device to determine the users secret key. The dynamic power trace must be simulated during the design of the encryption algorithm to make sure the embedded system is secure. Unfortunately most power simulation is performed at the transistor or module level which is too slow for complex processors. This paper for the first time introduces a high level instruction based dynamic power simulation model which is targeted towards encryption algorithms, yet can be also applied to any general DSP application to simulate power.

DSP processors are utilized in many embedded systems in particular for wireless communication applications, due to their low cost, low power dissipation and high performance. Recently advances in DSP cores have provided highly parallel VLIW (very long instruction word) based processor cores, such as the SC140 Star Core processor developed by Motorola and Lucent Technology [1]. This processor core has four functional units, two address arithmetic units, and one bit mask unit [1]. The SC140 core processor is based on a Variable Length Execution Set model [1]. Like VLIW processors, a group of instructions, called an execution set, can be executed in parallel. An atomic operation within an execution set is encoded by an individual instruction. An eight-word instruction set, called a fetch set, is fetched from memory every clock cycle and the processor detects the portion of this set that can be executed in parallel [1]. Based on grouping constraints, an execution set may vary from one to six atomic instructions [1]. The register file blocks within the SC140 DSP core are divided into two banks each containing eight 40bit registers. Based on the instruction types and the usage of the upper bank registers within an execution set the instructions are serially grouped or prefix grouped by the assembler [1]. Unlike other DSP processor cores, the SC140 can load or store eight sixteen bit words per cycle (providing a high data memory bandwidth[2]). In this paper we introduce a methodology for developing an empirical instruction-based model which is capable of estimating dynamic current, power or energy consumption for a DSP processor core. The methodology for creating the dynamic power model generally can be used with most processors, however it is illustrated in this paper with the SC140 DSP processor core.

2. PREVIOUS RESEARCH AND PROBLEM DEFINITION

A critical problem in wireless communication applications is the battery life time. The complexity of the targeted applications for the DSP processors makes low power designs very difficult. Transistor and module level power models have been developed [3] as well as software techniques [6]. Recent research work is directed towards finding power consumption modeling techniques at the software level which will benefit the compilers for these DSP processors [5,7,11]. Power modeling at the instruction level for processors were investigated in [4,7]. However dynamic power simulation is not supported and only static single average power values are returned for a given software program running on a processor. Furthermore few researchers have examined complex VLIW processors where more than one instruction execute in parallel. Dynamic power traces have been used in power-attacks of cryptographic devices. In particular the analysis of the variation of power, and computations on a number of power traces can be used to detect data and algorithmic dependencies. For example this information could be used to detect the secret key of a smart card, thus performing a power-attack. Currently power attacks of cryptographic devices, have been analyzed using slower clock frequency, non-VLIW processors in non-time critical applications, such as smart cards [8]. Future wireless communications devices such as encryption of voice and video will be time-critical and require VLIW processor implementation. Power attacks of more sophisticated processors with parallel instruction execution have not been reported in the literature. Thus RC circuit type models of power at the instruction level are new for the VLIW DSP processors. This paper will present a methodology for modeling dynamic power simulation for a complex VLIW DSP processor core, the SC140. An instruction level model based on RCs is developed and verified with real current measurements[9] of the DSP hardware VLSI core in a chip.

Given a complex VLIW DSP processor, the following questions are important for the design and analysis of power efficient applications:

1. Is there a relationship between the activity involved by a certain type of instruction, data handled by the instruction and the current consumption of the DSP processor?
2. Can we predict at any given time the current consumption behavior of an application based on the information received at assembly language level (or higher level) without knowing the detailed structure of the DSP processor and all the computational data handled by the application?

3. Is there a possibility of finding a general standard procedure of modeling the current behavior for an application at software level?

From extensive analysis of current dynamics of individual assembly language instructions and assembly language programs for the SC140 DSP processor[9,12] this paper will show that these questions can be answered positively. The methodology used to study these questions can also in general be applied to many other DSP processors.

3. METHODOLOGY

The following section will outline the empirical methodology for developing a dynamic power model for a processor core. The SC140 core processor is a pipelined processor with five stages and parallel execution capabilities [1]. The parallel capabilities are given by the core's hardware configuration and by the type of instructions executed. The maximum number of atomic instructions executed in parallel in an execution set is six. In Fig.1 an RC model is shown, representing the execution set level of the processor. Each atomic instruction present in an execution set has a parallel RC module component in the total current draw for the execution set. For example if two type 1 instructions [1] are present in an execution set then two RC modules have their contacts closed. The instantaneous current for an execution set is given by the sum of the instantaneous currents of the individual atomic instructions that are part of the execution set. In this model there is an RC module activated by the presence of a high bank register in any of the atomic instructions of the execution set. Waveform analysis performed in this research indicated that this component is mainly dependent on the internal activity involved by using a high bank register and not on the number of high bank registers used in the execution set or on the fact that the execution set has a 2-word prefix. In a processor at any given time there is intense switching activity at the transistor level. The number of transistors involved in the switching activity is proportional with the amount of activity required by the instruction that is executing. The SC140 core pipeline has a pre-fetch stage, fetch stage, dispatch stage, address generation stage and execution stage [1].

Figure 1. RC Model at the execution set level

Figure 2. Program level RC model

Typically the most active stage of an instruction in a pipelined processor is the execution stage. As a result in each clock cycle the current draw of the processor will have a predominant component described by the executing instruction at that time. Considering that the total current draw at any given time is proportional with the number of transistors activated, then each RC module of Fig.1 can be considered to be formed by a large number of RC elements that work in parallel, and each RC element models one transistor. The fact that the total number of transistors activated at any given time is variable gives the variable component for the R and C elements of Fig. 1.

Based on extensive current measurements performed at the instruction level on different instructions executing on different data and it was deduced that

the number of transistors active during the instruction's life time could be modeled by a distribution known as the gamma function[10], in Fig.3 :

$$g(x; n, \lambda) = \frac{\lambda(\lambda x)^n}{n!} e^{-\lambda x}; x \geq 0, \lambda > 0$$

In the equation above, n is the order of the gamma function and characterizes the shape of the waveform. We found that the instantaneous current value of each instruction for the SC140 core processor can be approximated by: $i(t) = K * g(x; n, \lambda)$ where λ , is dependent only on the clock frequency and K depends mainly on the type of the instruction executed (and could also characterize the data handled by the instruction).

Figure 3. The gamma function

An important observation which made possible the RC model development at the software level is that the waveform shape is not dependent on the data handled by the instructions and during the execution time of any execution set the instantaneous current waveform can be approximated using the linear superposition principle by the following sum:

$$RC_{es} = \sum_{i=1}^8 k_i g_i(t - t_{start} - t_{delay}^{(i)}; n, \lambda)$$

where t_{start} is the time when the execution set is pre-fetched, and $t_{delay}^{(i)}$ is a delay time which is present for *move* type instructions due to the fact that these instructions do not fully complete their operation in the last stage of the pipeline. Also $g_i()$ is the gamma function for the i 'th RC module of Fig.2 (i 'th instruction in the execution set, es) and k_i is the K coefficient for the RC module i . Good instruction and program approximations have been obtained for values of n equal with 1 and 2. If one of the modules of Fig.2 is not active then the corresponding K coefficient is zero. For most of the instructions in an execution set t_{start} is identical and $t_{delay}^{(i)}$ can be considered zero. Now let us consider an assembly language program that starts to

execute its first execution set at time t_0 . At run time we assume that the program executes a total number of execution sets equal with $(n+1)$ at a rate of one execution set per clock cycle. Also let us consider that $t_{delay} = 0$ for all the instructions and the clock cycle period is T . The equation for the m 's execution set becomes:

$$RC_{es}(t) = g(t - (t_{start} + mT); n, \lambda) \sum_{i=1}^8 k_i$$

$$= K_m * RC_p^{(m)}(t)$$

Figure 4. Gamma functions summed to model for power trace.

The general RC circuit for the above program is represented in Fig.3 , where the contact for each execution set ES_m ($m \in \{0, 1, 2, \dots, n\}$) is closed at time $t_0 + (m-1)T$ and remains closed until the execution stops. Evaluating the total current draw for the RC circuit of Fig.3 as a sum of the individual currents per execution set, we can develop the following general formula, given below and illustrated in Fig.4, for calculating the instantaneous current draw at the software level:

$$i(t) = I_0 + \sum_{m=0}^n K_m * RC_m(t - t_0 - mT)$$

where

$$RC_p^{(m)}(t) = \begin{cases} 0; & \forall t < t_0 + mT \\ K_m * 0.621 * g(x; 1, 0.0041); & x = t - (t_0 + mT); \end{cases}$$

4. EXPERIMENTAL RESULTS

The experimental setup for dynamic power measurements and results of the gamma model will be outlined in this section. The models were generated using MATLAB. A number of cryptographic applications were used to illustrate the gamma model and energy dissipation accuracy was quantized. The following procedure was used to acquire the instruction dynamic power models: Dynamic current waveform measurements were performed for each individual instruction operating on a mixed data values using experimental setup shown in Fig.5[9] using synchronized clock and interrupt scheme shown in Fig.6[9]. The instantaneous current waveforms were captured for a block of 8 identical execution sets (where each execution set was the instruction under measurement, or IUM). The peak to peak variation of the measured current was recorded as $P_k(IUM)$. Using the $P_k(IUM)$ value we generated the individual k_i coefficients for each instruction of the SC140 assembly language as $k_i = P_k(IUM)/8$. These coefficients were then used in the RC model for SC140 assembly language applications. The dynamic current waveforms measured and their models will be outlined next, followed by complete models and measured current for some cryptographic application programs.

Figure 5. Experimental Set up.

Figure 6. Synchronization of clock and interrupts

Fig. 7 shows three waveforms representing from highest to lowest amplitudes the three instructions: *CLR Dn* (sets a register to zero) , *DOSETUPn label* (sets up for a loop), and *DOENN #u6* (loads control register with a number). The measured power trace (thick ‘noisy’ waveform) is superimposed with the gamma function (thinner line waveform, specifically $RC_p^{(m)}(t)$) in the figure for each instruction.

Figure 7. Power traces and gamma functions for three instructions.

Figure 8. Power traces and gamma functions for load and exclusive or instructions.

Figure 8 shows the measured power traces and superimposed gamma models for 3 separate types of instructions: *MOVE.2L (EA),Da:Db*, *MOVE.L #s32,C4*, and *EOR Da,Dn* in order of highest to lowest amplitudes. The first instruction loads two 32bit words into two data registers. The second instruction loads an absolute 32bit value into a control register. The lowest current draw was obtained from the exclusive or on two registers. For plotting purposes in Fig. 8 the amplitudes for the exclusive or instruction (*EOR*) were multiplied by a factor of 2.

Fig. 9 presents the measured current and superimposed gamma model for a SC140 DSP cryptography application which runs for 160 clock cycles. The correlation of the measured current and the gamma model is 0.989 indicating the model is very good. The estimated energy of the gamma model has an error of 7%. Fig. 10 presents the superimposed waveforms for the same functional program but with a current variation generated by inserting a block of 40 NOPs (or no operation instructions), one per execution set. The purpose of this insertion was to verify that the gamma model could reproduce the current variation generated by the low current consumption instructions (NOPs). Both Fig. 9 and Fig. 10 were created using MATLAB. The clock frequency of the SC140 processor during the measurements was stable at 100MHz. As it can be seen the approximate current waveform is very close to the real current waveform and the current variation is faithfully reproduced by the model.

Figure 9. Program 1 power trace versus model.

Figure 10. Program 2 power trace versus model.

5. DISCUSSION AND CONCLUSIONS

This study presents for the first time an instruction level model for dynamic power simulation of a complex VLIW DSP processor core. Unlike previous research, power traces have been verified with real hardware VLSI chip power measurements. Energy estimates are accurate to 7%. This working modeling technique for the SC140 DSP processor can in general be applied to many other DSP processors. The importance of this modeling technique is that it can provide instantaneous current, power or energy information at the

software level without performing an actual power measurement for the application. As a result, analysis of power dynamics is supported with optimization goals or with goals of preventing cryptographic power attacks, thus supporting a new dimension to optimization, namely security. This research is crucial for supporting a methodology for designing software that is not only optimized for performance, power and cost, but also optimized for security, and supporting algorithmic modifications such that the desired current waveform shape or amplitude is generated. The k_i coefficients can be calculated one time for the whole processor's instruction set. These coefficients can be later used for analyzing the power consumption of any application regardless of the program's memory storage location, data handled by the individual instructions or the registers used by the individual instructions. Future research intends to apply these modeling techniques to develop optimized and secure communication applications targeting DSP processors. This research was supported in part by grants from NSERC, Motorola, and CITO.

6. REFERENCES

- [1] Alliance Group, Motorola and Lucent Technologies, SC140 DSP Core Reference Manual, <http://www.starcore-dsp.com/docs>, 1999.
- [2] C.H.Gebotys, "A Network Flow Approach to Memory Bandwidth Utilization in Embedded DSP core Processors", to appear IEEE Transactions on VLSI, 2002.
- [3] A. Chandrakasan, R. Broderon, Low Power Digital CMOS Design, Kluwer, 1995.
- [4] V.Tiwari, S.Malik, A.Wolfe, "Power Analysis of Embedded Software: A First Step Towards Software Power Minimization", IEEE Transactions on VLSI, vol2, 4, pp.437-445, Dec1994.
- [5] V.Tiwari, S.Malik, A.Wolfe, "Compilation Techniques for Low Energy", Int'l Symp. On Low Power Electr. Des., 1994.
- [6] V.Tiwari, D.Singh, S.Rajgopal, G.Mehta, R.Patel, F.Baez, "Reducing Power in High-performance Microprocessors", 35th Design Automation Conference, June 1998.
- [7] G.Qu, N. Kawabe, K. Usami, M. Potkjonak, "Function-level power estimation methodology for Microprocessors", Proc. of Design Automation Conf., 2000, pp810-813.
- [8] P.Kocher, J.Jaffe, B.Jun, "Differential power analysis", LNCS, vol 1666, Springer-Verlag, 1999, pp.388-397.
- [9] R.Muresan, C.Gebotys, "Current Consumption Dynamics at Instruction and Program Level for a VLES DSP Processor", Proc. of 14th Int'l Symp. On Sys. Level Synthesis, Montrl, Can, 2001, pp.130-135.
- [10] G. Hadley, "Introduction to Probability and Statistical Decision Theory", HOLDEN-DAY, Inc., 1967.
- [11] C.H.Gebotys, R.Gebotys, "Statistically Based Prediction of Power Dissipation for Complex Embedded DSP Processors", Microprocessors and Microsystems, Vol23, May1999, pp135-144.
- [12] R.Muresan, C.Gebotys "Dynamic Power Simulation Model for VLIW DSP Processor VLSI Cores with Secure Applications", IFIP Proceedings of 11th Int'l Conf. On VLSI SOC, Montpellier, Fr., 2001.pp67.-72.