Automist - A Tool for Automated Instruction Set Characterization of Embedded Processors

Manuel Wendt[1], Matthias Grumer[1], Christian Steger[1], Reinhold Weiß[1], Ulrich Neffe[2] and Andreas Mühlberger[2]

[1] Institute for Technical Informatics, Graz University of Technology
8010, Graz, Inffeldgasse 16, Austria
wendt@iti.tugraz.at
[2] NXP Semiconductors, Business Line Identification
8101, Gratkorn, Mikronweg 1, Austria
ulrich.neffe@nxp.com

Abstract. The steadily increasing performance of mobile devices also implies a rise in power consumption. To counteract this trend it is mandatory to accomplish software power optimizations based on accurate power consumption models characterized for the processor. This paper presents an environment for automated instruction set characterization based on physical power measurements. Based on a detailed instruction set description a testbench generator creates all needed test programs for a complete characterization. Afterwards those programs are executed by the processor and the energy consumption is measured. For an accurate energy measurement a high performance sampling technique has been established, which can be either clock or energy driven.

Keywords. Software energy estimation, automated processor characterization, testbench generator, current measurement, clock driven sampling, energy driven sampling.

1 Introduction

Nowadays the power consumption of digital systems is one of the key constraints in the design process. It is necessary to meet this constraint in order to fulfill the requirements requested by the user. In the case of mobile devices, these requirements can be, for instance, a higher computational power while having the same budget of energy in order not to decrease the battery life time of the device. But also other benefits of power aware systems such as lesser cooling efforts become more and more important for customers.

To meet those tight design constraints hardware and software has to be optimized at early stages in the design flow. Software power estimation can be done by determination of power consumption in a processor during execution. To allow parallel Hardware and Software development, accurate power models of processors are needed.

Dagstuhl Seminar Proceedings 07041
Power-aware Computing Systems
http://drops.dagstuhl.de/opus/volltexte/2007/1109

Tiwari et al. [1,2] introduced a methodology for software power estimation at instruction level based physical measurements. The model is mainly based on two factors: a) *instruction base cost* (BC), the energy consumed during execution of an instruction and b) *circuit state overhead cost* (CSO), which describes the energy consumed by circuit switching activity between two consecutive instructions. To extract the cost factors the processor is brought into a desired state by a dedicated testbench or/and certain input patterns and the power dissipation is measured.

The accuracy of this method depends on the underlying power model and the precision of the measurement technique. Especially for modern integrated circuits the requirements for the measurement hardware are very high. Because of the increasing clock frequencies in such circuits the measurement time slots, where the considered circuit state occurs are very small and therefore high sampling rates are required. Also the decrease of power consumption in modern mobile devices leads to errors. Especially in the field of smart cards, where asynchronous circuits are used, the power dissipation can be minimized below 10mW [3]. In order to achieve accurate results a dedicated measurement technique with a high resolution is required.

Another possibility to improve the accuracy of the software power estimation is to refine the underlying power models. Also other effects that influence the power consumption such as voltage fluctuations, different operation temperatures or process variations have to be modeled to reduce errors in the power estimation process. Such refinements lead to an increase of needed cost factors and therefore to a higher characterization effort. Since the characterization is a very time consuming procedure, an automated characterization is absolutely essential.

The remainder of this paper is organized as follows. Section 2 surveys related work for instruction set characterization for software power estimation. Section 3 depicts in detail the automated characterization tool. After a functional description of the used power measurement technique in section 4 the energy model for the software power estimation is introduction (section 5). Experimental results are presented in section 6. The conclusions are summarized in section 7.

2 Related Work

Based on Tiwaris methodology for instruction level power characterization several projects can be found in literature. To overcome the huge characterization effort several authors presented techniques to minimize required measurements by grouping the instructions depending on the functional unit used by the instructions [4]. However, up to 3 man months are necessary to characterize a processor, depending on the processor architecture.

The characterization methods are heavily dependent on the processor and require detailed knowledge of the architecture. Brandolese et al. propose a general methodology, independent of the specific processor [5]. The methodology obviates the architectural level and focuses on the functionalities involved in the instruction execution. The proposed model is built on an a priori knowledge of both the energy characterization of a set of instructions and the relevant functional characteristics of each instruction of the set. The model is thus a trade-off between accuracy and generalization.

Besides refined power models also improved power measurement techniques can be found in literature. In [1,6] integrating digital ammeter in the processor supply line have been used for data acquisition. Since the maximum frequency in the power spectrum of the supply current in digital systems can be several hundred MHz [7], this kind of measurement is too error-prone due to the limited bandwidth of digital multimeters.

Oscilloscopes with high sampling rates can overcome the phenomena of high frequencies in the power spectrum. The current is transformed by a shunt resistor to a proportional voltage drop, which can be measured by the oscilloscope. Shunt based measurement techniques have been used in [8,9]. Supply voltage fluctuations over the measurement resistor and noise introduced by the stray inductance limit the accuracy of such methods.

Reduced supply voltage fluctuations are archived in [10]. A simple current mirror based on 4 transistors provides a copy of the current drawn by the processor. The voltage drop over this mirror is constantly 2 times the basis emitter voltage and therefore independent of the input current. Since the copied current is measured by a digital oscilloscope spikes in the instantaneous current are one of the key problems of this method.

Hardware based current integration is one possible solution of this problem. In [7] the processor is powered by two capacitors. Every second clock cycle one capacitor acts as power supply and is discharged by the current drawn by the processor. Meanwhile the other one is charged by a supply voltage. The discharge current causes a voltage decrease which is proportional to the consumed power during one clock cycle. Also in this case supply voltage fluctuations due to the discharge process affect the accuracy of the measurement result.

While there are many projects dealing with instruction set characterization, we have not found any references in the literature dealing with a fully automated characterization process for different processor architectures. The presented process is based on a new current sampling technique, which can be either clock or energy driven.

3 Characterization System

The characterization of processors is a very time consuming procedure and implies detailed information about the processors architecture. For energy estimations of different SOC architectures during the design process of embedded systems an automated characterization is absolutely essential. For that matter we have implemented an automated characterization system (figure 1).

Information about the processors architecture has to be provided in form of a detailed instruction set description. The testbench generator creates all required assembly programs to characterize all BC and CSO factors for the instructions

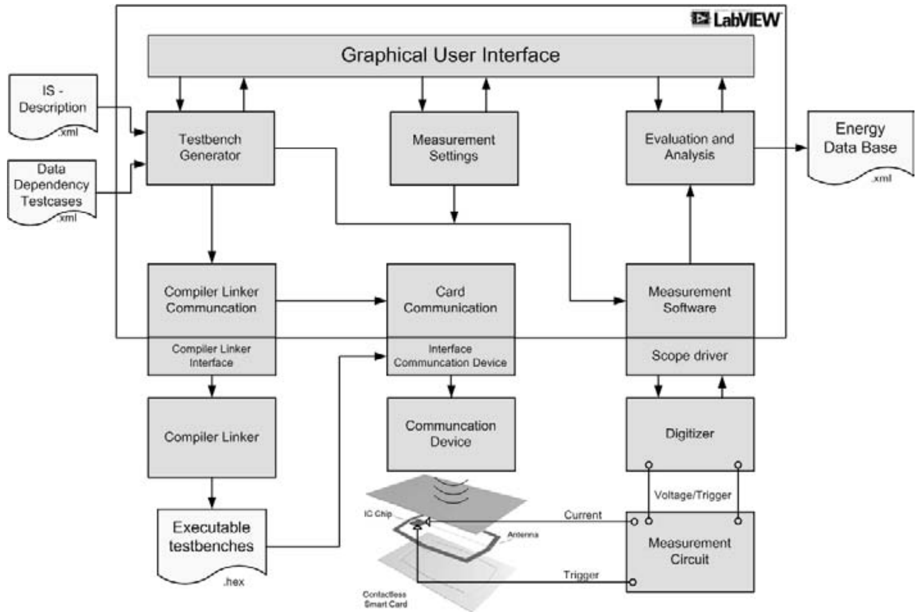

Fig. 1. Characterization system overview.

defined by the description file. The generation of additional testbenches to characterize the influence on power dissipation of data dependencies like different operand addresses or values can be forced by well defined data dependency xml file.

Afterwards these testbenches are compiled by an architecture dependent compiler and the executable programs are loaded one by one on the processor. During execution of these programs the processor sets a trigger signal to indicate that the initialization has been completed and the execution of the characterization code starts. This signal triggers the digitizer which samples the output of a dedicated power measurement circuit.

After the data acquisition step software filters eliminate spikes in the current profile. The measurement evaluation and analysis step extracts the cost factors for each characterization testbench and stores them into an energy data base. All steps in the characterization process can be controlled by the user via a graphical user interface implemented in LabView ®.

4 Measurement Hardware

The measurement hardware supports two types of energy samplings: a) *clock driven* and b) *energy driven* sampling. Clock driven sampling records the power profile cycle accurately. This method is needed for the characterization of complex operations like branch instructions with delay slots. For all other instructions the more accurate energy driven sampling is used. In this case it is not possible to assign exactly one energy value to each clock cycle because the sampling rate only depends on the current drawn by the processor.

4.1 Clock Driven Sampling

As it is mentioned above, the clock driven sampling records the consumed average power given by $P = I_{DD} \cdot V_{DD}$ during each processor cycle. Since the supply voltage V_{DD} of processors is constant, the power only depends on the average current I_{DD}. An average measurement of the current can be done by the circuit shown in figure 2. While one capacitor of the circuit is charged with an exact copy of I_{DD}, the other one is discharged over the resistor R. At the end of each clock cycle the voltage V_C on the charged capacitor is sampled and the capacitors are switched. Assumed the delay time of the switches is much less than the clock period the integration interval is given by $1/f_{CLK}$ and the average current can be calculated as follows: $I_{DD} = V_C \cdot C \cdot f_{CLK}$. R_{bias} in figure 2 is used for biasing purposes. An offset DC current value due to the R_{bias} has to be subtracted from the measured value.

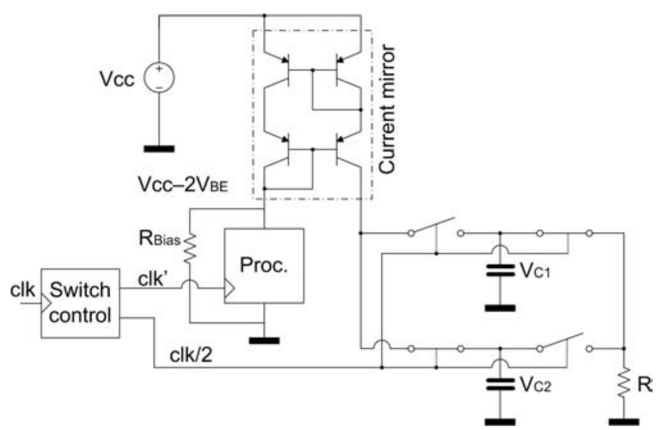

Fig. 2. Schematic of the clock driven sampling circuit.

The switch control circuit delays the processor clock for at least one switch delay. This causes a start of current integration just a tick before the processor clock rises. The reason for this short forward shift of the integration interval is that most of the energy in synchronous circuits is consumed during signal transitions at the beginning of each clock cycle. At the end of a clock period, the processor is in a stable state and consumes only leakage power. Since leakage currents of CMOS circuits are very small, errors caused by the switching capacitors are minimized.

4.2 Energy Driven Sampling

The second measurement technique is based on the same principle as the clock driven sampling described in the previous section. It differs only in the switch control. As it is shown in figure 3 a comparator compares the voltage V_C across the charged capacitor to a precise reference voltage V_{Ref}. If V_C exceeds V_{Ref}, a flip flop changes his state and a trigger pulse is generated, which leads to a change of the switch states. Afterwards the same procedure starts with the discharged capacitor.

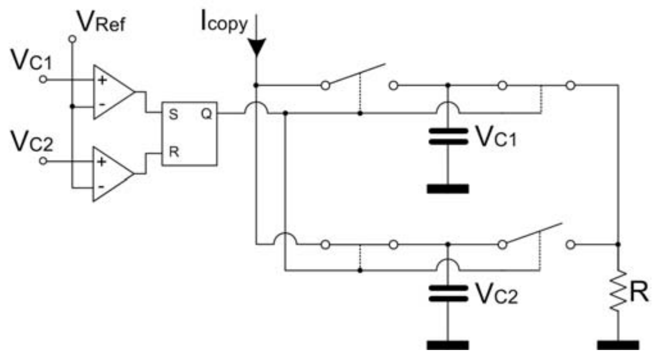

Fig. 3. Schematic of the energy driven sampling circuit.

The trigger pulse indicates that the charged capacitor has accumulated a charge of $Q_C = C \cdot V_{Ref}$, during which time the processor has consumed an energy of $E = Q_C \cdot V_{DD}$. The power profile of the processor can be derived from the time between every pair of pulses t_{pulse}. Because of delays in the switch control circuit, a constant delay t_{delay} has to subtracted from t_{pulse} to get the right time during which the energy E has been consumed. The average power P between two pulses can be calculated as
$P = E/(t_{pulse} - t_{delay})$.

5 Energy Model

The energy model developed is a flexible and accurate combination of an instruction-level energy model and a data dependent model. The total energy E_{total} consumed by a program is the sum over all clock cycles n_{total} of the energy consumption per cycle E_{cycle}:

$$E_{total} = \sum_{n=0}^{n_{total}} E_{cycle}(n). \qquad (1)$$

The energy per clock cycle can be further decomposed into four parts: instruction dependent energy dissipation E_i, data dependent energy dissipation

E_d, energy dissipation of the cache system E_c, and finally the dissipation of all external components E_e including the bus system, memories and peripherals:

$$E_{total} = \sum_{n=0}^{n_{total}} [E_i(n) + E_d(n) + E_c(n) + E_e(n)]. \tag{2}$$

The instruction dependent part of this model is based on the instruction-level energy model defined by Tiware et al. [1,2]. It defines base costs for each instruction and considers switching activity between different consecutive instructions by use of a Circuit State Overhead (CSO) term. To characterize a particular instruction a loop is used which is made of several of these instructions, small enough to fit in the instruction cache and large enough to minimize influence of the branch instruction at the end of the loop. For circuit state overhead, a loop with two alternating instructions is used to measure the circuit state overhead between these two instructions.

A pipeline aware model is used for the cycle by cycle estimation. The main idea of this approach is to distribute the measured costs among all pipeline stages. Thus all other inter-instruction costs (IIC) caused by inter instruction locks, transaction look-aside buffer or data/instruction cache misses can be modeled by pipeline stalls.

$$E_d = E' - E_{BC} \tag{3}$$

$$E_d(i) = \beta_0 + \beta_1 \cdot x(i) \tag{4}$$

Each instruction is responsible for the calculation of its data depended energy consumption E_d. E_d is, as it can be seen in equation 3 defined as the difference between the energy dissipation caused by an instruction with a varying operand value and destination address E' and the instruction BC E_{BC} characterized with reference data (typically 0).

Since a complete characterization of the whole range of values of the operands is only theoretically possible, the data dependent energy consumption is modeled by means of linear regression. In equation 4 $x(i)$ describes a property of the operand data word i going to be modeled. This property can be either the hamming distance between the data word and the reference value or the hamming weight of the data word. β_0 and β_1 are constants and $E_d(i)$ is the energy dissipation of the instruction with the data word i.

6 Experimental Results

The following experimental diagrams present the performance characteristic of the two measurement techniques mentioned above. These diagrams present typical cases from the multiple measurement test, which have been performed repeatedly in the lab.

To determine the measurement accuracy all measurement results have been compared to a constant current at the input of the current mirror. This input current was drawn by shunt resistors with a tolerance of less then 0.05%. The voltage drop across these resistors has been measured to calculate the size of the input current. The comparison of this current with the sampled values has been used to determine the measurement error.

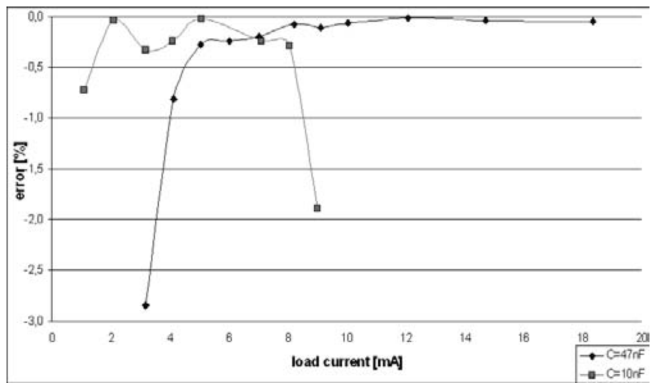

Fig. 4. Experimental results for the clock driven sampling at 2MHz with 2 different capacities.

Experimental results for the clock driven sampling have shown a high dependency between operation range, measurement accuracy and hardware configuration (Figure 4). If the input current is too big compared to the chosen capacitors, the capacitors become fully loaded before the clock cycle ends. On the other hand too big capacitors cause only a small voltage drop which leads to errors in analog to digital conversion.

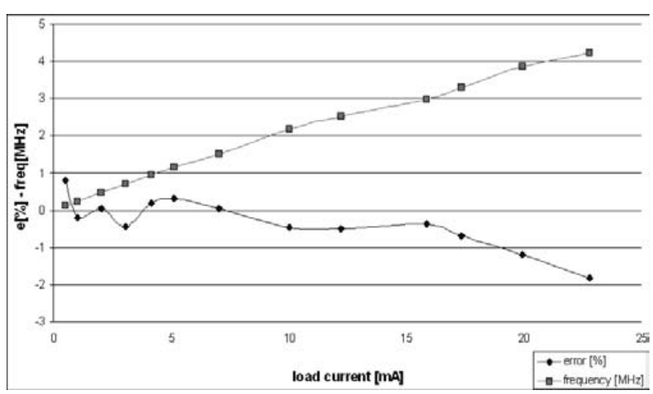

Fig. 5. Energy driven sampling: error and trigger frequency depending on the input current for a reference voltage of 1V and a capacitance of 4.7nF.

Figure 5 shows the input current dependent error behavior and frequency behavior of the energy driven sampling. In this case the maximum voltage across the capacitor is given by the constant reference voltage V_{Ref} and independent from the input current. This leads to an enlarged operation range for a fixed hardware configuration as it is shown in Figure 5, where the error is less than 0.9% over an oparation range of 0.5-20mA. The frequency on the other hand depends linearly from the input current.

7 Conclusion

In this paper we have presented an environment for automated instruction set characterization of embedded processors based on physical measurement. After a survey of the characterization system, two different power measurement techniques have been described in detail. Followed by the underlying energy model the performance of those power sampling techniques has been discussed. Future work will be done in further examinations on the degree of portability of this approach to different processor architectures.

References

1. Tiwari, V., Malik, S., Wolfe, A.: Power analysis of embedded software: a first step towards software power minimization. In: ICCAD '94: Proceedings of the 1994 IEEE/ACM international conference on Computer-aided design, Los Alamitos, CA, USA, IEEE Computer Society Press (1994) 384–390
2. Lee, M.T.C., Tiwari, V., Malik, S., Fujita, M.: Power analysis and low-power scheduling techniques for embedded dsp software. In: ISSS '95: Proceedings of the 8th international symposium on System synthesis, New York, NY, USA, ACM Press (1995) 110–115
3. van Gageldonk, H., van Berkel, K., Peeters, A.: An asynchronous low-power 80c51 microcontroller. In: Fourth International Symposium on Advanced Research in Asynchronous Circuits and Systems, 1998. (1998) 96–107
4. Nikolaidis, S., Laopoulos, T.: Instruction-level power consumption estimation of embedded processors for low-power applications. Comput. Stand. Interfaces **24** (2002) 133–137
5. Brandolese, C., Fornaciari, W., Salice, F., Sciuto, D.: An instruction-level functionally-based energy estimation model for 32-bits microprocessors. In: DAC '00: Proceedings of the 37th conference on Design automation, New York, NY, USA, ACM Press (2000) 346–351
6. Flinn, J., Satyanarayanan, M.: Powerscope: A tool for profiling the energy usage of mobile applications. In: WMCSA '99: Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, Washington, DC, USA, IEEE Computer Society (1999) 2
7. Chang, N., Kim, K., Lee, H.G.: Cycle-accurate energy consumption measurement and analysis: case study of arm7tdmi. In: ISLPED '00: Proceedings of the 2000 international symposium on Low power electronics and design, New York, NY, USA, ACM Press (2000) 185–190

8. Fay Chang, Keith I. Farkas, P.R.: Energy-driven statistical sampling: Detecting software hotspots. In: Power-Aware Computer Systems : Second International Workshop, PACS 2002 Cambridge, Cambridge, MA, USA, Springer Berlin / Heidelberg (2003) 110–129
9. Russell, J.T., Jacome, M.F.: Software power estimation and optimization for high performance, 32-bit embedded processors. In: ICCD '98: Proceedings of the International Conference on Computer Design: VLSI in Computers and Processors, Austin, TX, USA, IEEE Computer Society (1998) 328–333
10. Nikolaidis, S., Kavvadias, N., Neofotistos, P., Kosmatopoulos, K., Laopoulos, T., Bisdounis, L.: Instrumentation set-up for instruction level power modeling. In: PATMOS '02: Proceedings of the 12th International Workshop on Integrated Circuit Design. Power and Timing Modeling, Optimization and Simulation, London, UK, Springer-Verlag (2002) 71–80