International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

Energy Efficient C Code for ARM based Embedded System

Kaustubha Shah [1], Kshama S.[2] K.A.Nethravathi [3]

[1]*Student, R.V. College of Engineering*
[2]*Student, R.V. College of Engineering*
[3]*Professor, Dept. of ECE, R.V. College of Engineering, Bengaluru - 560059, Karnataka, India*

Abstract - *Recent years have seen rapid growth in research targeting at reducing energy consumption in the code. The potential energy savings achievable through optimization of software running on microprocessor [11]. This paper presents an overview of the techniques used in our work through code compilation. And the power reduction techniques as well that can be beneficial in this regard are reviewed. Deployment of system in challenging geographical areas where energy sources could not exist is a problem that the system has to cater. Portable devices designed with Size, Weight and Power (SWaP) constrained attributes is the solution. Furthermore, major work for low power embedded systems in the past has been done in VLSI domain and not on source codes.*

Key Words: **Power optimization, SWaP, Code optimization, ICT, Power optimized, Power un-optimized**

1. INTRODUCTION

Today, Information and Communication Technologies (ICT) amounts for 10% of the world energy [6] and 3% of the overall carbon footprint [7], which will keep growing in future [5]. Embedded computer systems are characterized by the presence of a dedicated processor which executes application specific software [4]. Application specific logic is getting increasingly expensive to manufacture and is the solution only when speed constraints rule out programmable alternatives. The solution to the above issue is met with software programmable solutions. A large number of embedded computing applications are power critical, which further helps in power estimation and low power design. Power efficient design of battery powered embedded systems demands optimizations in both hardware and software [1]. Hence accurate power consumption models help explore compiler and source code optimizations to reduce Power consumption which is vital for SWaP (Size, Weight and Power) constrained systems. ARM architecture-based processors provide a great balance between low power consumption and processing needs. Harnessing the architecture specific code optimization techniques in the software would aid in an optimized processing and power consumption platform. It is found that standard compiler optimizations give less than 1% energy savings and source code optimizations are capable of up to 90% energy savings [10].

1.1 System development

As embedded software often runs on processors with limited computation power, thus optimizing the code becomes a necessity. Hence different optimization techniques for C++ code are developed. These optimization techniques are sound coding practices. Hence, they are identified and implemented as code snippets. Power profiles are made both with and without implementation of practices, accordingly functions are created and optimized and simulated in Qt. And finally, libraries are made using these functions. The block diagram, shown in the Fig.1, depicts the major components and sequence of optimizing the code.

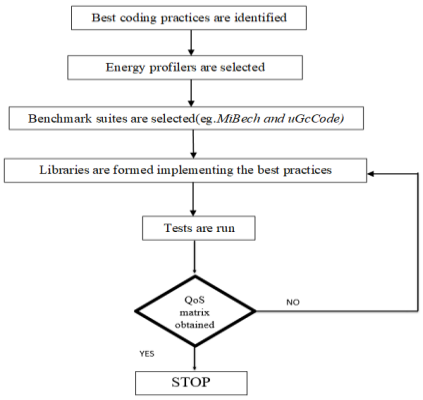

Fig.1: Block diagram of process flow

These functions were coded in Qt and power was profiled using Intel Power Gadget, analysis of the .csv file was done. The same methodology is mentioned in Figure 2.

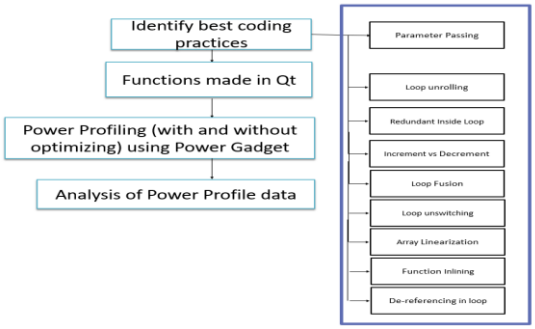

Fig. 2. Process Flowchart of implementation of functions

1.2 EXPERIMENTAL DETAILS

We leveraged Intel®Core™ i7-9750H CPU to conduct experiments. The processor peaks at 2.59 GHz. Intel® Power Gadget 2.0 is used to profile power used by the processor. Qt console from Qt creator is used to write, compile and run functions. The functions are looped one hundred thousand times to give near ideal average power. Both optimized and un-optimized functions are Power profiled, which would help identify the savings achieved graphically as well as quantitively.

2. RESULTS

The graphs in Fig. 3 shows a comparison between the power consumption of both the optimized and un-optimized functions. Red plots are optimised functions while blue plots for un-optimised functions.X-axis represents time in micro-seconds and Y-axis gives power consumed in Watts. Table 1 gives average power consumed by functions for both optimised and unoptimised runs.As it can be seen from power profiles for of th functions that some of the optimised functions have more peaks but finish off early hence they have less average power consumed while some of the functions have approximately same average power consumed for both optimised and un-optimised versions

Fig.3a Parameter Passing

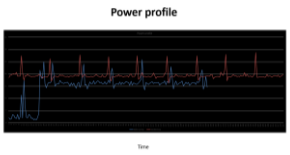

Fig.3b Loop Unrolling

Fig.3c Redundant inside Loop

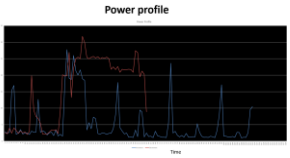

Fig.3d Inc v/s Dec

Fig.3e Loop Fusion

Fig.3f Loop Unswitching

Fig.3g Array linearization

Fig.3h Function inlining

Fig.3i De-referencing in loop

Parameter passing technique results in 10.4%, Loop unrolling results in 24.7%, Redundant inside loop results in 9.9%, Increment v/s Decrement gives 67.6%, Loop fusion results in 73%, Loop unswitching results in 6.2%, Array Linearization results in 0.2%, Function inlining 3.6%, Dereferencing in loop results in 0.9% of power optimization

Table 1:Average power consumed by functions

Functions	Simulation	
	Unoptimized(in W)	Optimized(in W)
Parameter passing	13.5	12.1
Loop unrolling	19.4	14.6
Redundant inside loop	16.2	14.6
Increment v/s Decrement	15.6	5.05
Loop fusion	17.8	4.8
Loop unswitching	11.3	10.6
Array linearization	9.8	9.78
Function inlining	21.4	20.63
Dereferencing in loop	21.8	21.6

3. CONCLUSIONS

As the software power efficiency has been still in its beginning stages as most of the power efficiency work focuses more on hardware part. There are very limited standards which can be followed to write power efficient code. This paper uncovers some of the ways at a granular level which will boost up the power efficient developing standards. We measured and compared the power efficiency of a techniques such as loop fusion, function in-lining, loop un-switching, parameter passing, loop unrolling, redundant inside loop, increment v/s decrement loop fusion, loop unswitching, array linearization, function inlining and dereferencing in loop.

We hope that these results will help software developers to build more power efficient C/C++ applications in future. This paper just scratches the surface of the field of software power efficiency. Our future work will focus on evaluating the power efficiency of more complex methods, data types in C++ and memory organization techniques for power efficiency. Researchers can also take a hint from this paper to explore other programming languages as well for power efficiency traits.

REFERENCES

[1] T. Simunic,L. Benini,G. De Micheli,M. Hans, "Source code optimization and profiling of energy consumption in embedded systems",in Proceedings 13th International Symposium on System Synthesis, Sept. 2000

[2] Yunsi Fei,Srivaths Ravi,Anand Raghunathan,Niraj K.Jha," Energy-optimizing source code transformations for operating system-driven embedded software", in IEEE Transactions on Embedded Computing Systems (TECS), Vol.2, Article No.: 2 , December 2007

[3] Chris Shore," Efficient C Code for ARM Devices", in ARM Technology Conference 2010,, Santa Clara CA, Session ATC-152, September 2010

[4]Vivek Tiwari,Sharad Malik,Andrew Wolfe,"Power Analysis of Embedded Software", in IEEE Transactions on Very Large Scale Integration(VLSI) Systems,Vol.2 No.4,December 2000

[5]Mohit Kumar,YouHuizi Lee,Weisong Shi,"Energy consumption in JAVA :An early experience", in Eighth International Green and Sustainable Computing Conference (IGSC), October 2017

[6] M.Mills,"The cloud begins with coal digital power group",2013.

[7] S. Mittal, "A survey of techniques for improving energy efficiency in embedded computing systems," in International Journal of Computer Aided Engineering and Technology, vol. 6, no. 4, pp. 440-459, 2014.

[8]Robert Michael Owens,Mary Jane Irwin,Debashree Ghosh,"Techniques for low energy software", in IE Proceedings of the 1997 international symposium on Low power electronics and design,August 2002

[9]Vivek Tiwari,Mike Tienchien Lee,"A memory allocation technique for low-energy embedded DSP software", in IEEE Symposium on Low Power Electronics. Digest of Technical Papers,October 2012

[10]W.Xu,A.Parikh,M.Kandemir,M.J.Irwin,"Fine grain instruction scheduling for low energy", in IEEE Workshop on Signal Processing Systems, October 2002

[11]Anish Muttreja,Anand Raghunathan,"Hybrid Simulation for energy estimation of embedded software",in IEEE Transactions on computer-aided design of integrated circuits and systems,Volume 26,No.10,October 2007

[12]G.Esakkimuthu,N.Vijay Krishnan,"Memory system energy:Influence of hardware-software optimisations", in Proceedings of the 2000 International Symposium on Low Power Electronics and Design (Cat. No.00TH8514) , July 2000

[13] V. Tiwari, S. Malik and A. Wolfe, "Compilation techniques for low energy: an overview," in Proceedings of 1994 IEEE Symposium on Low Power Electronics, San Diego, CA, USA, 1994, pp. 38-39.