Sixth Annual IEEE International Conference on Pervasive Computing and Communications

A Power Management mechanism for Handheld Systems having a Multimedia Accelerator

Junho Ahn[1], Jung-hi Min[1], Hojung Cha[1] and Rhan Ha[2]

1 Department of Computer Science, Yonsei University, Seoul Korea
{junho, jhmin, hjcha}@cs.yonsei.ac.kr
2 Department of Computer Engineering, Hongik University, Seoul, Korea
rhanha@cs.hongik.ac.kr

Abstract

Recently the demand on high graphic ability has increased in handheld systems due to multimedia or game application. Conventional DVS (dynamic voltage scaling) methods focus on reducing energy consumption of CPU. They didn't consider the influence of multimedia accelerators to make DVS policies. From the preliminary experiment, we found that the conventional DVS algorithm is not optimal to a system with a multimedia accelerator in context of reducing energy consumption of the system. We propose a power management mechanism that guarantees both QoS and reduction of energy consumption by considering the relation between the frequencies of the CPU and the frequencies of the multimedia accelerator. The proposed mechanism is a DVS technique in context of multimedia accelerator. We experimented the proposed mechanism on a testbed equipped with Intel XScale processor and Intel 2700G multimedia accelerator. The experimental result shows that the proposed method reduces energy consumption of the system as much as by 33% compared to conventional CPU-only DVS algorithms.

1. INTRODUCTION

Recently, mobile systems with built-in multimedia accelerators have increased to enhance graphic ability of the system. A multimedia accelerator enables a system to reduce its CPU usage during multimedia application are executed and to guarantee QoS of application. When a system has a multimedia accelerator, the energy consumption of accelerator needs to be considered as well as that of the CPU to reduce energy consumption of system. Many dynamic voltage and frequency scaling (DVFS) algorithms which select the frequency and voltage to reduce the energy consumption have been proposed [1-11]. Conventional DVFS methods execute a multimedia application with maximum frequency and then control the CPU frequency according to the difference of decoding and slack time. Low voltage and frequencies of the CPU lead low performance of the system. In this case, the frequency-tunable multimedia accelerator can provide high performance even with low frequencies for the CPU.

In this paper, we propose a power management technique that considers both the CPU and the multimedia accelerator simultaneously. We did preliminary experiments to find the opportunity of reducing more energy consumption in the system equipped with a multimedia accelerator. The proposed algorithm makes combination sets with the CPU and multimedia accelerator frequencies and then sorts them by processing performance. The proposed power management system selects frequencies set according to the effect of saving power consumption and guarantee of QoS in predefined set of the voltage and frequency of the CPU and the frequency of the multimedia accelerator. The selected frequency set by the proposed mechanism guarantees QoS of multimedia applications and reduces energy consumption of system in the context of both the CPU and a multimedia accelerator. The proposed method is evaluated using a testbed equipped with Xscale PXA270 [12] and an Intel 2700G multimedia accelerator [13, 14]. The processor used in this experiment supports DVFS and the Intel 2700G5 multimedia accelerator supports dynamic frequency scaling (DFS) [14].

The rest of the paper is organized as follows: Section 2 describes motivation of this work. We describe the integrated power management algorithm in Section 3, and validate the mechanism in Section 4. We conclude the paper in Section 5.

0-7695-3113-X/08 $25.00 © 2008 IEEE
DOI 10.1109/PERCOM.2008.82

2. MOTIVATION

In a system which doesn't have a multimedia accelerator, only the CPU executes multimedia applications. In this case, the performance of processing multimedia applications increases as the voltage and frequency of the CPU increases. Figure 1 shows the relation of performance and the CPU power consumption. Conventional DVS algorithms [1-3] select the voltage and frequency of the CPU to satisfy both the QoS of the multimedia application and energy saving of the CPU.

We did preliminary experiments to examine whether there is additional opportunity to reduce energy consumption in a system with multimedia accelerator. Figure 2 shows the architecture of the system used in the preliminary experiments. The clock frequencies used for the CPU (with corresponding voltages) were 104MH, 208MHz, 312MHz, 416MHz, 520MHz, and 624MHz.

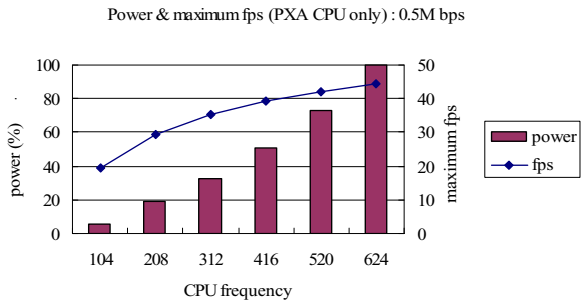

Figure 1: Power consumption vs. CPU frequency during processing multimedia data by Mplayer

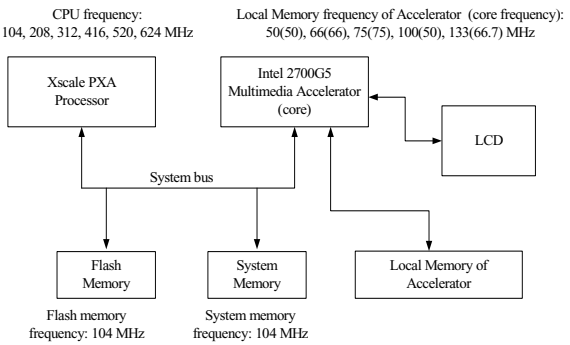

Figure 2: The architecture of the system used for proposed mechanism

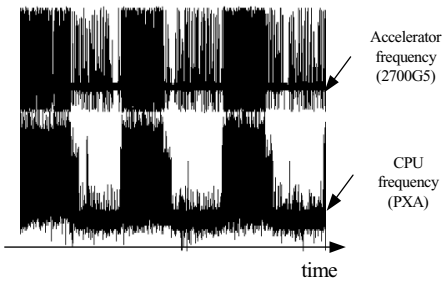

Figure 3: The frequency trace of the CPU and the accelerator for a multimedia application

The frequencies used for the multimedia accelerator were 50MHz, 66MHz, 75MHz, 100MHz, and 133MHz. The CPU and multimedia accelerator are not independent of each other because CPU sends multimedia data to the local memory and then accelerator executes the data. To identify the relation between the CPU and the multimedia accelerator, we measured the frequency of two components of the system of which CPU and a multimedia accelerator are connected by the system bus.

Figure 3 shows the trace of frequencies of the CPU and the multimedia accelerator during execution of a multimedia application. When the CPU sends a frame of multimedia data to a multimedia accelerator from memory, the frame data is stored in the buffer of accelerator. After sending one frame, the CPU turns to lower power mode, and the multimedia accelerator starts decoding and sending for displaying. The CPU and multimedia accelerator operate together when multimedia data is executed. However, the processing performance of the CPU is not proportional to that of the multimedia accelerator. From Figure 3, we can see that there are several periods that frequencies of the CPU do not much change while the frequencies of the multimedia accelerator increase a lot. This means that the conventional DVS methods might not be optimal methods to reduce energy consumption of the system which has a multimedia accelerator. Figure 4 shows the maximum FPS (frame per second) as a function of the pair of frequencies. From Figure 4, we can see that the processing performance does not have a linear relation between the voltage and frequency of the CPU and the multimedia accelerator. Let define a pair of CPU and multimedia accelerator frequencies as the frequency of the CPU / the frequency of the multimedia accelerator. The system shows higher processing performance at 416 / 75MHz than system executes at 520 / 50MHz, which consumes more energy than 416 / 75MHz.

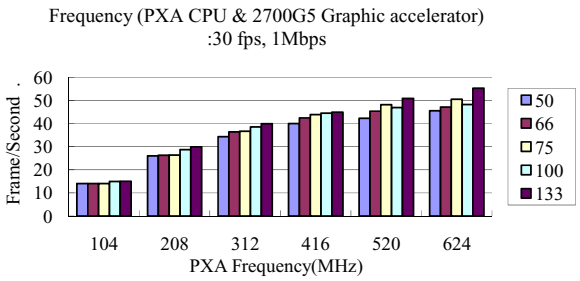

Figure 4: Maximum processing speed vs. frequency of the CPU and the multimedia accelerator (1Mbps)

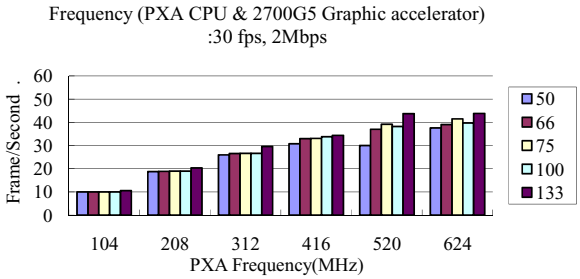

Figure 5. Maximum processing speed vs. frequency of the CPU and the multimedia accelerator (2Mpbs)

This experimental result implies that the influence of frequencies of the incoming data is trivial. Figure 5 show the maximum FPS as a function of frequencies on 2Mbps. It shows the similar patterns as Figure 4.

We sorted pairs of the voltage and frequency of the CPU and frequency of multimedia accelerator by total energy consumption. As we can see from Figure 4 and 5, the frequencies selected by the conventional DVS mechanism satisfy the QoS of the multimedia application, but there is opportunity to reduce more energy if considered dependency between the CPU and multimedia accelerator. We describe the proposed algorithm which can reduce more energy than conventional DVS algorithms in the system with multimedia accelerator in the next section.

3. INTEGRATED POWER MANAGEMENT ALGORITHM

In this section, we describe the proposed power management algorithm that considers both the CPU and the multimedia accelerator. The power management algorithm needs to choose appropriate frequencies for the CPU and the multi-media accelerator. In this algorithm, we assume that the system has all pairs of possible frequencies of the CPU and the multimedia accelerator as a database. An execution speed in the multimedia accelerator is determined by the frequencies of both CPU and accelerator. To get proper pair of the frequencies of the CPU and multimedia accelerator, the proposed system estimates whether the current pair is the best one by searching with following the pairs sorted by processing performance.

Figure 6 shows the flowchart of the proposed power management algorithm. In the initialization step, frequency pairs of the CPU and multimedia accelerator are sorted based on their maximum speed. And then the highest frequency pair of the CPU and multimedia accelerator is set by the proposed system. At the first state, the handheld system is executed with given maximum frequency pair having maximum speed. And then the incoming multimedia data are decoded. If QoS is guaranteed after decoding, the proposed system evaluates whether the determined frequency pair is optimal pair in context of both the CPU and the multimedia accelerator or not. If it is not optimal pair, it selects a lower frequency pair of CPU and accelerator than the current frequency pair. If it is optimal pair, the proposed system checks whether processing of all frames is finished. Otherwise, it selects next frame and goes back to the decoding state. In case that QoS is not satisfied, the proposed mechanism selects the frequency pair satisfying the lower power consumption and higher processing speed that means next level of pair than current frequency pair. This process goes routinely until all frames are processed. The proposed algorithm also calculates the slack time of each frequency pair and controls the frequencies of the CPU and the multimedia accelerator to enhance energy saving with satisfying QoS. Low frequency pair is not the proper pair because it consumes least power but does not guarantee QoS. If selected frequency pair does not guaranteed QoS, this algorithm returns the frequency pair to the previously frequency pair which QoS is satisfied with. The last selected frequency pair means the frequency pair can reduce power consumption for the system with processed application. For example, when the system tries to maintain 40fps of processing speed, the lower bound of the frequency pair, the frequency of the CPU / the frequency of the multimedia accelerator, which satisfies QoS is 520 / 50Mhz. However the optimal frequency pair that guarantees the QoS and consumes least power is 416 / 66MHz. To find the optimal frequency pair, the algorithm finds

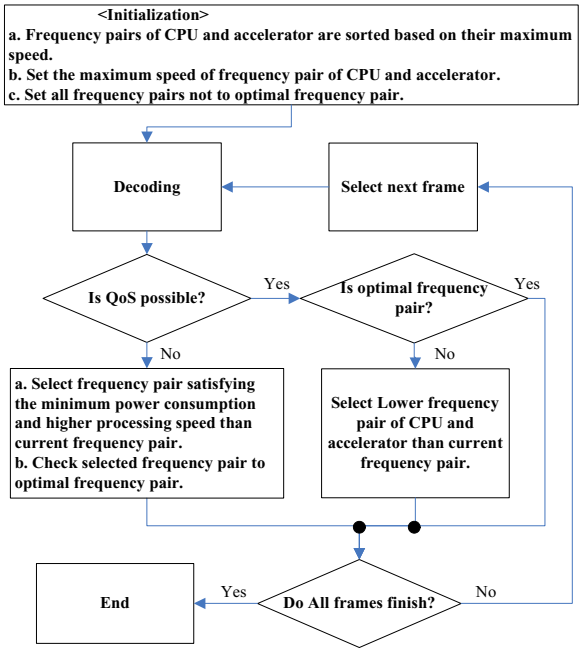

Figure 6. The integrated power management algorithm

the pair with minimum power consumption in context of both the CPU and the multimedia accelerator among the pairs that satisfy the QoS.

Figure 7 shows the variation of frequencies during the algorithm is being operated. The power consumption order is as follows: 520 / 100MHz is higher than 520 / 50MHz. 520 / 50MHz is higher than 416 / 100MHz. Process performance order is as follows: 520 / 100MHz is higher than 416 / 100MHz. 416 / 100MHz is higher than 520 / 50MHz. The proposed algorithm decreases the frequencies by process performance order within guarantee of QoS. When the system realizes that 520 / 50MHz is the minimum performance level that guarantees QoS, the algorithm finds the frequency pair that consumes minimal power among the pairs that have been estimated. After that, the system would maintain the optimal frequencies of the CPU and the multimedia accelerator to reduce power consumption as shown in Figure 8.

4. EXPERIMENTS

We implemented the proposed power management algorithm with the Linux 2.6.5 kernel. We used an XScale PXA processor and an Intel 2700G5 multimedia accelerator on an X-Hyper270B board. The CPU supports six different speeds: 104, 208, 312, 416, 520 and 624 MHz.

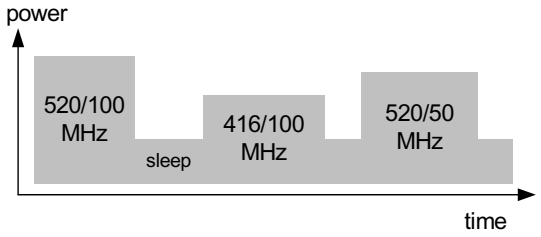

Figure 7. Frequency variation during optimization

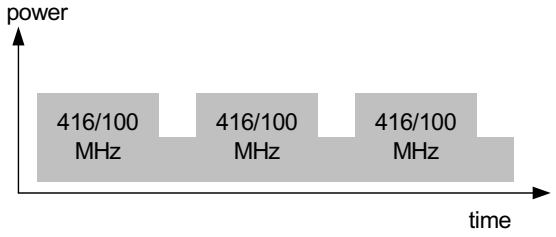

Figure 8. Frequency after optimization

The local memory of the accelerator supports five different speeds: 50(core frequency 50), 66(66), 75(75), 100(50) and 133(66.7) MHz. The speeds of both can be adjusted dynamically with operating system control. The CPU supports DVS, and the multimedia accelerator supports DFS. We used multimedia files with 1Mbps, 2Mbps, 3Mbps and 4Mbps. The FPS

(frame per second) is 30, and the frame size is 640x480. The Mpeg decoder was used for the experiment. We used the Agilent DSO6034A oscilloscope as a measuring device. Figure 9 shows experimental environment.

For the performance evaluation of the proposed algorithm, we experimented three different cases. First, we examined a case in which no power management algorithm was applied to the system. In this case, the CPU always runs at the highest speed. We also measured the power consumption of a system using the CPU-DVS algorithm. And then, we estimated the proposed mechanism. The experimented data were obtained using 1Mbps, 2Mbps, 3Mbps and 4Mbps Mpeg files. Figure 10 shows the energy consumption of each technique when we executed a multimedia file of 1Mbps at 30 fps. Our algorithm saves 33% and 7.5% of energy than the system with no management and CPU-DVS algorithm, respectively. The CPU-DVS algorithm chooses 416 / 66MHz (CPU / accelerator frequency) as the optimal frequency pair. The next level from 416 / 66MHz is 208 / 133MHz because the levels are sorted by the amount of power consumed. Because 208 / 133MHz does not guarantee QoS, the CPU-DVS chooses 416 / 66MHz as the optimal pair even though other frequency pairs consume less power and guarantee the QoS. On the other hand, our algorithm sorts the frequency pairs by process performance and finds the pair that consumes the least power among the pairs that guarantee the QoS. As a result, our algorithm chooses 312 / 50MHz, which consumes less power than 416 / 66MHz, as the optimal frequency pair.

Figure 11 shows a trace for the CPU when the system executed a multimedia file of 1Mbps. We compared the energy consumption of the CPU-DVS and the proposed technique. The frequency pair 416 / 66MHz used more than 312 / 50MHz, but the decoding time was shorter. The voltage of 416 / 66MHz is higher than 312 / 50MHz. According to the energy consumption, 416 / 66MHz consumes more power than 312 / 50MHz; hence the proposed technique chooses the frequency pair 312 / 50MHz to save more energy while the 416 / 66MHz frequency pair selected by the CPU-DVS. Figures 12, 13 and 14 show the results when the system executes multimedia files of 2Mbps, 3Mbps and 4Mbps with no-DVS, CPU-DVS and the proposed method. The proposed technique reduced energy consumption by 18.1% with a 2Mbps, 22.5% with a 3Mbps, and a 3.4% with 4Mbps multimedia file compared with No-DVS algorithm and 5.1%, 7.6% and 0.8% compared with the CPU-DVS, respectively.

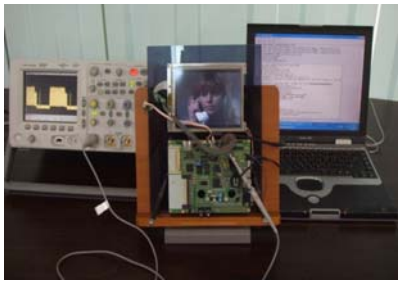

Figure 9. Experimental setup

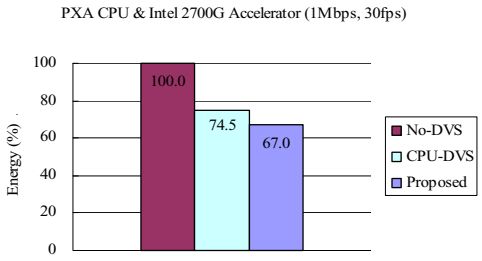

Figure 9. Energy comparison: 1Mbps, 30 fps

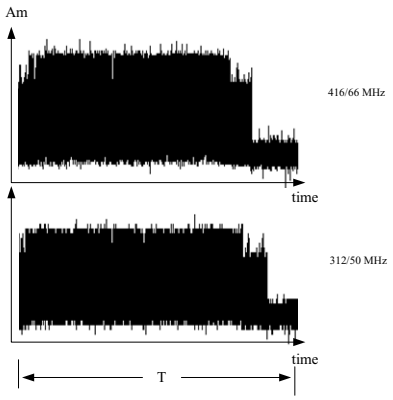

Figure 11. The current traces for CPUs at two different frequencies with a 1Mbps multimedia file

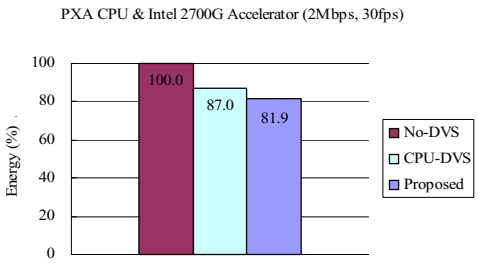

Figure 12. Energy comparison: 2Mbps, 30 fps

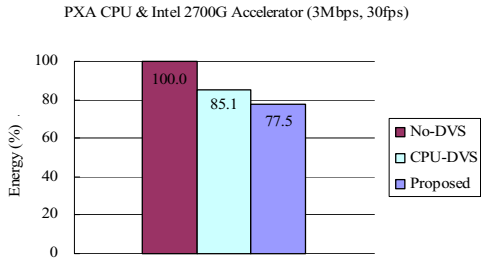

Figure 13, Energy comparison: 3Mbps, 30 fps

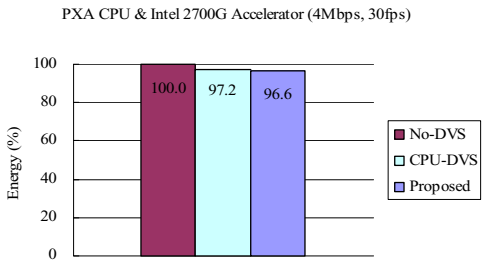

Figure 14. Energy comparison: 4Mbps, 30 fps

5. Conclusion

In this paper, we proposed an effective power management algorithm in a handheld system that includes a multimedia accelerator. The algorithm is based on a periodic process monitoring mechanism. Instead of reducing only the CPU frequency, the proposed method sorts frequency levels by processing time (fps) and chooses an appropriate level that guarantees QoS as well as reducing energy consumption. We implemented the proposed method on XScale-based testbed that includes a multimedia accelerator. Experimental result shows that the propsed method can reduce energy consumption of system as much as by 33% compared to a system with no energy-saving policy and by 8% compared to DVS methods.

Acknowledgements

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the National Research Lab. Program funded by the Ministry of Science and Technology (No. M10500000059-06J0000-05910) and the Seoul R&BD Program of City of Seoul, Korea.

6. References

[1] K. Choi, K. Dantu, W. Cheng and M. Pedram, "Frame-based dynamic voltage scaling for a MPEG decoder," in Proc. of ICCAD'02 -A give the CM/IEEE Int'l Conf. on Computer Aided Design, 2002, pp. 732–737.

[2] M. Mesarina and Y. Turner, “Reduced energy decoding of MPEG streams,” in Proc. of Multimedia Computing and Networking, San Jose, CA 2002.

[3] Y Tan, P Malani, Q Qiu and Q Wu “Workload Prediction and Dynamic Voltage Scaling for MPEG Decoding,” in Proc. of 11th Asia and South Pacific Design Automation Conference, 2006, pp. 911–916.

[4] H. Aydin, R. Melhem, D. Mosse, and P. Alvarez. “Dynamic and aggressive scheduling techniques for power-aware real-time systems.” In Proc. of 22nd IEEE Real-Time Systems Symposium, Dec. 2001.

[5] P. Pillai and K. G. Shin. “Real-time dynamic voltage scaling for low-power embedded operating systems.“ in Proc. of 18th Symposium on Operating Systems Principles, Oct. 2001.

[6] Wanghong Yuan and Klara Nahrstedt, “Practical Voltage Scaling for Mobile Multimedia Devices,” in Proc. of ACM Multimedia 2004, New York, NY, October, 2004.

[7] K. Flautner and T. Mudge. “Vertigo: Automatic performance-setting for Linux.” In Proc. of 5th Symposium on Operating Systems Design and Implementation, Dec. 2002.

[8] D. Grunwald, P. Levis, K. Farkas, C. Morrey III, and M. Neufeld. “Policies for dynamic clock scheduling.” In Proc. of 4th Symposium on Operating System Design and Implemen-tation, Oct. 2000.

[9] J. Lorch and A. Smith. “Improving dynamic voltage scaling algorithms with PACE.” In Proc. of ACM SIGMETRICS 2001 Conference, June 2001.

[10] W. Yuan and K. Nahrstedt. “Energy-efficient soft real-time CPU scheduling for mobile multimedia systems.” In Proc. Of 19th Symposium on Operating Systems Principles, Oct. 2003.

[11] W. Yuan, K. Nahrstedt, S. Adve, D. Jones, and R. Kravets. “Design and evaluation of a cross-layer adaptation framework for mobile multimedia systems.” In Proc. of Multimedia Computing and Networking Conf., Jan. 2003.

[12] Intel Corporation, "Intel PXA270 Processor Electrical, Mechanical, and thermal Specification datasheet." Intel technical report, http://www.intel.com, February 2005.

[13] Intel Corporation, "Intel 2700G Multimedia Accelerator DataSheet." Intel technical report, http://www.intel.com, April 2004.

[14] Intel Corporation, " Mobility Maximized: Intel Redefines What a Wireless Handheld Can Do.” Intel technical report, http://www.intel.com, May 2004.