*Journal of Embedded Systems, 2017, Vol. 4, No. 1, 7-12*
Available online at http://pubs.sciepub.com/jes/4/1/2

DOI:10.12691/jes-4-1-2

# A Simplified Software Energy Consumption Estimation for Embedded System

**V.A. Kulkarni[1,*], G.R. Udupi[2]**

E&C Engineering, GIT, Belgavi, India
*Corresponding author: vak@amgoi.edu.in

**Abstract** Software energy in an embedded system is the energy consumed while running the software. As the embedded system is expected to execute the task repeatedly, software energy is the major component in total energy consumption of an embedded system. For battery powered embedded applications, energy consumption estimation is very important. Different methods are available for software energy estimation. A simple method of software energy estimation with good accuracy is presented in this paper. Results are validated with micro benchmark programs. Accuracy of 2.01% to -5.83% is obtained for different micro benchmark programs. Also equation for instructions using immediate data is developed. $R^2$ value of 0.9 is obtained.

***Keywords:*** *current measurement, curve fitting, embedded software, energy consumption, energy estimation*

**Cite This Article:** V.A. Kulkarni, and G.R. Udupi, "A Simplified Software Energy Consumption Estimation for Embedded System." *Journal of Embedded Systems*, vol. 4, no. 1 (2017): 7-12. doi: 10.12691/jes-4-1-2.

## 1. Introduction

Many of the embedded applications are battery dependent i.e. they use battery as energy source. The total energy consumption of an embedded application is because of hardware and software. Once we know the amount of energy consumption, we can estimate battery life. Alternatively, energy efficient software can be used once we know the software energy. Energy consumption model of a processor software can be classified as low level models (hard ware models) and high level models. Different abstraction levels at low level models are: logic gate level, RT level and architectural level. High level models uses instructions and functional units from software perspective. Instruction Level Power Analysis (ILPA) and Functional Level Power Analysis (FLPA) are the classifications of high level power estimation models. In ILPA, each instruction is assigned with base cost (energy required to execute instruction), inter instruction cost (when two instructions are executed, total cost may be more than sum of base costs of both instructions). Other energy sensitive factors also need to be considered. Thus software power is sum of base cost and inter instruction cost of all instructions used in the program.

This paper presents a simplified method for estimation of software energy for an ARM Cortex M4 processor based embedded system. The effect of immediate value in an instruction affects the energy consumption. Polynomial equation which defines instruction energy consumption based on number of 1's in the immediate value is obtained. $R^2$ value ranging from 0.76 to 0.96 is obtained for different instructions.

This paper is organised as follows: literature review is presented in Section 2. In Section 3, measurement method for software energy estimation is explained. The effect of number of 1's in an instruction is discussed in Section.4. Micro bench mark program and validation of energy estimation is given in Section 5. Conclusions of the present work is addressed in Section 6.

## 2. Literature Review

An instruction level power analysis was first proposed by Tiwari et al. [1]. As the voltage in most of the embedded applications is fixed, instruction power can be calculated by finding the current taken by the instruction. Loop containing the same instruction is executed to get stable reading and to minimize the effect of branching.

Cycle accurate measurement is carried out by Chang et al [2]. Charge transfer using switched capacitor method is used. Voltage across a capacitor is measured before and after instruction execution, the difference in voltage level in indication of instruction energy.

Bazzaz et al. [3] ignored the inter instruction effect as it is around 5% of base cost. MiBench benchmark tool is used for result validation. Error less than 6% in energy estimation is reported.

Wang et al. [4] also ignored inter instruction effect. This saves calculation and number of measurements to be taken. Loop consisting of 2000 instructions is considered. For resultvalidation, six benchmark programs were considered.

Lubomir et al. [5] considered microprocessor a black box. Current is measured with shunt resistor. Each instruction is repeated 1326 times to get stable reading. Relative error of 5% is reported.

## 3. Measurement Method

If the average current taken for executing an instruction is known, then instruction energy will be multiplication of current, voltage and time required for execution. ARM Cortex-M4 based microcontroller is used for experiment. On-board current measurement circuit is used which increases accuracy of measurements and overcomes many of limitations of current measurement mentioned in literature. It consists of a MAX9634T current monitor chip and a 12-bit ADC with a 12- bit sample at 50k to 200ksps. The MAX9634 multiplies the sense voltage by 25 to provide a voltage range suitable for the ADC to measure. On board current measurement is used for energy calculation. The ARM Cortex-M4 is a 32-bit core with 3 stage pipeline and Harvard architecture. Sample rate of 200ksps (5us period) is chosen for all measurements. Average current for a period of 1 second is considered for energy calculation. To find base cost, each instruction is executed 1000 times in a loop. This minimizes the effect of “BL loop” instruction on base cost. Calculation of inter instruction cost involves lot of measurements. Number of measurements is given by [n(n-1)/2]. Where ‘n’ is number of instructions in Instruction Set Architecture. For a microcontroller with 100 instructions, 4950 combinations of measurements to be carried out to find inter instruction cost. This large volume of measurement is tedious and time consuming. To overcome this problem, some researchers used NOP to find inter instruction cost i.e. NOP is executed with target instruction. With this the measurements for inter instruction cost reduces to ‘n’ only. This approximation saves time and resources. In some case, inter instruction cost is less than 5%, hence it is neglected [3]. In certain case, it is found to be between 14% and 48% [6]. The total energy is taken as sum of static energy (overall energy consumption of plat form with core and other peripherals in idle state), base energy, inter instruction energy and penalty due to resource constraints. From our experiments it is found that except base cost, all other costs put together works out to be 20%. This 20% has been taken care in estimated energy. It will simplify the process of estimation to a great extent. The various issues related to Instruction Level Power Analysis like current measurement method, inter instruction cost, etc. is discussed in [8].

## 4. Effect of Number of 1’s in an Instruction

One of the energy sensitive factor is the number of 1’s in an instruction. Following are the instructions in ARM Cortex M4 based microcontroller which use 32 bit data. ADC, ADD, AND. BIC, CMN, CMP. EOR, MOV, MVN, ORN and ORR. Experiments conducted with immediate data containing various combinations of number of 1’s. Current taken by processor core (in mA) is measured when the data is #0x00000000, #0x00000001, #0x00000003, #0x00000F00, #0x0000FF00, #0xAAAAAAAA, and #0xFFFFFFFF which corresponds to number of 1’s as 00, 01, 02, 04, 08, 16 and 32.

The current taken by the core in mA for different combinations of number of 1’s for the instructions is shown in Figure 1.

Vertical axis represents current taken by the processor core in mA. It can be seen that the variation in current for different number of 1’s is very small and can be neglected. The difference between minimum value and maximum value of current and % difference is shown in Table 1.

**Table 1. % difference between min. and max. value**

| | Max | Min | % diff |
|---|---|---|---|
| ADC | 3.495 | 3.212 | 8.097282 |
| ADD | 3.443 | 3.212 | 6.709265 |
| AND | 3.286 | 3.2 | 2.617164 |
| BIC | 3.294 | 3.209 | 2.580449 |
| CMN | 3.319 | 3.23 | 2.681531 |
| CMP | 3.321 | 2.59 | 22.01144 |
| EOR | 3.526 | 3.205 | 9.1038 |
| MOV | 3.317 | 3.243 | 2.230932 |
| MVN | 3.33 | 3.253 | 2.312312 |
| ORN | 3.301 | 2.966 | 10.14844 |
| ORR | 3.296 | 3.21 | 2.609223 |

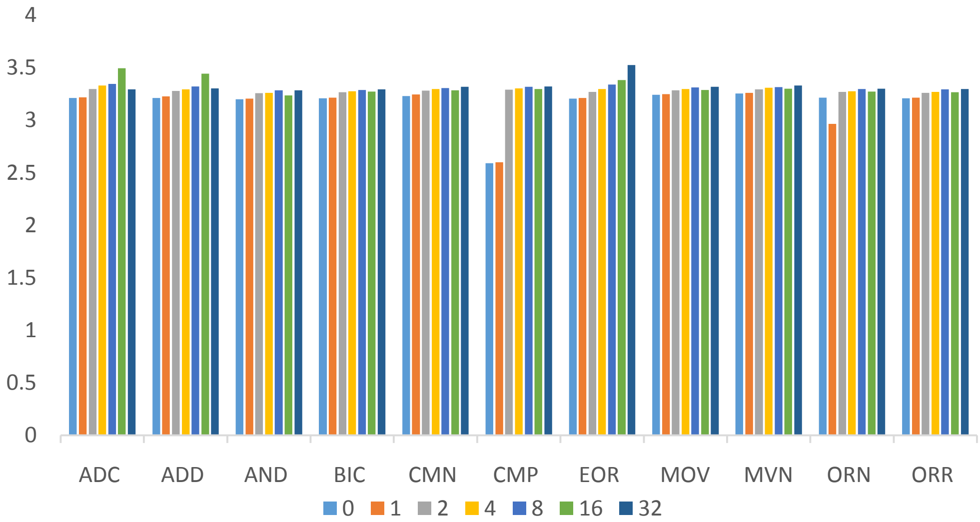


**Figure 1.** Effect of number of 1’s

Polynomial equations are derived to represent each instruction given above. $R^2$ is a statistical measure of closeness of data with fitted regression line. Its value is between 0% and 100%. Higher the $R^2$ value, the better the model fits the data. $R^2$ value of each equation is found. Except for ORN instruction, all other instructions can be expressed in polynomial equation form. This equation can be used to find current and hence energy, when number of 1's is known.

**Table 2. Polynomial equation for instructions**

| Instruction | Polynomial equation | $R^2$ |
|---|---|---|
| ADC | $y = -0.0008x^2 + 0.0296x + 3.21$ | 0.9142 |
| ADD | $y = -0.0006x^2 + 0.0231x + 3.2121$ | 0.9272 |
| AND | $y = 4E\text{-}05x^3 - 0.0021x^2 + 0.0248x + 3.1973$ | 0.9238 |
| BIC | $y = 3E\text{-}05x^3 - 0.0017x^2 + 0.0221x + 3.209$ | 0.9048 |
| CMN | $y = 4E\text{-}05x^3 - 0.0017x^2 + 0.0215x + 3.2326$ | 0.9561 |
| CMP | $y = 0.0003x^3 - 0.0156x^2 + 0.212x + 2.6224$ | 0.7633 |
| EOR | $y = -0.0001x^2 + 0.0136x + 3.2219$ | 0.9652 |
| MOV | $y = 3E\text{-}05x^3 - 0.0016x^2 + 0.0198x + 3.2412$ | 0.9541 |
| MVN | $y = 3E\text{-}05x^3 - 0.0014x^2 + 0.0181x + 3.2529$ | 0.9469 |
| ORN | $y = 6E\text{-}05x^3 - 0.0032x^2 + 0.0449x + 3.1166$ | 0.377 |
| ORR | $y = 4E\text{-}05x^3 - 0.0018x^2 + 0.0228x + 3.2076$ | 0.9479 |

## 5. Energy Estimation and Validation

To find base cost, measurements are carried out as explained in section 2. Cortex-M4 instruction set can be divided in to 9 groups: memory access, data processing, multiply and divide, saturating, packing and unpacking, bit field, branch and control, floating point and miscellaneous instructions [13]. The base cost for different instructions are shown from Figure 2 to Figure 6. These results are used to estimate software energy. In all these figures, x axis indicates current in mA.

To validate the results, six micro benchmark programs are used. Each micro benchmark consisting of different proportion of instructions. Figure 7 shows the composition of types of instructions in six micro benchmark programs.

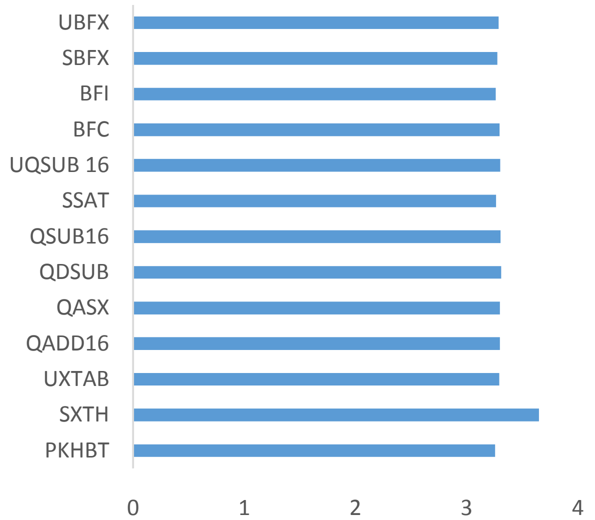


**Figure 2.** Base Cost of Saturating, Packing & Bit field Instructions

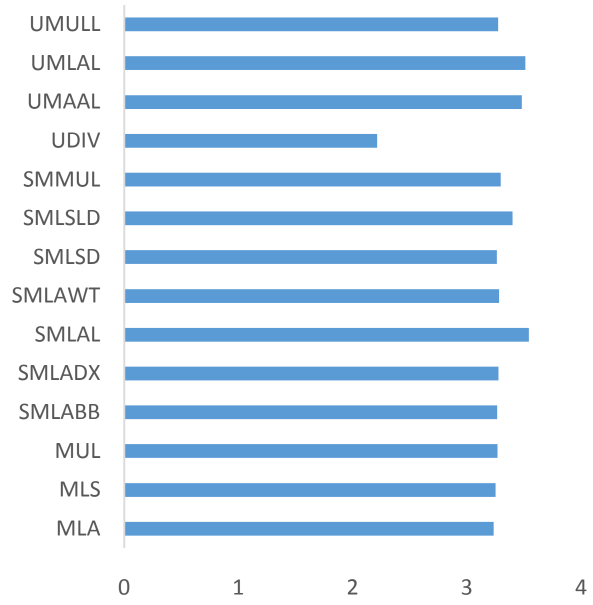


**Figure 3.** Base Cost of Multiply & Divide Instructions

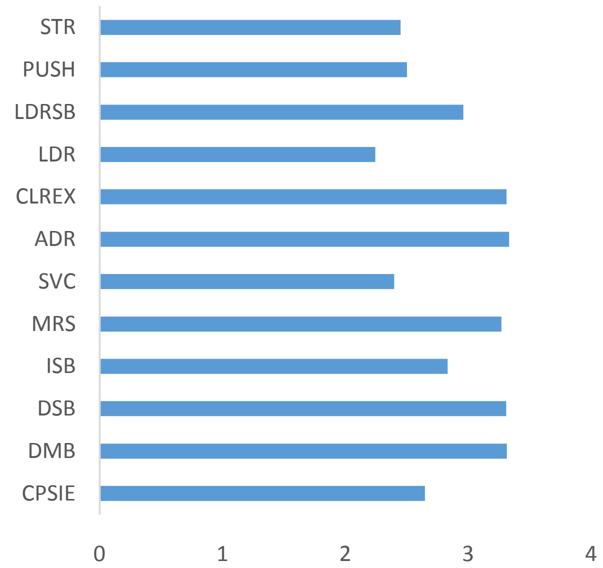


**Figure 4.** Base Cost of Memory Access & Miscellaneous Instructions

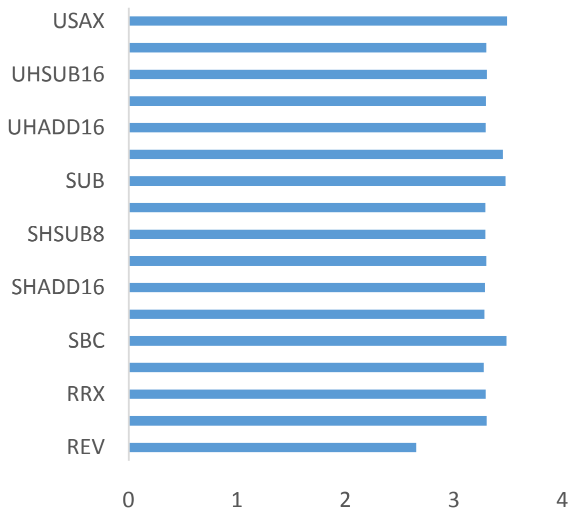


**Figure 5.** Base Cost of Data Processing Instructions

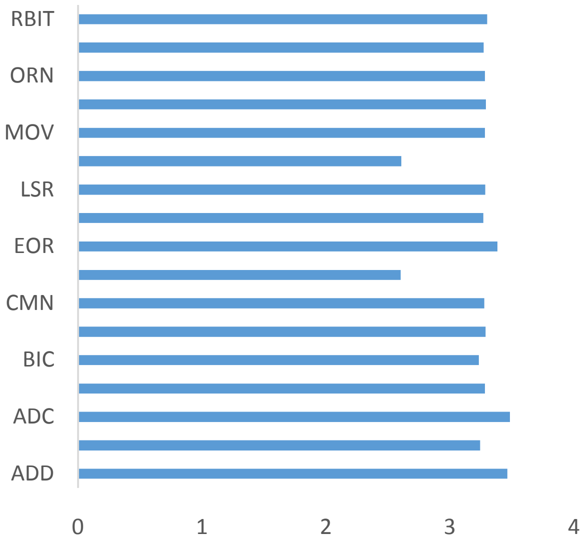


**Figure 6.** Base Cost of Data Processing Instructions

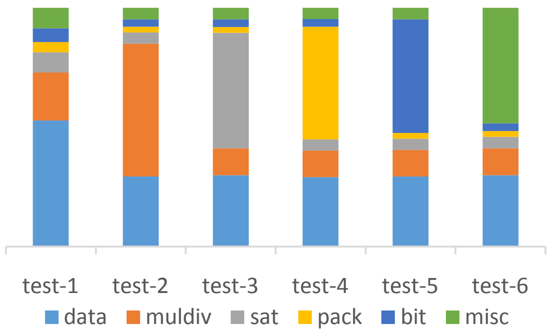


**Figure 7.** Composition of instruction types

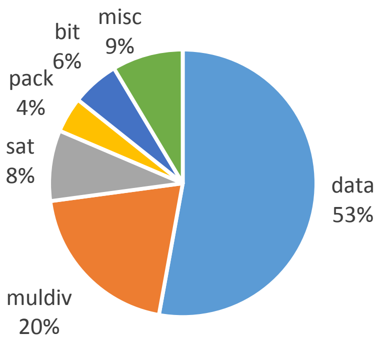


**Figure 8.** Composition of instructions in Test1

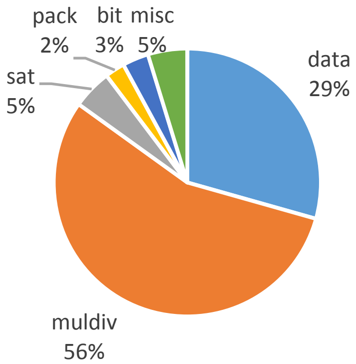


**Figure 9.** Composition of instructions in Test2

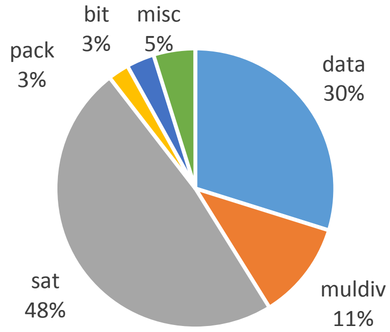


**Figure 10.** Composition of instructions in Test3

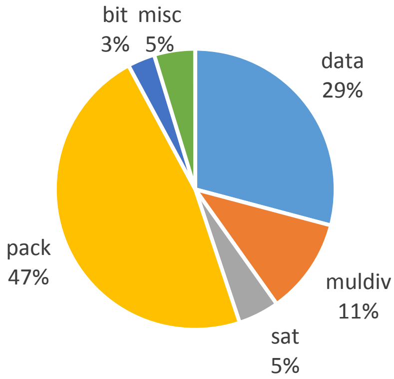


**Figure 11.** Composition of instructions in Test4

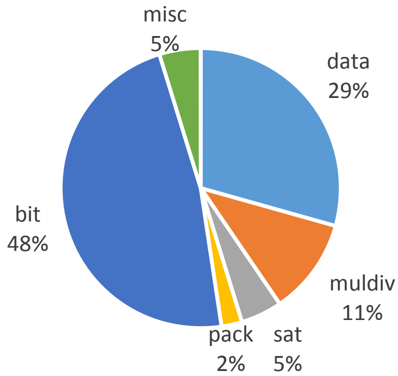


**Figure 12.** Composition of instructions in Test5

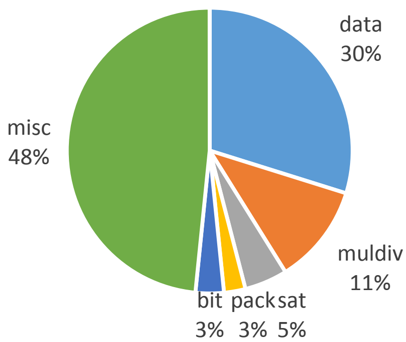


**Figure 13.** Composition of instructions in Test6

% composition of each type of instructions in micro bench mark program 1 to 6 is shown from Figure 8 to Figure 13.

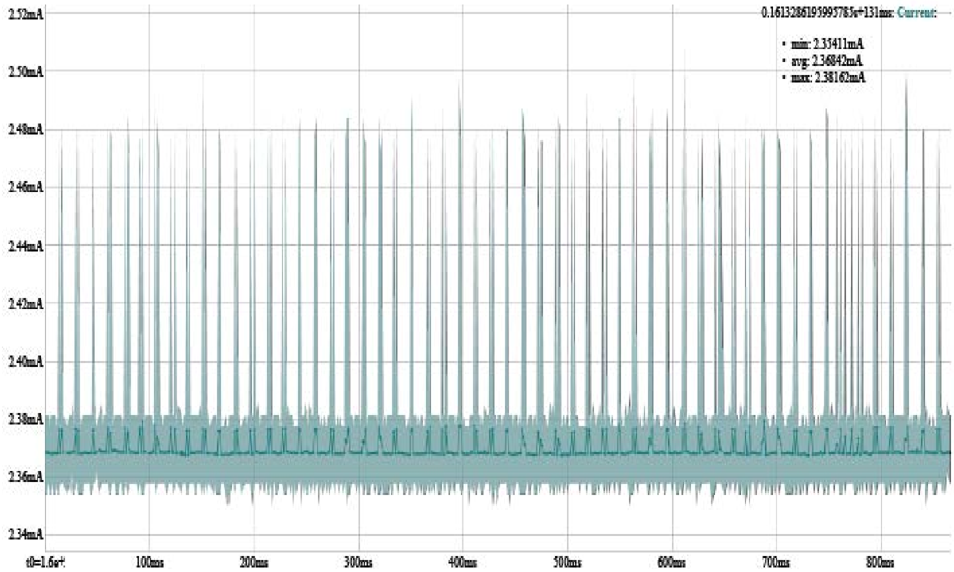


**Figure 14.** Current waveform in Test 1

While measuring current taken to execute each micro bench mark program, sample rate of 200ksps (5us period) is chosen for all measurements. Average current for a period of 1 second is considered for energy calculation. Current waveform observed during execution of micro bench mark program 1 (Test 1) is shown in Figure 14.

Calculation of actual energy consumption for micro bench mark program-1 (Test 1) is given in Table 3.

**Table 3. Energy consumption for Test1**

| I (mA) | V (Volts) | Time Period (uS) | Total no. of cycles | Actual energy (nJ) |
|---|---|---|---|---|
| 2.367 | 3.3 | 0.0833 | 83 | 54.00516 |

Similar calculations carried out for other micro bench mark programs. Table 4 shows the estimated energy consumption and actual energy consumption for six micro bench mark programs (Test1 to Test 6).

**Table 4. Estimated and actual energy consumption**

| | Estimated energy (nJ) | Actual energy (nJ) | % error |
|---|---|---|---|
| Test1 | 54.93666 | 54.00516 | 1.724821 |
| Test2 | 102.9734 | 100.9382 | 2.016204 |
| Test3 | 94.07154 | 95.0773 | -1.05784 |
| Test4 | 97.53889 | 103.5868 | -5.83849 |
| Test5 | 95.32394 | 99.67786 | -4.368 |
| Test6 | 106.9107 | 112.6601 | -5.10335 |

As can be seen from Table 4, the estimated energy consumption of software is very close to actual energy consumption. These results confirm estimation and measurement of proposed method.

## 6. Conclusion

A simplified method of software energy estimation is presented. With some approximations, backed by experimental results, lengthy and tedious calculations can be avoided. A fairly good value of accuracy is achieved. The results are to be validated with different benchmark covering diverse areas of applications.

## References

[1] Tiwari, S. Malik, and A. Wolfe, “Power analysis of embedded Software: A first step towards software power minimization,” IEEE Trans. VLSI Systems, vol. 2, no. 4, pp. 437-445, Dec. 1994.

[2] Naehyuck Chang, Kwanho Kim, and Hyung Gyu Lee, “Cycle-Accurate Energy Measurement and Characterization with a Case Study of the ARM7TDMI”, IEEE Trans. VLSI Systems, vol. 10, no. 2, April 2002, pp. 146-154.

[3] Mostafa Bazzaz, Mohammad Salehi and Alireza Ejlali, “An Accurate Instruction-Level Energy Estimation Model and Tool for Embedded Systems”, IEEE Trans. Instrum. Measurement, VOL. 62, NO. 7, JULY 2013, pp. 1927-1934.

[4] 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.

[5] 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.

[6] Momcilo V. Krunic, Miroslav V. Popovic, Vlado M. Krunic, Nenad B. Cetic, “Energy Consumption Estimation for Embedded Applications”, ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 22, NO. 3, 2016, pp. 44-49.

[7] V.A.Kulkarni and Dr. G.R.Udupi, “A Simplified Method for Instruction Level Energy Estimation for Embedded System”, EJERS, European Journal of Engineering Research and Science Vol. 2, No. 5, May 2017.

[8] V.A.Kulkarni and Dr. G.R.Udupi, “Instruction Level Power Consumption Estimation – Issues and Review”, Journal of Multidisciplinary Engineering Science and Technology (JMEST), Vol.4, Issue 2, Feb. 2017, pp 6776-6781.

[9] Nikolaos Kavvadias, Periklis Neofotistos, Spiridon Nikolaidis, C. A. Kosmatopoulos, and Theodore Laopoulos, “Measurements Analysis of the Software-Related Power Consumption in Microprocessors”, IEEE Trans. Instrum. Measurement, VOL. 53, NO. 4, AUGUST 2004, pp. 1106-1112.

[10] V. Konstantakos, A. Chatzigeorgiou, S. Nikolaidis, T. Laopoulos, “Energy consumption estimation in embedded systems”, IEEE Trans. Instrum. Measurement, VOL. 57, NO. 4, APRIL 2008 pp. 797-804.

[11] Theodore Laopoulos, Periklis Neofotistos, C. A. Kosmatopoulos, and Spiridon Nikolaidis, "Measurement of Current Variations for the Estimation of Software-Related Power Consumption", IEEE Trans. Instrum. Measurement, VOL. 52, NO. 4, AUGUST 2003, pp. 1206-1212.

[12] N. Kavvadias, P. Neofotistos, S. Nikolaidis, K. Kosmatopoulos, and T. Laopoulos, "Measurements analysis of the software-related power consumption of microprocessors," IEEE Trans. Instrum. Measurement, vol. 53, no. 4, pp. 1106-1112, Aug. 2004.

[13] Cortex-M4 Devices, Generic User Guide, ARM Limited, 2010.

[14] F. N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," IEEE Transactions on VLSI Systems, vol.2, pp. 446-455, 1994.

[15] 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.