



# THE DYNAMIC VOLTAGE AND FREQUENCY SCALING BASED ON THE ON-CHIP MICROCONTROLLER SYSTEM

<sup>1,2</sup>TIEFENG LI, <sup>1</sup>CAIWEN MA, <sup>1</sup>WENHUA LI

<sup>1</sup> Xi'an Institute of Optics and Precision Mechanics of Chinese Academy of Sciences, Xi'an, 710119, China

<sup>2</sup>The Graduate University of Chinese Academy of Sciences, Beijing, 100049, China

E-mail: [litiefeng@opt.cn](mailto:litiefeng@opt.cn), [macaiwen@opt.ac.cn](mailto:macaiwen@opt.ac.cn), [lwh@opt.ac.cn](mailto:lwh@opt.ac.cn)

## ABSTRACT

With the rapid increase of complexity and size of the on-chip microcontroller system (OCMS), the power consumption issue for the OMCS is increasingly becoming critical and needs to be solved quickly. Because the CPU is often the major power consumer in the OCMS, so an important strategy to achieve energy saving is via the dynamic voltage and frequency scaling (DVFS), which can enable a processor to operate at a range of voltages and frequencies. However, it needs to be emphasized that the conventional DVFS is fully executed by the software scheduler in the operating system, and its main drawback is that the scheduler can't accurately track the performance requirements of CPU when the dormant frequency of CPU is increasing continuously. In this paper, we firstly present a typical hardware DVFS architecture, which can automatically carry out DVFS without the software scheduler involvement. Therefore, it avoids increasing the software's workload and reduces the power consumption.

**Keywords:** *Power Consumption, DVFS, Software Scheduler, Performance Requirement*

## 1. INTRODUCTION

The power consumption is one of the prominent topics of interest to the designers and researchers and remains to be a matter of improvement. Portable electronic devices, which are typically powered by batteries, rely on energy efficient schedules to increase the battery lifetime; while non-portable systems need energy efficient schedules to reduce the operating cost. Early in the design stage when there is a lack of conception of saving power, it is usually ignored by designers because the previous design of the OCMS is considerably simple. However, in the development of microcontroller and the integrated circuit technology, the system is more complicated and consumes more energy, so it is often desirable to simultaneously maximize power duration and minimize power consumption in order to achieve the best OCMS.

Most works cover DVS in the OCMS are aimed at software scheduler method with soft real-time system (Peng et al., 2008; J.O. Coronel et al., 2012; Farshad et al., 2011). Those software DVFS are based on the reclamation of additional slack resulting from the early completions of tasks. These are then used to further reduce the processor frequency and save more energy. These algorithms

are applied at run-time. However, few researchers concern about hardware DVFS in the past ten years.

In this paper, we intend to deal with the power consumption issue of the OCMS with an efficient hardware approach and see how it is different from others design [1-2]. In doing so, we put forward one new DVFS concept that is based on the hardware design. The differences between the novel DVFS and the old DVFS are emphasized in the introduction. Firstly, in the traditional design, software scheduler method in the operating system is still popular in reducing power consumption, which estimates CPU workload according to the frequencies of calling scheduler [3]. However, with the quick development of CPU technology, the dominant frequency of current CPU has achieved to more than 1.5G. For example, the dormant frequency of ARM Cortex A9 is about 1.6G. Obviously, this software approach is not enough in dynamic environments because it can't accurately trace the status of CPU workload in the deadline when the frequency of CPU is higher than before. It is mainly due to the fact that many developers are unable to change hardware architecture of the OCMS at all. Thus, they have to lower power via software compensation in the operating system. By contrast, the new DVFS that is fully presented from hardware side has a remarkable improvement in saving power. Because DVFS is integrated into the

OCMS with three different voltages and frequencies, the various performance requirement of CPU can be rapidly responded by DVFS architecture at the first moment, thus the speed and accuracy of tracking CPU workload are largely improved in essence. The most significant is that all tasks of tracking CPU workload and predicting CPU performance can be finished by hardware automatically, which can transparently lessen the software burden of operating system in the OCMS. Secondly, many new features are firstly used in this paper. For instance, CPU timing check, averaging algorithm, threshold check. Those features will be depicted in detail below.

Generally speaking, the contribution in this paper is that new DVFS aims at novelty in the realization of hardware architecture, instead of following the beaten path of conventionality. In actual application, it is proved that the reliable DVFS can keep the OCMS work with lower power consumption and increase the OCMS working cycle better.

The rest of the paper is organized as follow. Section 2 introduces the conventional DFVS. Section 3 describes the proposed DVFS. Section 4 depicts the performance implementation. Section 5 is experiment result. Section 6 concludes the paper.

## 2. THE CONVENTIONAL DVFS

For the OCMS, its dynamic power formula is stated as

$$P = \alpha CV^2 f \tag{1}$$

where  $\alpha$  represents the percentage of logic cell between 0 and 1 switching, C is a constant that represents the circuit load, V represents the CPU voltage, f represents the CPU frequency.

We can easily know according to the above formula [(1)] that the power of CPU can be lowered by reducing the voltage and frequency.

Traditionally, the researchers and designers adopt software method predicts performance requirement of CPU according to the sequence of event priority in the software scheduler [4]. Fig.1 shows one pure software DVFS method, which is still used according to the priority of all tasks in the operating system. Although it can lower power in a way, it is only efficient on the case that CPU frequency is not high, it will become difficult with the increment of CPU frequency because the software method can't correctly respond to the high frequency of CPU, so it is also failed to estimate the performance

requirement of CPU in time. Moreover, what's even more disconcerting is that the conventional software DVFS need to be called frequently in the operating system. To some degree, it is tedious to change the dynamic voltage and frequency by means of software scheduler. In a summary, the current approaches to DVFS can hardly keep up with the dormant frequency increase of new CPU so that new DVFS technology has to be considered.

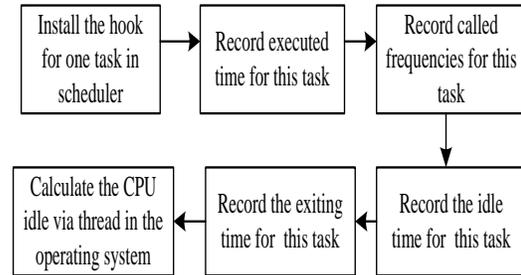


Fig.1 The Conventional Software DVFS

## 3. THE PROPOSED HARDWARE DVFS

To overcome this, the hardware DVFS that has very fast tracking and response speed on CPU behaviour is fully introduced in this paper.

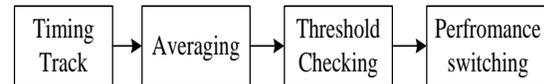


Fig.2 The Hardware DVFS Architecture

Fig.2 represents a block diagram of the proposed implementation of this hardware architecture, which includes timing tracking, averaging, threshold checking and performance switching.

Due to those new features, the hardware DVFS can not only cut down the software overloads that is from operating system but it can also intelligently respond to the external dynamic environment. Once the hardware DVFS function is enabled, the voltage and frequency of the CPU needn't to be adjusted by the software scheduler again and again. In comparison to the previous software DVFS, it firstly strengthens the accuracy of voltage and frequency estimation. Secondly, it lightens the load of CPU timing tracking in a way [5-7].

The calculation of the average idle times is triggered at each monitoring period boundary and for each of the 2 supported scaling steps (down, up). An individual average idle time value is calculated and maintained.

It is also important to note that the Moving Average Algorithm (MAA) is firstly introduced in averaging block. The MAA not only tracks and samples the idle time of the every CPU with small enough intervals but also executes the accumulation and average calculation of the idle times. The MAA formula is given as follows:

$$T_{up}(n+1) = \frac{1}{N} \sum_{k=0}^{N-1} T(n-k) \quad (2)$$

$$T_{down}(n+1) = \frac{1}{M} \sum_{k=0}^{M-1} T(n-k) \quad (3)$$

where  $n$ ,  $M$  and  $N$  are the positive integers and usually  $M > N > 0$ .  $T_{up}(n+1)$  stands for the average of the idle time from the sampling timing 0 to  $N-1$ ,  $T_{down}(n+1)$  stands for the average of the idle time from the sampling timing 0 to  $M-1$ . Based on the Equations (14), (15), the voltage and clock of CPU scaling down step condition is fulfilled if  $T_{down}(n+1) > T1$ . Similarly, the voltage and clock of CPU scaling up step condition is fulfilled if  $T_{up}(n+1) < T2$ . Fig 3 shows the specific automatic transition for the DVFS.

The DVFS has its own timer that can be set to the expected maximum voltage settling time. For example, if a voltage ramping slew rate of 5mV per microsecond is used in changing to the adjacent voltage, it only takes 40 microseconds to stabilize from  $V_{Low}$  (0.8 V) to  $V_{Medium}$  (1.0 V). Whenever the voltage scaling timer elapses, an interrupt can be triggered.

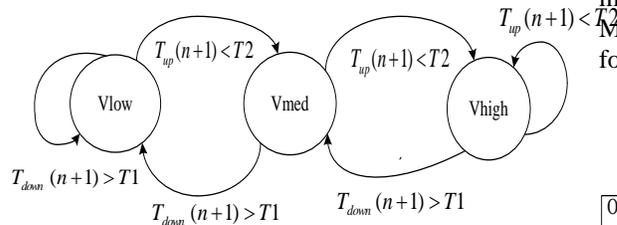


Fig.3 The Intelligent Transition For DVFS

In application, CPU can be configured with the scaling down threshold value (T1) and the scaling up threshold value (T2). It's worth emphasizing that the two threshold values of the CPU idle time have to be set before the DVFS is requested.

#### 4. THE PERFORMANCE IMPLEMENT

As shown in Fig.4, the clock generator includes a Phase-Locked Loop (PLL), a clock dividers and a digital multiplexers. PLL can convert a low-frequency external clock signal that is generated by the on-chip 32.768 kHz oscillator to a high-speed internal clock for maximum. Depending on the different frequency requirement, the clock frequency output may be configured by programming desired P, N and K values according to formula [(4)]. Usually, the on-chip 32.768 kHz oscillator output clock to the PLL. The PLL output can be defined as

$$f_{sys} = f_{osc} * N / (P * K) \quad (4)$$

where N, P and K are pre-defined factors according to the actual requirement. The CPU clock is derived from the oscillator clock ( $f_{osc}$ ), multiplied by N, divided by P, and divided by K. The clock output from the clock dividers can be stated as follows:

$$f_{high} = f_{sys} / X1 \quad (5)$$

$$f_{med} = f_{sys} / X2 \quad (6)$$

$$f_{low} = f_{sys} / X3 \quad (7)$$

where X0, X1, X2, X3 and X4 are the integers and  $X0 < X1 < X2 < X3 < X4$ ,  $f_{high}$  represents the high input clock frequency of CPU,  $f_{med}$  shows the medium input clock frequency of CPU,  $f_{low}$  means the low input clock frequency of CPU. Moreover, we can conclude according to the above formulas [(5)-(7)] that  $f_{high} > f_{med} > f_{low}$ .

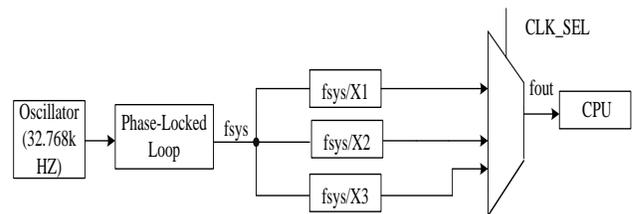


Fig.4 The PLL Clock Generator

In practice, depending on our needs, five voltages and frequencies of CPU may be initialized with below setting.

$$(V_{high}, f_{high}) = (1.20V, 520MHz) \quad (8)$$

$$(V_{med}, f_{med}) = (1.0V, 360MHZ) \tag{9}$$

$$(V_{low}, f_{low}) = (0.90V, 240MHZ). \tag{10}$$

**5. EXPERIMENT RESULT**

Firstly, the accuracy of responding to the CPU performance requirement is evaluated between the conventional software DVFS and the proposed hardware DVFS. In the experiment, the frequency of CPU is continuously increased, the number of performance requirements were sent by the OCMS when the frequency of CPU is equal to f1. The number of efficient response times from DVFS will be recorded at another frequency point f2( f2>f1 and there is enough time interval between f1 and f2 ).According to Fig.5,we know that the speed of software DVFS is unable to keep pace with the faster changes in the frequency of CPU, i.e. the frequencies of CPU are faster, the sensitivity of software DVFS are lower, so it can't respond to the performance requirements of CPU as quickly as possible. Instead, the hardware DVFS can timely react to the dynamic performance requirement of CPU, which deals with nearly 100% of dynamic performance requirements from CPU.

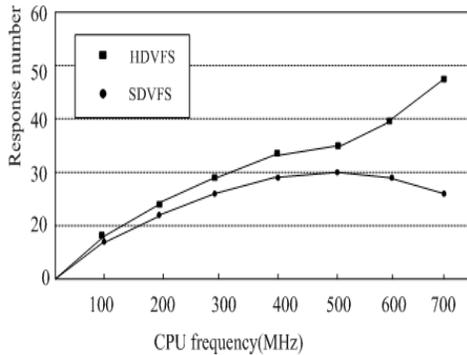


Fig.5 Response Number Comparison

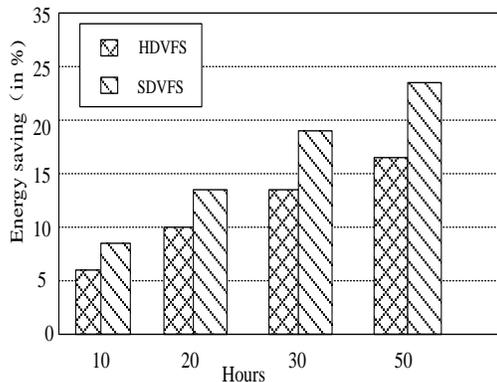


Fig.6 Saving Power With The DVFS

As shown in the above Fig.6, it is obvious that the hardware DVFS has more advantages than the

software DVFS in reducing power consumption[8-14], and the effect of saving energy become evident with the delay of running time. Furthermore, compared with software DVFS, the hardware DVFS are more competitive in saving energy. Finally, we clearly get a conclusion that the hardware DVFS is an efficient and smart way to save energy. In future, the hardware DVFS will be dominant in the OCMS because of the high efficiency.

**6. CONCLUSIONS**

With the intelligence of hardware DVFS, the OCMS can be easily configured with five different voltages and frequencies according to the actual needs. On the other hand, it is worth noting that the new hardware DVFS design has been used to solve the problem on how to reduce the power consumption and save energy. In general, it is obvious that the new DVFS can make the OCMS save the power well, which has been successfully used in the real project. Considerable more work, hopefully, will be done in this area on how to achieve the lowest power consumption in the OCMS by this method provided in this paper.

**ACKNOWLEDGE**

The authors would like to express their gratitude to the associate editor and the anonymous reviewers for their constructive comments.

**REFERENCES:**

- [1] PengManman, LiRenfa,WangYuming, "A Dynamic Voltage Scaling Algorithm Based on Program Section". *Journal of Computer Research and Development*. June 2008.pp.1093-1098
- [2] J.O. Coronel, J.E. Simó. "High performance dynamic voltage/frequency scaling algorithm for real-time dynamic load management ", *Journal of Systems and Software*, Volume 85, Issue 4, April 2012, Pages 906-919
- [3] Farshad Firouzi, Mostafa E. Salehi, Fan Wang, Sied Mehdi Fakhraie. "An accurate model for soft error rate estimation considering dynamic voltage and frequency scaling effects", *Microelectronics Reliability*, Volume 51, Issue 2, February 2011, Pages 460-467
- [4] Min Yeol Lim, Vincent W. Freeh, David K. Lowenthal. "Adaptive, transparent CPU scaling algorithms leveraging inter-node MPI communication regions", *Parallel Computing*,



- Volume 37, Issues 10–11, October–November 2011, Pages 667-683.
- [5] Minming Li. “Approximation algorithms for variable voltage processors: Min energy, Max throughput and online heuristics “,*Theoretical Computer Science*, Volume 412, Issue 32, 22 July 2011, Pages 4074-4080.
- [6] Nikzad Babaii Rizvandi, Javid Taheri, Albert Y. Zomaya. “Some observations on optimal frequency selection in DVFS-based energy consumption minimization”, *Journal of Parallel and Distributed Computing*, Volume 71, Issue 8, August 2011, Pages 1154-1164
- [7] B. Indu Rani, C.K. Aravind, G. Saravana Ilango, C. Nagamani. “A three phase PLL with a dynamic feed forward frequency estimator for synchronization of grid connected converters under wide frequency variations ”,*International Journal of Electrical Power & Energy Systems*, Volume 41, Issue 1, October 2012, Pages 63-70
- [8] Chuberre, N., et al. “Satellite digital multimedia broadcasting for 3G and beyond 3G systems”, *13th IST mobile & wireless communication summit 2004*, Lyon, France.
- [9] M. Etinski, J. Corbalan, J. Labarta, M. Valero, “Optimizing job performance under a given power constraint in HPC centers”, *International Conference on Green Computing*, vol. 0, 2010, pp. 257 - 267.
- [10] X. Fan, W.-D. Weber, L.A. Barroso, “Power provisioning for a ware housesized computer, in: *ISCA’07: Proceedings of the 34th Annual International Symposium on Computer Architecture*, ACM, New York, NY, USA, 2007, pp. 13 - 23.
- [11] N. Kappiah, V.W. Freeh, D.K. Lowenthal, Just in time dynamic voltage scaling: exploiting inter-node slack to save energy in MPI programs”, in: *SC*, vol. 0, 2005, pp. 33.
- [12] E. Le Sueur, G. Heiser, “Dynamic voltage and frequency scaling: the laws of diminishing returns”, *Proceedings of the 2010 Workshop on Power Aware Computing and Systems*, HotPower’ 10, Vancouver, Canada, October 2010.
- [13] M.Y. Lim, V.W. Freeh, “Determining the minimum energy consumption using dynamic voltage and frequency scaling”, *Parallel and Distributed Processing Symposium*, international, vol. 0, 2007, pp. 348.
- [14] D Love, R Heath, V Lau, D Gesbert, B Rao, M Andrews, “An overview of limited feedback in wireless communication systems”, *IEEE J Sel Areas Communication*. 2008 vol.26, no 8, pp.1341–1365.