Analysis of Power Management for Hard Real-Time Embedded System between Adaptive Power Management Method and Idle Energy Reduction Method

Chengming Li *

Department of Electrical, Computer and Energy Engineering, University of Colorado Boulder, Boulder, United States

*Corresponding author: chli1077@colorado.edu

Abstract: Static and dynamic power management have become crucial in real-world applications of embedded systems development. While many technologies and algorithms can reduce power consumption efficiently, like dynamic voltage and frequency scaling, varying voltage and frequency make overall power consumption less efficient. The comprehensive embedded system spends time and energy on the transition between states, voltage, and frequency. To better understand how these problems can be addressed, this paper will explore two methods, the adaptive power management method and the idle state energy reduction method, to mitigate issues on the algorithm side. Furthermore, this paper is intended to give researchers and real-time embedded systems developers an overview of power management methods and to help them better develop the embedded systems.

Keywords: Low energy, Energy efficiency, Real-time embedded system, Power management.

1. Introduction

For over a decade, power management and low energy development have been essential design concerns. Battery-driven real-time embedded systems, like smartphones and wireless headphones, are the primary product that is highly related to these concerns. For better performance of the devices and longer battery life, minimizing the dynamic and static energy are the two primary sources of the rising power consumption problems. The transition activities decide active energy consumption between different voltage and different frequencies. Static energy consumption is determined by the leakage current conducted in an abnormal path under normal system operations. Dynamic energy consumption is the primary source of energy inefficiency for the semiconductor side, and several power management methods only address dynamic energy reduction. As the scaling down in semiconductor technology, static energy consumption is the primary source of overall system consumption [1]. And static energy consumption is considerably comparable with dynamic energy consumption under the deep micro-scale or nanoscale because leakage current increases exponentially.

The structure of this paper is as follows: Section 2 explains the value of energy efficiency in real-time embedded systems. Section 3 discusses the first method, adaptive power management. Section 4 discusses the second method, reducing both dynamic and leakage energy consumption by integrating dynamic voltage and frequency scaling (DVFS) with power mode management. Section 5 analyzes the advantage of both two ways. And Section 6 concludes the paper.

2. Importance of energy efficiency in real-time embedded systems

The types of batteries can be divided into primary batteries (non-rechargeable) and secondary batteries (rechargeable). And the recharging battery is the major power supply used in most applications [2]. Thus, minimizing the power consumption and extending the life cycle of such battery becomes the major limitation in the embedded system design. With high power consumption, the amount of time it can use for other purposes is significantly reduced. And due to the limited power supply size, the efficiency of those embedded systems devices is considerably declined. Battery-driven embedded systems devices are used as smartphones and in the biomedical field, as well as test

measurement equipment. Thus, power management is essential to take place in the development of longer and more efficient embedded system devices.

On the other hand, for the hardware side, high power consumption means high heat dissipation of the device and a higher failure rate of the device. Temperature rises of up to 15 degrees Celsius can result in a factor of two increase in device failure rates [3]. Moreover, high heat dissipation introduces the possibility of component damage and even the functionality of the overall embedded system. Thus, power management plays a vital role in mitigating the chance of device damage and failure.

Due to the challenge faced under the nanoscale, dynamic hardware voltage and frequency scaling [4] seems less efficient because scaling can't control and reduce energy consumption precisely and accurately. To meet the minimum requirements for systems operating correctly and ensuring the optimization of energy consumption, the software algorithm plays an essential role in controlling the system's overall scheduling and reducing energy consumption by minimizing the number and period of idle state. As Niu et al. [5] mentioned, to reduce current leakage, numerous scheduling techniques have been devised by delaying the execution of the incoming task when the processor is in an idle state or by reducing the number of power transitions. And some major approach is shown in Table 1. Moreover, many embedded systems are implemented in a hard real-time system, which is restricted to having all instances of all tasks finish their computation by the relative deadline specified to them as Lipari et al. [6] stated. One primary method to reduce energy consumption is power mode management, which will be discussed in this paper.

Table 1. The primary approach in power management [7].

Method	Hard Real-Time (HRT) Implications	Advantages	Disadvantages
Intra-task DVFS [8]	High risk of deadline miss	Early identification of the slack period	Code mark insertion, multiple frequency/voltage shifts
Inter-task DVFS	Algorithm-dependent, time overhead	Algorithm dependent	Many voltage and frequency variations. Time lag caused by transitions
Static DVFS	Predictable Appropriate for HRT	There aren't many voltage or frequency variations. Predictable and appropriate for HRT	suitable only for tasks that have been fully outlined in advance. limited to a small number of scheduling techniques only
Dynamic DVFS	High, algorithm dependent	Suitable for any scheduling mechanism	Many voltage and frequency variations. Time lag caused by transitions
Aggressive DVFS [9]	Highest	High energy reduction if the prediction functions are accurate	Hard real-time applications are unsuitable
ABB [10]	Small	Reduction of static power	Supplementary hardware
DPM [11]	Smaller than DVFS method	Reduction of total power	High overhead (time and energy) because of switching between states

3. Adaptive Power Management

According to Huang et al. [12], an online algorithm that procrastinates the buffered and future events as far as feasible while taking into account historical and future event arrivals can adaptively forecast the mode transition. The idea behind this algorithm is to track the event arriving in history's

past, which will serve as evidence to predict future event activity. One event with recent appearances will only have a small probability of arrival in the near future. Two central ideas are developed to approach this adaptive algorithm under one event stream: deactivation and activation scheduling decisions.

Deactivation scheduling decisions have two different choices that need to make, depending on the algorithm. The first decision is for the device to change to sleep mode immediately, and another is for the device to stay in standby mode to serve future event arrival. And there is one algorithm, History-Aware Deactivation, presented in the paper. The principle is that the system will turn the device to sleep mode from activation only when energy saving is possible.

Activation scheduling decisions have two different choices that need to make, depending on two different algorithms. The first is for the processor that rapidly switches to active mode to serve upcoming events. And another decision is for the device to stay in sleep mode and aggregate more events to prevent unnecessary state changes. In addition, two algorithms are developed based on this approach. The first one is Worst-case-Greedy Activation (WCG), which activates the device only when the worst case occurs. Otherwise, it keeps the device in sleep mode until a new activation moment is computed. This algorithm decides when the system needs the device to stay in low-power mode. And the second algorithm is Event-Driven-Greedy Activation (EDG), which determines the newest time that the device needs to be in the activated mode to fulfill the time restriction. This algorithm decides when the system requires the device to be in the activated mode instantly. The general workflow is shown in Figure 1.

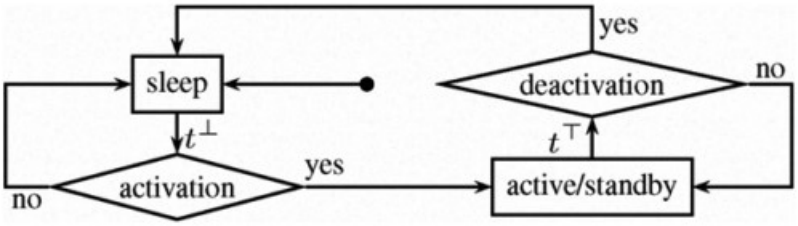

Figure 1. Workflow of Adaptive Power Management method [12].

Under multiple event streams, Huang et al. [12] present solutions for the EDG algorithm using earliest-deadline-first scheduling and fixed priority scheduling. Based on Huang et al. [12] simulation results, the Event-Driven-Greedy algorithm and earliest-deadline-first scheduling are more computationally costly than the Worst-case-Greedy algorithm and fixed priority. In other words, the Event-Driven-Greedy algorithm and earliest-deadline-first scheduling are better methods for real-time embedded systems to reduce static energy consumption in general.

4. Idle Energy Reduction

Niu et al. [5] propose a scheduling technique that is the combination of dynamic voltage scaling and shut down to minimize the dynamic and leakage energy consumption. The concept behind this technique is to reduce energy consumption from two approaches. The first strategy involves balancing the energy that is dynamic and that is static when the processor is turned on and operating at the selected speed. Another strategy is to postpone the forthcoming job until the processor is idle, so the inter-task interval is extended as Niu et al. expected. And the purpose of the second approach is to give the processor enough idle interval time to shutdown/wake up such that the energy overhead wouldn't outweigh the benefit of saving energy with the shutting down technique.

The first approach, DVSLK, is given in a general algorithm that reduces dynamic and leakage current consumption while the deadlines of tasks are guaranteed. For dynamic energy reduction, Niu

et al. [5] assert that energy consumption can be minimized without affecting the functionality of the processor by using a dynamic voltage schedule to determine the threshold speed. But, as the energy problem goes into the leakage energy field, frequent power transitions make the leakage energy consumption significantly considerable. And the approach that Niu et al. [5] propose, the better way to reduce leakage energy consumption is by delaying the execution time of the new upcoming task when the processor is in the idle state until the interval is long enough or the processor will wake up if the new task is going to miss its deadline.

The second approach, DVSLK-O, is given in a more precise computation method that ensures all tasks meet their deadlines according to the dynamic voltage schedule and has linear computation complexity. And the method used to compute the satisfied deadline requirements is the task completion time. So, the task processing wouldn't be stuck in a deadlock loop [13], i.e., the lower priority task successfully receives information and data from the previous higher priority task if the higher priority task is handled correctly. So, the lower priority task can move into the next step. Moreover, Niu et al. [5] mention that the task, which is processed in the low-frequency domain, will also process in the threshold frequency. In other words, such a task with a low-frequency processing speed as the upcoming speed can be delayed if it doesn't fail the deadline requirement. Then, the processor would have more time to stay in the idle interval and minimize the power transition stages.

As the experiment results, Niu et al. [5] compare DVSLK and DVSLK-O with other approaches using the same power mode and the same processor. The result reveals that the idle energy consumption is significantly reduced by 80% compared to the CS-DVSP method. The average energy consumption by different methods is shown in Figure 2.

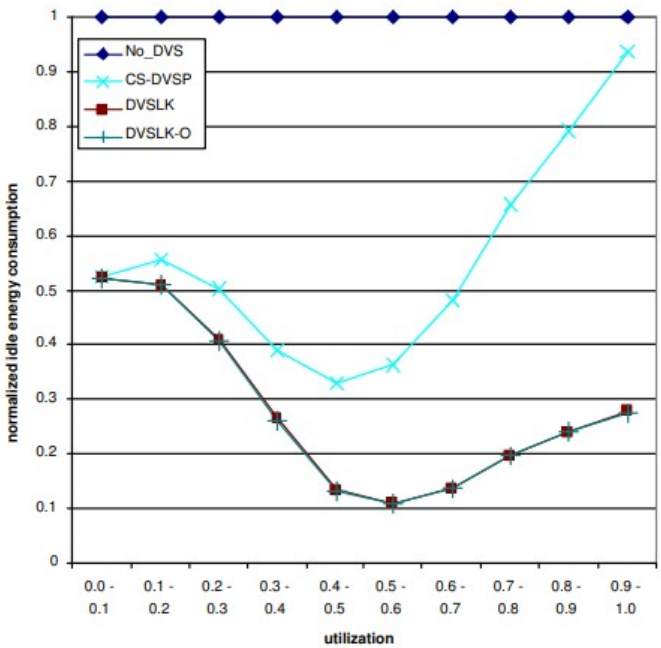

Figure 2. The average energy consumption by different methods [5]

5. Comparison between two methods

These two methods analyzed above share the same concept in general. i.e., staying in the low power mode as long as possible and minimizing the probability of switching power mode. But the approaches used in the two methods are different. So, in this section, this paper will analyze the common and differences between the two methods and the advantages and disadvantages of these two methods.

As stated above, the general concept behind the two methods is power mode management. The first method, adaptive power management, tends to keep the processor to stay in sleep or low-power mode long enough to reduce dynamic and leakage energy consumption by preventing frequent mode transitions. The second method, the idle energy reduction method, also tends to keep the processor in idle or low-power mode long enough to reduce leakage energy consumption. Moreover, these two methods are built under a hard real-time system, which means the deadline requirement is also one of the most critical problems concerning algorithm design. As two methods developed, both algorithms meet deadline requirements precisely, and the development of the algorithm is based on the deadline requirements. Although the general approach of these algorithms is the same, the methods used to minimize the power mode transition are different. The adaptive power management method predicts future tasks' activity based on past tasks' activity to prevent unnecessary power mode transition. The idle energy reduction method minimized the power mode transition by delaying the execution of the upcoming task and extending the idle interval in the idle mode. Furthermore, adaptive power management complicates the algorithm's computation complexity, which needs to consider different situations and evaluate many conditions, then make energy-saving decisions. Unlike the first method, the advanced second method only takes linear computation complexity to make energy-saving decisions. Last, the first method's algorithm is used to make mode-switching decisions. However, the algorithm of the second technique is utilized to postpone the completion of impending tasks and lengthen the idle times when in the idle mode.

The adaptive power management method is most suitable for a sequence of tasks that have a uniform appearance probability. Under such conditions, its algorithm is easier to decide the switching of power mode. Furthermore, this algorithm aims to reduce the leakage energy, which is significantly increased when the technology goes down to the nanoscale. However, the disadvantage of this method is that it does not consider the situation when a sequence of tasks doesn't have a uniform appearance probability. This situation will probably cause the algorithm to predict the wrong future activity, so the wrong actions have been chosen. Such bad decisions will make the tasks miss their deadline requirements and the overall system stuck in a deadlock. In addition, such algorithms aim to change the power mode directly. There is a possibility that the frequent power mode transition is possible, and it increases the probability of higher power consumption overall.

The idle energy reduction method is most suitable for a sequence of tasks that process in a low-frequency domain so that the algorithm can delay the execution time of upcoming tasks and extend idle intervals as long as possible. Additionally, depending on the DVS voltage schedule at the first step, this strategy dials down the dynamic energy use. Because the dynamic and leakage energy are balanced in the first stage, the algorithm may then be particularly optimized for the leakage energy. Lastly, this method takes linear computation complexity to finish and is easier to implement than the first method. An algorithm with linear computation complexity is also a good sign for reducing energy consumption from the software perspective. However, this method is unsuitable for a sequence of tasks that process in a high-frequency domain because those tasks would halt the algorithm from making decisions or bypass much logic to make a wrong decision. And high-frequency domain tasks usually introduce the probability of increasing energy consumption overall. So, the energy reduction goal cannot be maximized under such conditions.

6. Conclusion

This paper explores two methods, adaptive power management and dynamic and leakage energy reduction, used to reduce dynamic and static leakage current energy consumption by switching states and delaying the execution time of upcoming tasks. These two methods are usually referred to as power mode management and are widely adopted in real-time embedded systems development. Overall, this paper is intended to show the importance of power management in embedded systems development and to address the feasible methods and algorithms used to minimize power consumption in real-world applications.

References

[1] Zhang K. Challenges and opportunities in nano-scale VLSI design[C]// IEEE Vlsi-tsa International Symposium on Vlsi Design. IEEE, 2005.

[2] Pedram M, Wu Q . Design considerations for battery-powered electronics[C]// 1999:861-866.

[3] Anderson D, Dykes J, Riedel E. More than an interface—SCSI vs. ATA[C]// File and Storage Technologies. USENIX Association, 2003.

[4] Bradley J, Fang D. Analysis and Implementation of DVFS Techniques in a Multitasking Embedded System.

[5] Niu L, Gang Q. Reducing both dynamic and leakage energy consumption for hard real-time systems[C]// International Conference on Compilers. ACM, 2004.

[6] Lipari G, Palopoli L. Real-Time scheduling: from hard to soft real-time systems[J]. Computer Science, 2015.

[7] Stangaciu C S, Micea M V, Cretu V I . Energy efficiency in real-time systems: A brief overview[C]// Applied Computational Intelligence and Informatics (SACI), 2013 IEEE 8th International Symposium on. IEEE, 2013.

[8] Sasaki H, Ikeda Y, Kondo M, et al. An intra-task dvfs technique based on statistical analysis of hardware events[C]// Conference on Computing Frontiers. 2007.

[9] Li T, Chen Z, Zong Z, et al. A2E: Adaptively aggressive energy efficient DVFS scheduling for data intensive applications[C]// Performance Computing & Communications Conference. IEEE, 2014.

[10] Le Y, Luo J, Jha N. Combined Dynamic Voltage Scaling and Adaptive Body Biasing for Heterogeneous Distributed Real-time Embedded Systems[C]// International Conference on Computer Aided Design. IEEE, 2003.

[11] Benini L, Bogliolo A, Micheli G D. A Survey of Design Techniques for System-Level Dynamic Power Management[J]. Readings in Hardware/Software Co-Design, 2002, 8(3):231-248.

[12] Kai H, Santinelli L, Chen J J, et al. Adaptive Dynamic Power Management for Hard Real-Time Systems[C]// 2009 30th IEEE Real-Time Systems Symposium. IEEE, 2009.

[13] Choi Y, Kwon J, Jeong S , et al. Work-in-Progress: Lightweight Deadlock Detection Technique for Embedded Systems via OS-Level Analysis[C]// International Conference on Embedded Software. 0.