Energy Efficiency of Embedded Controllers

Mustafa Engin
Ege University Ege Higher Vocational School Department of Electronics Technology
Bornova, 35100 İzmir Turkey
mustafa.engin@ege.edu.tr

Abstract— **One of the most important limitations to be considered in the design phase of the embedded system is the energy consumption. The energy consumption is more efficient, especially in the embedded devices fed by the battery, because the charging time is an obstacle to the use of the device. In addition, energy efficiency in terms of power dissipation and system health is important in other embedded applications such as biomedical, test and measurement, industrial control and robots. Reducing energy consumption during the design phase of the embedded system is generally considered as the task of the hardware. In fact, the software should also undertake the task of improving energy efficiency. In this research, the control algorithm of the embedded system is written in two different methods and their energy consumption was compared.**

Keywords-component; embedded controller; energy efficiency; PI control; PID control, Instructional-level energy estimation

I. INTRODUCTION

Embedded system designers have to regard energy consumption more than in the past, and the restrictions on this issue are increasing day by day. Consequently, processor and controller chip manufacturers offer a wide variety of microcontroller devices and many integrated specific peripherals that increase energy efficiency.

In general, different abstraction level can be used to perform energy estimation at different abstraction levels. Source-code level is the highest abstraction level in which we could analyze energy consumption. Functional-level approach estimates energy consumption by analyzing the energy consumption of functional units on processor. The energy parameters for energy model of functional-level are the access rate of the on-chip memory, the clock frequency, or the degree parallelism etc. Moreover, Blume et al. [1] present a hybrid approach that combines functional-level and instruction-level power model to estimate energy consumption. Early studies mainly focus on measuring energy cost per instruction without considering design details of processors; recent studies take into account pipeline details so that energy cost for each instruction could be measured and modeled at each pipeline stage. The original intention of energy consumption estimation of instruction-level is to minimize the energy cost of the software during the compilation phase so that the whole system could achieve the energy efficiency. Thus, energy consumption estimation is often considered to be the first step towards software energy minimization [2].

The goal of this paper is to provide a fast accurate power consumption estimation for a given application on embedded system in the early design stage. We present some energy models to estimate energy consumption and integrate the model into a widely used simulator Keil. Besides, we use Keil system analyzer processor to estimate energy consumption and validate the proposed model.

The paper organization is as follows: Section II analyzes factors causing energy dissipation in an embedded processor from different abstraction levels. Section III analyses the merits and drawbacks of these energy estimation approaches. Section IV presents the energy consumption results using the base cost for ARM Cortex M4. Section V concludes the study, and proposes some suggestions for future work.

II. FACTORS CAUSING POWER CONSUMPTION IN PROCESSORS

Factors causing energy consumption could be interpreted from different abstraction levels. From the view of the digital circuit, we briefly explain the energy consumption in a processor. Then, we analyze main energy consumers on a processor as well as off-chip energy consumers on an embedded system. Finally, we summarize some major factors of processor design and runtime environment that could affect energy consumed by a processor.

A. Circuit Level

The total energy or power consumption of a digital Complementary Metal Oxide Semiconductor (CMOS) circuit consists of two components, namely, static energy dissipation and dynamic energy dissipation. It can be expressed as:

$$E = E_{static} + E_{dynamic} \quad (1)$$

where E_{static} is the energy dissipation that occurs when CMOS is in standby mode, generally regarded as waste of energy. Leakage current causes static dissipation [3]. However, for properly designed CMOS processor, this static dissipation is often insignificant but it becomes significant when the transistors become smaller and faster [4]. The dynamic part of energy dissipation accounts for most of the energy dissipation on CMOS circuit. It is composed of two terms:

$$E_{dynamic} = E_{swiching} + E_{short-circuit} \quad (2)$$

where $E_{swiching}$ is energy consumed when the load capacitance are charged and discharged associated with each node in CMOS circuit. It is given by:

978-1-7281-1740-9/19/$31.00 ©2019 IEEE

$$E_{swiching} = \frac{1}{2} C V_{dd}^2 \quad (3)$$

where C is the effective load capacitance, and V_{dd} is the supply voltage of the circuit.

B. Functional level

First functional unit is data path that includes a group of functional units, such as arithmetic logic units or multipliers that perform data processing operations, registers [5]. Any operation in data path can reflect energy consumption because it causes switching activity in that particular component contributing to the dynamic energy consumption [6].

Cache is the fastest and the smallest on-chip memory to a processor. When reading/writing request comes, row and column decoder into both row and column select signals. After that, address is decoded into a set of word lines. Word lines select row to be read/written. Apparently, all of these procedures incur energy consumption due to the dynamic energy dissipation [7]. In [8], cache is regarded as one of the greatest energy consumer on processor, which accounts for 43% of total energy consumption.

Energy consumption can be influenced by design details of the processor. Addressing model influences the energy consumption because the variation of addressing model has different duration of calculating memory address and using different number of registers. The register file is an array of registers on a processor, a number of register is allocated while executing an application. The energy consumption increases with increasing clock frequency [9]. The dynamic power consumed by a processor is approximately proportional to the CPU frequency, and to the square of the processor voltage.

$$P = C\, V^2\, f \quad (4)$$

Thus, some techniques such as dynamic frequency scaling could be employed to adjust the frequency of the processor "on the fly" so that energy consumption and heat dissipation could be decreased.

There are two main energy consumers for an embedded system, which are outside of the processor. The first one is the bus that is responsible for transferring data between processor and cache, cache and memory or memory and peripheral. The second one is main memory. Memory access is an energy hungry operation since external memory accessing which incurs longer duration of latency and execution time has lower speed than cache. Another paper claims that the statistical model may not be particularly obvious in some circumstances [10]. To overcome this problem, their energy consumption estimation model considers activity factors and architectural statistics such as macro-instructions-per-cycle, instructions-fetched-per-cycle, and number of loads, number of stores, number of branch instructions, and number of floating-point instructions. In addition, this cannot be used to estimate unknown architecture or technology [11]. Lee, Sheayun et al. [12] present energy model that aims to estimate the energy consumed in each clock cycle so that all energy consumed at all pipeline stages can be summed. Their energy model can reset the static dissipation as base cost, the number of bits flipped, and the number of logical 1's as dynamic dissipation.

III. MEASUREMENT HARDWARE

External measurements refer to use a device that can measure current drawn or power. For example, in [12] the authors used dual-slope integrating digital ammeter to measure MCU and DRAM subsystem current. Another study [13] employs digitizing oscilloscope to measure voltage difference. Study [14] also uses oscilloscope that records power value in Comma Separated Values (CSV) files. Studies [15],[16], [17] developed a dedicated device that can monitor the current drawn in every clock cycle. They are the key device to monitor the energy consumed for instructions at given pipeline stage.

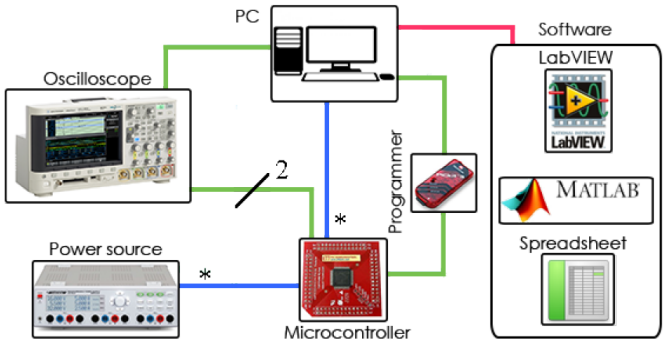

Figure 1. Measurements system for power estimation.

The monitor chip reads the energy consumed by the processor and external memory at micro seconds rate. In addition, some off-the-shelf energy measurement boards use power monitor chip. As mentioned earlier, energy measurement chip can be used to monitor energy consumed on the target processor. The current, voltage, and power data can be measured, and recorded at 2,000,000 samples/s. As shown in Figure 1, the measurement results can be viewed through energy monitoring graphics that can be drawn using Excel.

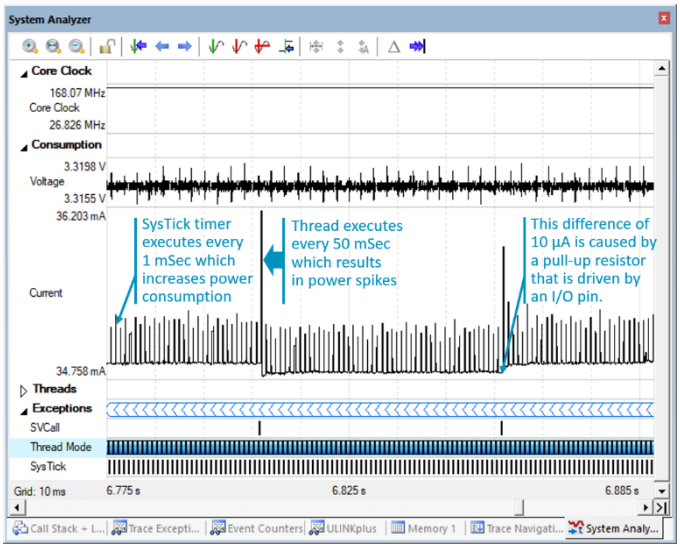

Figure 2. Power estimation using Keil Simulator and System Analtzer.

The main purpose of using the measurement device is to obtain either energy parameters or base energy cost for the energy model. For this, we have used Keil simulator system analyzer tool that is designed to evaluate the performance of embedded processors [18]. This tool also presents energy measurement infrastructure, which measures the energy consumption of most embedded processors by continuously monitoring the current drawn, voltage and power for execution of the processor. Considering the dedicated energy consumption benchmark applications and high accuracy measurement device, we believe their measurement result is reliable. Thus, we decide to use the power dissipation of each instruction category that is obtained from system analyzer as shown in Figure 2.

IV. EXPERIMENTAL WORK AND MODEL VERIFICATION

Directly measuring the energy dissipation on the target processor is the best way to validate the proposed energy estimation approach. In Table 1, the MCU metadata and technical data is presented. We have implemented the base energy cost in the Keil simulator to estimate energy consumption for ARM Cortex M4. We also use system analyzer for energy consumption estimation. The results of base energy cost estimation for ARM Cortex M4 shows that the number of instructions is appropriately proportional to energy consumption. The variance between each instruction category is small.

TABLE I. MCU METADATA AND TECHNICAL DATA.

MCU	***Producer***	***IDE***	***Clock source***	f_{osc}	***Voltage***
TM4C123	TI	Keil	Internal, PLL	120 Mhz	3.3 V

In this study, self-balancing mobile robot is the controlled device. This robot has TM4C123GH6 MCU, two DC motors, and motor drivers. TM4C123GH6 is produced by TI and this 32-bit MCU uses ARM Cortex M4 core [19].
In Table 2, the benchmark programs used in the experiments are described. The column "Lines executed" presents the total result of the profiling of the program. Finally, the column "Atomic operations" are the total number of atomic-operations in the benchmark.

TABLE II. BENCHMARK PROGRAMS.

Benchmark Programs	***Description***	***Lines executed***	***Atomic Operations***
PI	Proportional Integral Control	374	1292
PID	Proportional Integral Derivative Control	723	2557

As can be seen from Figure 3 and 4, the relationship between estimated energy consumption and the number of instructions are almost linear for both control algorithms. One of the reasons is that the value of the energy parameters is too small compared to the scale of number of instructions. Another reason is that the standard deviation of energy per instruction for each category of PI and PID are 3.1233e-05 and 1.2943e-04 respectively, which indicate that energy per instruction for both control algorithm are very close to each other and hence to the mean value. Consequently, the energy model for both control algorithms can be regarded as a linear function.
We conclude by reviewing Figure 3 and 4 that the number of cycles does not seem to be proportional to the number of instructions. In other words, the relationship between the number of cycles and the number of instructions are non-linear.

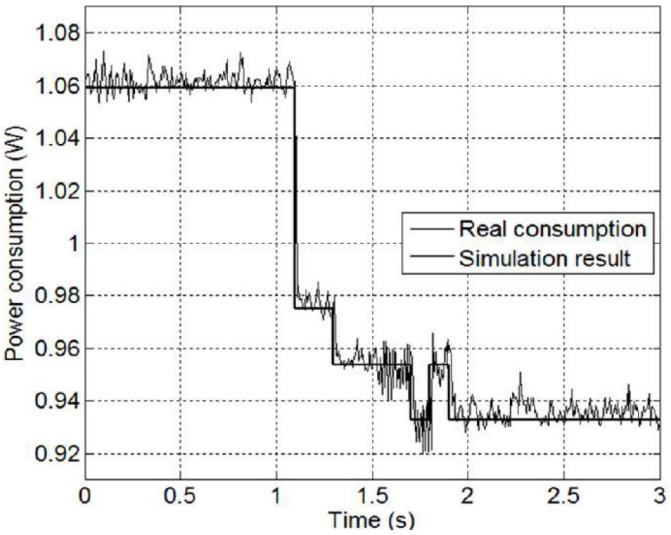

Figure 3. PI control program power estimation.

Moreover, the total cycle counts for a given application is effected not only the number of instructions but also by pipeline stall. Besides branch faulty prediction, memory operation, cache miss hit and pipeline hazards may cause pipeline stall. Hence, pipeline stall results that the execution time is difficult to predict. Therefore, based on above analysis, only the number of instructions cannot solely determine the cycle counts for a given application in ARM Cortex.

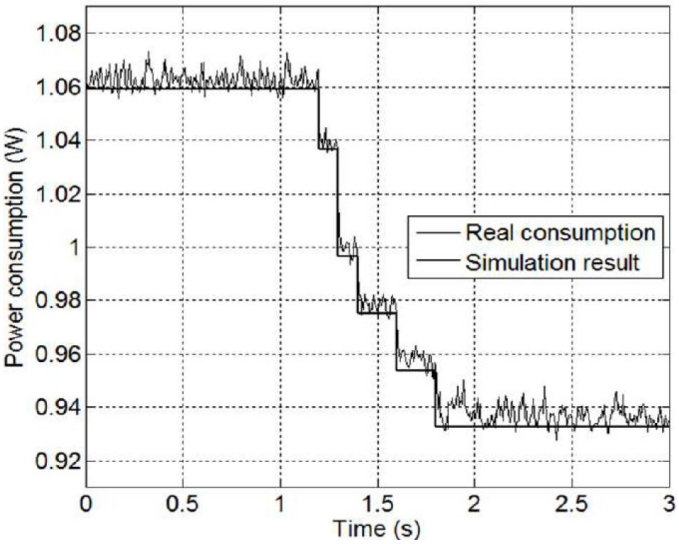

Figure 4. PI control program power estimation.

The number of instructions is appropriately proportional to energy consumption on processors. There is indirect relationship between cycle counts and energy consumption on processors. The number of instructions cannot solely determine the cycle counts. Because of pipeline stalls, which can be caused by faulty branch prediction, cache miss hit, and pipeline hazard will lead to spending more clock cycles to complete task on processors. Finally, instruction-level energy consumption estimation is the best choice for the compiler energy-efficiency optimizations and instruction simulation-based energy consumption estimation. Therefore, instruction-level energy is irreplaceable in future energy estimation technological developments.

V. CONCLUSION

This paper discussed methodologies on energy consumption estimation for ARM Cortex M4. We found that instruction-level energy estimation is the research hotspot in all abstraction levels of energy consumption estimation. Compared with energy-consumption estimation approach at other abstraction levels, instruction-level energy estimation has high accuracy, retarget ability, and low-complexity. Thus, we believe that future third party commercial energy estimation tools could depend on instruction-level energy consumption.

REFERENCES

[1] H. Blume, D. Becker, L. Rotenberg, M. Botteck, J. Brakensiek, and T. G. Noll, "Hybrid functional- and instruction-level power modeling for embedded and heterogeneous processor architectures," *J. Syst. Archit.*, vol. 53, no. 10, pp. 689–702, Oct. 2007.

[2] E.-Y. IEEE Circuits and Systems Society., L. Benini, and G. De Micheli, *1999 IEEE/ACM International Conference on Computer-Aided Design : digest of technical papers : November 7-11, 1999, San Jose, California*. IEEE Computer Society Press, 1999.

[3] M. Salehi, A. Ejlali, and M. Shafique, "Run-Time Adaptive Power-Aware Reliability Management for Manycores," *IEEE Des. Test*, vol. 35, no. 5, pp. 36–44, Oct. 2018.

[4] Q. Tang, Á. M. Groba, E. Juárez, C. Sanz, and F. Pescador, "Real-time power-consumption control system for multimedia mobile devices," *IEEE Trans. Consum. Electron.*, vol. 62, no. 4, pp. 362–370, Nov. 2016.

[5] D. Notkin, B. H. C. Cheng, and K. Pohl, *2013 35th International Conference on Software Engineering (ICSE) : proceedings : May 18-26, 2013, San Francisco, CA, USA*. IEEE Press, 2013.

[6] I. Kadayif, M. Kandemir, G. Chen, N. Vijaykrishnan, M. J. Irwin, and A. Sivasubramaniam, "Compiler-directed high-level energy estimation and optimization," *ACM Trans. Embed. Comput. Syst.*, vol. 4, no. 4, pp. 819–850, Nov. 2006.

[7] S. Lee, A. Ermedahl, and S. L. Min, "An Accurate Instruction-Level Energy Consumption Model for Embedded RISC Processors," *ACM SIGPLAN Not.*, vol. 36, no. 8, pp. 1–10, 2005.

[8] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar, "A 5-GHz mesh interconnect for a teraflops processor," *IEEE Micro*, vol. 27, no. 5, pp. 51–61, Sep. 2007.

[9] M. Bazzaz, M. Salehi, and A. Ejlali, "An accurate instruction-level energy estimation model and tool for embedded systems," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 7, pp. 1927–1934, Jul. 2013.

[10] M. D. Powell, A. Biswas, J. S. Emer, S. S. Mukherjee, B. R. Sheikh, and S. Yardi, "CAMP: A technique to estimate per-structure power at run-time using a few simple parameters," in *Proceedings - International Symposium on High-Performance Computer Architecture*, 2009, pp. 289–300.

[11] G. F. Riley, F. Quaglia, J. Himmelspach, S.-I. Institute for Computer Sciences, and ACM Digital Library., *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques : SIMUTools 2012 : 19-23 March 2012, Desenzano del Garda, Italy*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012.

[12] H. Blume, D. Becker, M. Botteck, J. Brakensiek, and T. G. Noll, "Hybrid functional and instruction level power modeling for embedded processors," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, vol. 4017 LNCS, pp. 216–226.

[13] J. T. Russell and M. F. Jacome, "Software power estimation and optimization for high performance, 32-bit embedded processors," in *Proceedings International Conference on Computer Design. VLSI in Computers and Processors (Cat. No.98CB36273)*, 2002, pp. 328–333.

[14] G. S. P. Delicia, T. Bruckschloegl, P. Figuli, C. Tradowsky, G. Marchesan, and J. Becker, "Bringing Accuracy to Open Virtual Platforms (OVP): A Safari from High-Level Tools to Low-Level Microarchitectures," *Int. J. Comput. Appl.*, pp. 22–27, 2013.

[15] S. Nikolaidis, N. Kavvadias, T. Laopoulos, L. Bisdounis, and S. Blionas, *Journal of embedded computing.*, vol. 1, no. 3. IOS Press, 2005.

[16] S. Lee, A. Ermedahl, S. Min, and N. Chang, "Statistical Derivation of an Accurate Energy Consumption Model for Embedded Processors," vol. 16, pp. 0–19, 2002.

[17] V. Parikh, C. Desai, D. Joshi, and G. Nagababu, "Estimation of electricity generation potential by solar radiation on Sardar Sarovar dam," *Energy Procedia*, vol. 158, pp. 167–172, Feb. 2019.

[18] "µVision User's Guide: System Analyzer." [Online]. Available: http://www.keil.com/support/man/docs/uv4/uv4_db_dbg_systemanalyzer.htm. [Accessed: 06-Apr-2019].

[19] Y. Bai and Z. S. Roth, "Introduction to Tiva C MCU LaunchPad™—TM4C123G," in *Advances in Industrial Control*, Springer, Cham, 2019, pp. 35–99.