

Instruction Level Power Consumption Estimation – Issues and Review

V.A.Kulkarni

Research Scholar, E&C Department
GIT, Belgavi, India
vak@amgoi.edu.in

Dr. G.R.Udupi

Professor, E&C Department
GIT, Belgavi, India
grudupi@git.edu

Abstract— Power consumption estimation in battery operated embedded system is crucial. Power consumption minimization not only increases battery life but also reduces heat generation in chip and other relevant issues. Once the software power consumption is known, different techniques can be used to minimize it. In this paper various issues related to Instruction Level Power Consumption Estimation for pipelined processors is discussed. Literature review is also presented.

Keywords—current measurement, embedded system, energy estimation, software power

I. INTRODUCTION

Embedded systems are being used extensively in day to day life. Their application ranges from domestic, medical, agriculture to defense and space applications. The functionalities and complexities of embedded systems are increasing drastically with customer expectation of numerous functions on single system. The rate at which transistors are added in ICs is much more than the rate at which battery technology is improving. The current rate of battery life improvement of around 5% per year [1] means that the limited energy budget could delay the introduction of the future chips needed to support workloads whose complexity increases by one order of magnitude every 5 years [2]. With the rapid progress of semiconductor technology, the transistor count on a single chip has reached 10,000,000,000 in SPARC M7 processor [3]. Table.1 shows the increase in transistor count in ARM processors.

TABLE I. TRANSISTOR COUNT

Processor	Transistor count	Year	Area
ARM 1	25,000	1985	50 mm ²
ARM 2	30,000	1986	30 mm ²
ARM 9TDMI	111,000	1999	4.8 mm ²
ARM Cortex-A9	26,000,000	2007	31 mm ²

The reduced power consumption, apart from increasing battery life, also helps in less heat generation. A formula based on the Arrhenius Law suggests that life expectancy of component decreases 50% for every 10⁰C increase in the temperature. Thus, reducing a component's operating temperature by the same amount (consuming less energy), doubles its life expectancy [4]. Lesser power consumption reduces heat dissipation, resulting in low cost packaging, cooling methods and increases device reliability.

II. POWER CONSUMPTION MODELS AND METHODS

Power consumption model of the processor software can be categorized as:

Low-Level models.

High-Level models.

Low level models are also called as hardware models. Various levels of abstractions under this category are:

- Circuit/Transistor level[6,7]
- Logic gate level [5].
- RT-level [8,9]
- Architectural level [10]

High-Level models deal only with instructions and functional units from the software point of view and without electrical knowledge of the underlying architecture. The existing high level power estimation models can be classified as:

- Instruction Level Power Analysis (ILPA)
- Functional Level Power Analysis (FLPA) [11,12]

In ILPA, a power consumption model is associated with instructions or instruction pairs. The power consumed by a program running on the processor can be estimated using the model. Large number of experiments required to obtain the model is major drawback.

In FLPA the processor is separated into functional blocks. The power consumption of each block is characterized through mathematical functions obtained from several measurements and/or simulations. The various issues related to ILPA are:

The method of getting energy consumption data - Energy consumed can be calculated knowing the value of current drawn by processor core, core voltage and time taken to execute the instruction. Methods in use are: measuring voltage across a precision resistor connected in core power line (average current), current mirror method or cycle

accurate current monitoring using digital oscilloscope. More complex circuitry results in more accurate readings.

Base cost – It is the energy consumed while executing a particular instruction. This can be measured by placing the instruction in a loop so that a stable reading can be obtained. Small loop may result in error (due to branch effect) and a large loop may result in cache miss. Thus choosing loop size is very crucial.

Inter instruction cost – It is the energy consumed when executing two consecutive instructions. Say Instruction-1 base cost is 3 nJ and Instruction-2 base cost is 2 nJ. When Instruction 1 & 2 are executed, the energy consumed can be 5.5 nJ. This 0.5 nJ is due to inter instruction effect. It can be negative also. Lot of measurements to be carried out to check all combinations of Instruction set Architecture (ISA) and is given by $n(n-1)/2$. Where n is number of instructions in ISA. The different ordering of instructions result in different values i.e. it is not symmetric.

Energy sensitive factors - Apart from base cost and inter instruction cost, software power depends on certain factors like register number, register value, immediate value, operand value, operand address and fetch address.

Resource constraints – Pipe line stalls and pipe line flushes results in delay.

Energy model - Instruction level energy model can use Statistical method or Non statistical method. Earlier research mainly focused on non-statistical method [15]. Necessity of large number of measurements to be taken is major constraint. To simplify the complexity involved, several instructions can be clubbed together as instruction groups. These grouped instructions could simplify the energy cost calculations. Grouping can be based on type of instructions, addressing mode or cycles used for execution. Another method uses average energy consumed by a program as equal to average energy consumed per instruction multiplied by number of instructions in the program. This method is applicable for CISC architecture only. Recently statistical method is used to model software energy. Linear regression is used to obtain coefficients for energy consumption model. In this method, the processor is treated as a black box without the design details of the processor. The statistical method finds the correlation between instructions and energy dissipation. The statistical method gives regression coefficients for each instruction.

III. LITERATURE REVIEW

In [15] only average power is estimated. Measurements are taken with a standard off-the-shelf, dual-slope integrating digital multimeter. Voltage across a known small resistance gives the value of average current. The value of shunt resistance chosen is very small so that its effect on circuit is minimum. The voltage drop across the resistance degrades the performance of device. Loops containing same instruction is executed and average

current drawn is used to model the energy consumption of the instruction. No pipe stalls or cache misses. Base cost and inter instruction cost measured. A short startup code for the hardware is executed only once so that it does not affect the measurement. Power reductions of up to 40% obtained by rewriting the code using the information provided by the instruction level power model. Distinction between base instruction cost and base processor cost (any other cost always present in processor and is unaffected by instruction being executed) is not made. Impact of data width on processor energy consumption not taken into account.

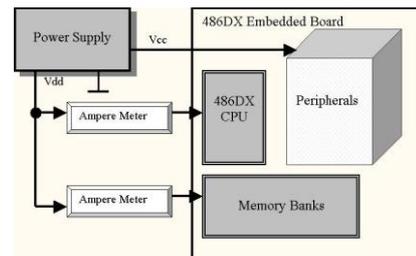


Fig.1. Experimental set up used in [15]

In [16] a complex circuit topology for cycle accurate energy measurement is proposed. It is based on charge transfer using switched capacitors. The idea is to power up the processor from charged capacitor for a short period say execution time of a test instruction and measure the voltage V_1 at the beginning and the voltage V_2 at the end of measurement period. The change in the voltage level across the capacitors is proportional to the square of the consumed energy and this value is used for the calculation of energy in a clock cycle. $E = (1/2) (C) (V_1 - V_2)^2$. The approach was validated over the DMM method and the measurement errors were found not to exceed 2–3 %. This approach cannot provide shape of current waveform. To measure the energy, various reference and test (*ref*, *test*) instruction pairs are formed. The measurement setup requires high sampling frequency data acquisition means. An energy consumption model for the ARM7 processor is obtained.

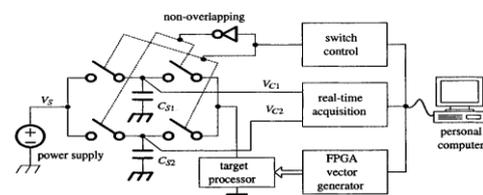


Fig.2. Experimental set up used in [16]

In [17] various energy sensitive factors like, register number, register value, immediate value, operand value, operand address and fetch address are modeled by a coefficient. For measurement of pure base cost NOP instruction is used. Instantaneous current is measured using a current mirror circuit with BJTs and a high frequency digital

storage oscilloscope [20]. Thus the drawback of voltage fluctuations found in series resistor inserted in processor power line is eliminated and the resolution of the measurements is considerably increased. It is shown that instructions with same number of cycle will consume almost same base energy (difference less than 20%). Condition in an instruction will not significantly influence base energy. To simplify the number of measurements to be made for inter instruction cost, four groups are formed in data processing instructions. Inter instruction cost found to be 5% to 15% of base cost. No symmetry and negative value of inter instruction cost is observed. Single absolute overhead value is used to describe pipeline stalls and flushes. Accuracy of method is checked by a number of programs with $CPI = 1$ and $CPI > 1$. The error found to be up to 1.5%

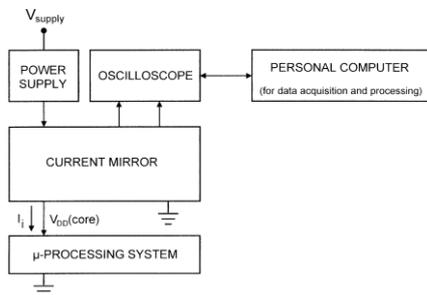


Fig.3. Current measurement scheme used in [17]

In [18] an energy consumption modeling technique for embedded systems is presented, in which the number of cycles is considered instead of the number of executed instructions. Besides, it computes the energy by a polynomial expression. This work adopts analytical models that may require so many simplifications and assumptions that may affect the results accuracy. Here the energy consumption of a system consisting a Motorola HC908GP32 microcontroller, an external A/D converter and an external DRAM is modeled. The energy consumption of each module is measured separately. The energy consumption of the core is estimated based on the number of execution cycles of each instruction rather than the type of the instruction.

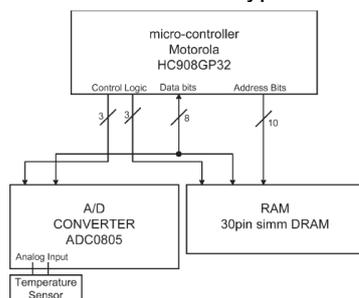


Fig.4. Target platform used in [18]

Hence the number of parameters is significantly reduced which results in a simpler model. But the model is only tested for a single benchmark that is the logging application, and various benchmarks have not been considered.

The energy model proposed in [19], models the energy consumption of the CPU, Flash memory, SRAM and the memory controller. It provides more accurate estimates than the classic model introduced earlier. The memory model is still quite simple as no cache hierarchy is modeled and does not include the energy consumption of read-only memory (ROM) or peripherals either. The memory and memory controller related energy is added to the relevant stages. As the inter instruction energy cost is about 5% of the base instruction cost, detailed estimation of inter instruction cost is not carried out. This simplifies the model. Other parameters like the Hamming distance and weight of the instructions, number of bit flips in the register bank, and the number of shift operations are considered. Instantaneous current is also measured. This method use a digitized oscilloscope to read the voltage difference over a precision resistor. The resistor is placed between the power supply and the core supply pin of the processor. Digital oscilloscope sometimes shows inaccurate results due to current spikes. The benefit of this model is that it considers the factors which make the pipeline stall and addresses the problem of overhead energy. But there are 38 parameters in this model which makes it difficult to use. Energy model for the ARM7TDMI uses 60 specialized tests to estimate the coefficients of each energy sensitive factor. On top of this, there are 35 parameters for the model including: the ARM7 instruction set, register bank bit, instruction word, hamming distance etc. The energy cost of pipeline stalls is approximated by an average value from a benchmark suite.

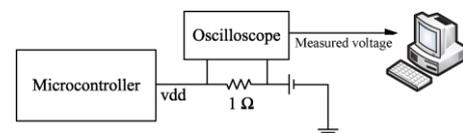


Fig.5. Architecture of measurement system used in [19]

In [21] an instruction-level power model based on an ARM1176JZF-S processor to predict the software power is developed. It is shown that the power is related to both the distribution of instruction types and the operations per clock cycle (OPC) of the program. It is proved that energy per operation (EPO) decreases with increasing operations per clock cycle. Instead of individual instructions, average power of a program is considered. Inter instruction effect is not considered which saves lot of calculation and measurements. For inter instruction cost measurements will be proportional to the square of the number of the instructions in the instruction set architecture (ISA). There are 51 instructions in ARM assembly language, it would need at least 1300 measurements. Pipeline stall effects are also considered by OPC instead of cache miss. Power measurements carried out with a 0.51 series resistor between the power supply and the CPU. A digitizing oscilloscope with a sample rate of 2GHz used to measure the instantaneous power. To find instruction cost, loop consisting of same opcode of size of 8KB

(2000 instructions) considered to avoid cache miss (L1 data and instruction caches are 16 KB). It is assumed that all arithmetic and logic instructions consume the same basic power in all addressing modes. The instructions are divided into three clusters based on their power and behaviors: arithmetic/logic, load, and store. The model shows good performance with a maximum estimation error of -8.28% and an average absolute estimation error is 4.88% over six benchmarks.

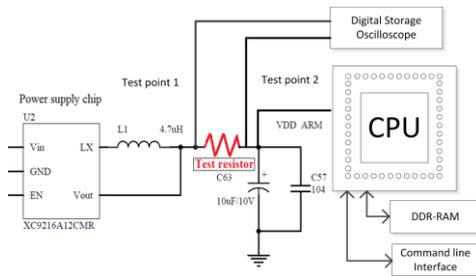


Fig.6. Power supply and test platform used in [21]

In [22] instruction level software power analysis for TMS320C6713 DSP processor is presented. Three power test points are provided on DSK (DSP Starter Kit). Core current, I/O current and system current can be measured using DSK. The current drawn by DSP Core is measured by Agilent U1241B 4 digit digital multimeter. Loop size for instruction cost is obtained with the help of Code Composer Studio (CCS) profiler. Several sample programs used to validate the results. Non consideration of cache penalty and precision of DMM used are the main reasons for error in calculated and actual value.

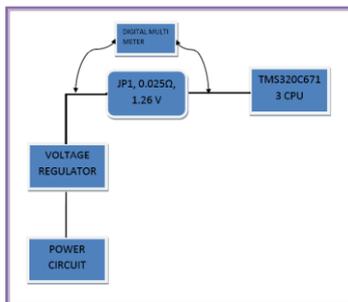


Fig.7. TMS320C6713 DSK used in [22]

In [23] a hard ware flat form is used to implement Dynamic Voltage and Frequency Scaling (DVFS). To measure power, a resistor is placed between each microcontroller power pin and the power supply line, and the voltage drop across the resistor is measured. The measured value gives the current drawn by the power pin. Since the current is less than 100 mA and such a low value cannot be used by the ADC of microcontrollers, the voltage value is amplified using an operational amplifier. 10 bit ADC converts this to digital signal and the data are sent to the computer.

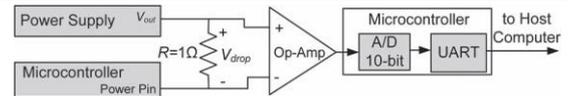


Fig.8. Power measurement set up used in [23]

In [24] base cost and inter instruction cost of Cortex M4 processor is given. Various cost associated with software power and energy consumption in pipe lined processor is discussed. To find processor base cost, a loop with NOP is executed. To find effect of operand value, readings taken for immediate values of ffff, 0000 and 1111. Average of these three values are taken as instruction base cost. It is shown that data movement instructions draw more current followed by comparison, arithmetic and logical instructions. Inter instruction cost is given.

In [25] microprocessor is considered a black box. Current is measured with a shunt resistor and differential amplifier that will introduce error since there will be a voltage decrease of the microcontroller supply. The time window is measured with the help of a free GPIO pin that is toggled before and after the execution of the instruction. The microprocessor of interest is ARM Cortex-M4 implemented on a Texas Instruments. Tiva TM4C123G deeply-embedded microcontroller. The energy model file text file that contains a look-up table for each instruction from the target ISA. Different domains like memory name (FLASH, SRAM, FRAM, EEPROM, etc.), start and end address, temperature, voltage, frequency and operands are declared.

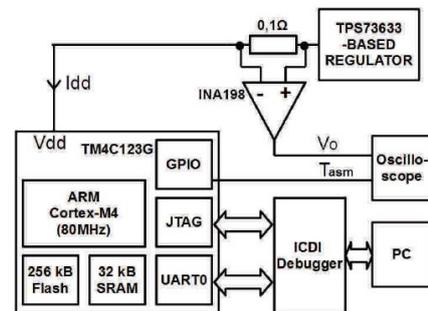


Fig.9. Experimental set up used in [25]

In [26] energy consumption estimation for embedded applications based on ultra-low power heterogeneous multicore DSP platform is presented. The DSP platform used contains five heterogeneous cores. The embedded platform is developed to enhance performance in a hearing aid. Battery life is estimated for different types of input signal. Pipeline stalls and cache misses are not considered. Overhead due to the core's shared resources is not included in this model. Total power is sum of static power (processor cost), power of active peripherals and power of active cores. Measurements are carried out using Fluke True-RMS Industrial Logging Multimeter. To find inter instruction cost, for ISA with 100 instructions requiring 4950 measurements, the task is simplified by measuring inter instruction cost of NOP and test instruction and repeating loop for 1000 times. This approximation has reduced the number of measurements from 4950 to 100. Thus great saving in

time and resources is achieved. Results obtained are validated against two micro benchmark programs and accuracy of 99.97% is achieved.

IV. CONCLUSION

The paper explains in detail various issues related to Instruction Level Power Analysis. With more complex instrumentation, one can get cycle accurate results. By making suitable assumptions, which will save considerable time and resources, results with good accuracy can be obtained. The paper also reviews research work carried out. Estimation of software power with reasonable accuracy, using generally available instrumentation at short time will accelerate research in estimation of software power thereby in power management in embedded system.

REFERENCES

- [1] Nam Sung Kim et al., Micro architectural power modeling techniques for deep sub-micron microprocessors, in: Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '04), ACM, NewYork, NY, USA, 2004, pp. 212–221.
- [2] C.H. van Berkel, Multi-core for mobile phones, in: Proceedings of the Conference on Design, Automation and Test in Europe (DATE '09), Belgium, 2009, pp. 1260–1265.
- [3] https://en.wikipedia.org/wiki/Transistor_count (visited on 16-02-2017).
- [4] R. Ge, et al., "Performance-constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters," presented at the Proceedings of the 2005 ACM/IEEE conference on Supercomputing, 2005.
- [5] T. Chou and K. Roy, "Accurate Estimation of Power Dissipation in CMOS Sequential Circuits," IEEE Transaction VLSI Systems, vol. 4, pp. 369–380, September 1996.
- [6] F. N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," IEEE Transactions on VLSI Systems, vol. 2, pp. 446–455, 1994.
- [7] S. Gupta and F. N. Najm, "Power macromodeling for high level power estimation," in proceedings of the 34th annual conference on Design automation DAC'97. New York, NY, USA: ACM, 1997, pp. 365–370.
- [8] D. Marculescu, R. Marculescu, and M. Pedram, "Information theoretic measures of energy consumption at register transfer level," in proceedings of the International Symposium on Low Power Design ISLPED'95. New York, NY, USA: ACM, 1995, pp. 81–86.
- [9] Q. Wu, Q. Qiu, M. Pedram, and C.-S. Ding, "Cycle-Accurate Macro-Models for RTLevel Power Analysis," IEEE Transaction VLSI Systems, vol. 6, no. 4, pp. 520–528, 1998.
- [10] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A Framework for Architectural- Level Power Analysis and Optimizations," SIGARCH Computer Architecture News, vol. 28, no. 2, pp. 83–94, 2000.
- [11] J. Laurent, E. Senn, N. Julien, and E. Martin, "High Level Energy Estimation for DSP Systems," in proceedings International Workshop on Power And Timing Modeling and Optimization and Simulation PATMOS'01, September 2001, pp. 311–316.
- [12] M. Schneider, H. Blume, and T. G. Noll, "Power Estimation on Functional Level for Programmable Processors," in journal of Advances in Radio Science, vol. 2, May 2005, pp. 215–219.
- [13] J. Castillo, H. Posadas, E. Villar, and M. Martinez, "Energy consumption estimation technique in embedded processors with stable power consumption based on source-code operator energy figures," in XXII Conference on Design of Circuits and Integrated Systems, 2007.
- [14] F. Rosa, L. Ost, T. Raupp, F. Moraes, and R. Reis, "Fast energy evaluation of embedded applications for many-core systems," in Power and Timing Modeling, Optimization and Simulation (PATMOS), 2014 24th International Workshop on, IEEE, 2014, pp. 1-6.
- [15] V. Tiwari, S. Malik, and A. Wolfe, "Power analysis of embedded software: A first step towards software power minimization", IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 2, NO. 4, DECEMBER 1994, pp. 437-445.
- [16] Naehyuck Chang, Kwanho Kim, and Hyung Gyu Lee, "Cycle-Accurate Energy Measurement and Characterization With a Case Study of the ARM7TDMI", IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 10, NO. 2, APRIL 2002, pp. 146-154.
- [17] Nikolaos Kavvadias, Periklis Neofotistos, Spiridon Nikolaidis, C. A. Kosmatopoulos, and Theodore Laopoulos, "Measurements Analysis of the Software-Related Power Consumption in Microprocessors", IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 53, NO. 4, AUGUST 2004, pp. 1106-1112.
- [18] V. Konstantakos, A. Chatzigeorgiou, S. Nikolaidis, T. Laopoulos, "Energy consumption estimation in embedded systems", IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 57, NO. 4, APRIL 2008 pp. 797–804.
- [19] Mostafa Bazzaz, Mohammad Salehi and Alireza Ejlali, "An Accurate Instruction-Level Energy Estimation Model and Tool for Embedded Systems", IEEE TRANSACTIONS ON

- INSTRUMENTATION AND MEASUREMENT, VOL. 62, NO. 7, JULY 2013, pp. 1927-1934.
- [20] Theodore Laopoulos, Periklis Neofotistos, C. A. Kosmatopoulos, and Spiridon Nikolaidis, "Measurement of Current Variations for the Estimation of Software-Related Power Consumption", IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 52, NO. 4, AUGUST 2003, pp. 1206-1212.
- [21] Wang, Wei and Zwolinski, Mark (2014) "An improved instruction-level power model for ARM11 microprocessor". In, *High Performance Energy Efficient Embedded Systems (HIP3ES)*, Berlin, DE, 23 Jan 2013. 7pp.
- [22] P. V. Joshi, K. S. Gurumurthy, "Software power analysis for embedded DSP software", in *Proc. of the Intl. Conf. on Advances in Computing and Information Technology, (ACIT 2014)*, Bangkok, Thailand, 2014.
- [23] Mohammad Salehi and Alireza Ejlali, "A Hardware Platform for Evaluating Low-Energy Multiprocessor Embedded Systems Based on COTS Devices", IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 62, NO. 2, FEBRUARY 2015, pp. 1262-1269.
- [24] V. A. Kulkarni and G. R. Udupi, "Software Power Measurement of ARM Processor Based Embedded System", *EJERS, European Journal of Engineering Research and Science* Vol. 1, No. 5, November 2016, pp.5-9.
- [25] LUBOMIR BOGDANOV, "LOOK-UP TABLE-BASED MICROPROCESSOR ENERGY MODEL", International Scientific Conference on Engineering, Technologies and Systems TECHSYS 2016, Technical University – Sofia, Plovdiv branch , 26 – 28 May 2016, Plovdiv, Bulgaria.
- [26] Momcilo V. Krunic, Miroslav V. Popovic, Vlado M. Krunic, Nenad B. Cetic, "Energy Consumption Estimation for Embedded Applications", *ELEKTRONIKA IR ELEKTROTEHNIKA*, ISSN 1392-1215, VOL. 22, NO. 3, 2016, pp. 44-49.