An Automated Instruction Level DSP Energy Profiling Framework

Seth Molloy David Kaeli
Northeastern University, Boston, MA
Analog Devices, Norwood, MA
http://www.ece.neu.edu/groups/nucar
seth.molloy@analog.com,kaeli@ece.neu.edu

January 17, 2007

1 Introduction

Power and energy have come to the forefront in processor design. The increasing demand for portable electronics in new areas has tightened constraints on power and energy consumption for embedded systems. Power densities are creeping up over 100 $watts/cm^3$ [2]. High energy consumption presents significant design challenges.

Instruction Level Energy Estimation (ILEE) is intended for use during the software design phase: after the chip has been produced, when the transistor-level description is not publicly available, and the accuracy requirements are not as strict as in transistor simulation. A fast, accurate estimation is preferable. Accurate high-level estimations can be generated up to 1000 times faster than transistor-level estimations[1].

In this abstract we describe our work on ILEE and how we have moved this framework into the Analog Devices Blackfin VisualDSP++ toolchain. We provide examples on how the framework used instruction level energy analysis to produce complete program power budgets.

2 Instruction Level Energy Estimation (ILEE)

ILEE is founded on the assertion that: An accurate estimation of energy consumption can be obtained by considering the energy effects when a single instruction is executed, as well as the energy effects when two different instructions are executed in sequence. The information acquired when considering only these two factors is sufficient to provide an overall estimation [3]. We have supplemented this base estimation theory with studies specific to the Analog Device Blackfin DSP in order to understand how to tailor the framework to take advantage of unique characteristics of this processor.

There are two sets of data required for ILEE, the *Base Cost* dataset, and the *Overhead Cost* dataset. The *Base Cost* dataset represents the amount of energy required to execute a single instruction. The *Overhead Cost* dataset represents an additional, or reduced, amount of energy incurred by different instructions executing in sequence. The two sets of data are obtained in part by running test programs on the processor and acquiring the average current drawn during their execution. The *Base Cost* is based on acquired current measurements taken while executing an infinite loop containing multiple instances of a single instruction type. The *Overhead Cost* dataset is based on the mentioned *Base Cost* measurements as well as a second set of current measurements taken while executing an infinite loop containing two instructions (with multiple instances in the loop body).

In order to calculate the two datasets, the following equations are used:

$$p_l = I_l V_{dd} \tag{1}$$

$$E_l^m = p_l c_l \frac{1}{f} \tag{2}$$

$$I^{exp}_{1,2} = \frac{I^m_1 c_1 + I^m_2 c_2}{c_1 + c_2} \quad (3)$$

$$I^o_{1,2} = \frac{1}{2}(I^m_{1,2} - I^{exp}_{1,2})(c_1 + c_2) \quad (4)$$

$$E^o_{1,2} = I^o_{1,2} * V_{dd} * \frac{1}{f} \quad (5)$$

The first two equations describe the derivation of the *Base Cost* dataset. The power is proportional to the measured current and the supply voltage(1), while the energy is calculated by accounting for the number of cycles consumed as well as the power and frequency(2). The last three equations describe the derivation of the *Overhead Cost* dataset. Equation 3 calculates the expected current based on the *Base Cost* measurements. Equation 4 calculates the overhead current based on the dual instruction measurements and the expected current calculated in 4. The final equation calculates the energy much like Equation 1, however using the current calculated in 4.

Our current model for estimating the energy consumption or a sequence of instructions, or a basic block, is quite simple. Following our assertions, given a sequence of n instructions, L, the total estimation,E_S, can be found by summing the measured single instructions' base costs, $E^m_l, \forall l \in L$, and all estimated overhead costs, $E^o_{l,l+1}, \forall l \in L, l \neq l_n$:

$$E_S = \sum_{l=1}^{n} E^m_l + \sum_{l=1}^{n-1} E^o_{l,l+1} \quad (6)$$

We have also consider processor stalls, parallel instructions, processor temperature, and data values in our modeling work. These additional factors each have a considerable impact on the energy consumption of the processor and must be taken into account in our estimation framework.

3 Energy Acquisition and Profiling (EAP) Application

We have developed a framework using Analog Devices Inc's Blackfin 537 processor to demonstrate the effectiveness of ILEE. EAP has been designed to provide an architecturally independent framework for ILEE. The application provides not only a framework for acquiring the required data, but also the functionality to utilize the data to provide estimations. Using EAP, we are able to estimate power budgets for short sequences of instructions with less than 5% error.

EAP is comprised of three modules. The Instruction Programming Module(IPM), the Measurement Acquisition Module(MAM), and the Power Simulation Module(PSM). The IPM is the base module containing instruction characteristics that facilitate power simulation and measurement acquisition. This is the most critical module, as it is a base for the other two modules. The MAM is a module that interfaces to a processor and a measurement device to acquire data used for power profiling. The PSM is responsible for utilizing the given data. Interfacing with the IPM, this module loads characteristic data specific to a given processor, and calculates the results, an energy calculator.

4 Conclusion and Future Work

Using this automated system, a complete energy profile of the Blackfin processor can been acquired for different models of the Blackfin processor. We have produced estimations at the basic block level with less than 5% error. A version of this methodology has been integrated into Analog Device's VisualDSP++ development software.

We intend to continue development of the framework to provide for complete program estimation. We are also working to produce estimations on large programs.

References

[1] D. Brooks, V. Tiwari, and M. Martonosi. Wattch. pages 83–94, 2000.

[2] V. Kulkarni. Viewpoint: Power displaces timing as top IC design challenge? *Microwaves & RF*, August 2003.

[3] V. Tiwari, S. Malik, and A. Wolfe. Power analysis of embedded software. *IEEE Transactions on VLSI Systems*, 2(4):437–445, 1994.