

Developing Energy-Aware Strategies for the Blackfin Processor

Steven VanderSanden, David R. Kaeli
Dept. Electrical and Computer Engineering
Northeastern University
Boston, MA 02115 USA
{svanders,kaeli}@ece.neu.edu

Giuseppe Olivadoti, Richard Gentile
Analog Devices, Inc.
1 Technology Way
Norwood, MA 02062 USA
{giuseppe.olivadoti,richard.gentile}@analog.com

Abstract

Energy usage is becoming an increasingly important design constraint for all computer systems. This issue is particularly critical in battery powered, embedded designs. Although many embedded processors have developed sophisticated power management schemes, few have produced an accurate, easy-to-use energy estimation framework. In this presentation we will describe the development of an instruction-level energy modeling framework for the Analog Devices Blackfin family of processors. Using this model, we are able to accurately estimate the energy consumed when running this code. While our main goal is to demonstrate that we can perform accurate energy estimation, we also plan to develop a framework that is fully integrated with compilation in order to produce more energy-efficient binaries. In this abstract we briefly describe our methodology and show data that illustrate some of the difficulties encountered when attempting to statically model energy.

1 Introduction and Methodology

The design specifications of many embedded systems include strict energy budgets. In order to reduce the time to market (and still meet these constraints), a designer must be able to accurately predict the energy usage for the system. The goal of our work is to develop an energy estimation scheme for the Analog Devices Blackfin 533 (ADSP-BF533). Two of the more common modeling options employed for energy estimation are architectural-level and instruction-level estimation. Architectural-level tools, which include the Wattch [1] and SimplePower [2] power modeling frameworks, compute energy based on functional unit usage considering transitions of individual signals. Instruction-level tools calculate the energy budget by characterizing individual instructions and inter-instruction energy usage. Instruction-level tools can only be used when the microarchitecture of the underlying processor is simple (e.g., on embedded cores).

We have chosen to use instruction-level energy estimation for our work. This form of estimation was employed previously by Tiwari et al. at Princeton to develop models for a number of embedded processors [3, 4]. They developed accurate models for the Intel 486DX2 and the Fujitsu SPARClite. We are following a similar approach, but extending it to consider further power aspects of the microarchitecture and applying these extension to the ADSP-BF533.

As mentioned previously, an instruction-level estimation is constructed by characterizing the energy usage of individual instructions (i.e., base energy cost) and then computing the overhead that is incurred when two different instructions are executed consecutively (inter-instruction effects). The total energy for a program is computed by summing the base energy costs of the individual instructions and the total inter-instruction effects.

To capture the base energy cost for an instruction, we place several instances of that instruction in a loop, run the loop, and measure the average current produced. The base energy cost is directly proportional to this measured current multiplied by the number of cycles required for the execution of one instance of the instruction. Inter-instruction effects are those effects that cannot be captured in the base energy cost. Inter-instruction effects can be characterized as effects related to resource constraints and delays (e.g., pipeline stalls, cache misses, write buffer stalls, etc.) and circuit state overhead (the added cost of switching within the circuit when executing two different instructions in succession). The circuit state overhead can be measured by placing many repetitions of a pair of instructions in a loop and measuring the average current. The inter-instruction overhead can be calculated by computing the difference between the measured current and the average of the two base costs of the instructions in the loop.

Instruction r7 = r3 + r4;		
r3 Value	r4 Value	Current(mA)
0x1	0x1	52.20
0x80000000	0x80000000	52.31
0x90B	0x371F	52.82
0xCCCCCCC	0xCCCCCCC	53.27
0x33333333	0x33333333	53.33
0xFFFF	0xFFFF	53.44
0x7FFFFFFF	0x7FFFFFFF	54.34
0xFFFFFFFF	0xFFFFFFFF	54.34

Table 1: Impact of data operand values

Instruction	Initial Values	Current (mA)
r6 = -r3;	r3 = 0x90B	51.98
r3 = -r3;	r3 = 0x90B	60.53

Table 2: Impact of toggling register values

To estimate the total energy consumed by various programs, the base energy cost must be measure for each instrucion in the instruction set and the circuit state overhead for a large number of instruction pairs needs to be computed. The estimation framework will use a table lookup strategy to sum the base energy costs and the inter-instruction effects of a program to estimate the energy usage. To reduce the amount of time to produce this tables, we can identify similaries in energy costs of similar instructions.

The goal of his work is to produce an instruction-level energy model for the ADSP-BF533 and to use that model as a base for an energy estimation framework. In addition to these goals, we are also trying to improve the methods currently used for energy profiling. The remainder of this paper will discuss some of the issues encountered during energy profiling and we will also provide an example result used to verify our approach.

2 Results

In this section we will show some examples of the measurements that need to be obtained to perform instruction level modeling. In addition collecting a large number of base energy cost measurements and circuit state overheads, we also looked at the role that data values play in our ability to accurately collect this data. In Table 1, we show the effects of using different data operand values for an add instruction. As we can see, using different data values can have a significant impact on the average current and therefore the overall energy consumed (4.0% in this example). One interesting observation is that since many values in a computer are typically close to zero, two-complement can be an inefficient representation when considering energy consumptions (due to the large number of bit flips when a value changes sign).

In addition to investigating the impact of data operand values, we also looked the the impact output operands and the cost of toggling destination register values. In this example, the input data values were kept constant, but the destination register was varied. In Table 2 we show the results of a simple negate instruction. As we can see, large changes in current occur for when the destination register value is toggled. We can see clearly how dependent current measurements are on the number bit flips performed in a cycle.

As an example of the fidelity of our approach, we provide a small example program. We have both utilized our profile data to produce an estimated energy budget for this snipit, as well as have measured the current drawn. The code of the program is shown in Table 3. We have run the program on the ADSP-BF533 and measured the average current during program execution. The energy estimation using our approach was computed to be 3.2 nJ, while the average energy on the BF533 was measured to be 3.3 nJ. This is only a 3% difference.

To date, our results have clearly demonstrated that we can utilize this approach and obtain accurate measurements. In the presentation of this work, we will discuss some further power issues related to leakage energy and temperature dependence. We will also discuss some of the difficulties of estimating energy in the memory hierarchy.

r1 *= r2;
r2 = [i1++];
r0 = r0 + r1 (ns);
r1 = [p2++];
nop;

Table 3: Simple program to validate our approach

References

- [1] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations," *Proceedings of 27th International Symposium on Computer Architecture*, May 2000.
- [2] W. Ye, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, "The Design and Use of SimplePower: A Cycle-Accurate Energy Estimation Tool," *In the Proceedings of the Design Automation Conference*, June 2000.
- [3] V. Tiwari, S. Malik, and A. Wolfe, "Power Analysis of Embedded Software: A First Step Toward Software Power Minimization," *IEEE Transactions on VLSI Systems*, pp. 437-445, Dec. 1994.
- [4] M. T. Chien, V. Tiwari, S. Malik, and M. Fujita, "Power Analysis and Minimization Techniques for Embedded DSP Software," *IEEE Transactions on VLSI Systems*, pp. 123-135, Mar. 1997.