Frequency-Voltage Cooperative Power Reduction for Multi-tasking Multimedia Applications

MISAKA Satoshi, TOYAMA Keisuke, ARITSUKA Toshiyuki, UCHIYAMA Kunio, AISAKA Kazuo, KAWAGUCHI Hiroshi, SAKURAI Takayasu

Central Research Laboratory, Hitachi, Ltd.
Center for Collaborative Research, University of Tokyo

Keyword: low power, frequency-voltage cooperative power control, multi-tasking, application processor, real-time OS

1. Introduction

Mobile applications, especially for cellular phones, need to satisfy the requirement for low power consumption keeping multimedia performance high enough. Thus software and hardware combined approach for low power becomes really important. One of such techniques is “frequency-voltage cooperative control (FV control)”[1] that lowers the CPU clock and core voltage at the same time when maximum execution speed is not necessary. FV control drastically decreases the power consumption without sacrificing real-time feature because frequency and voltage contribute to its reduction by the order of three. This technique is suitable for moving picture processing which is one of main applications for 3G mobile phone, because its workload is various during execution. FV control can be applied to several levels, such as a whole application, a task, or a fragment of a program (program slice). We mainly develop a program slice level FV control because program structure can be used to control power more precisely[2].

In FV control based on program slices, where CPU changes its speed during program execution, timing constraints must be satisfied when such programs are running as multi-tasks. Also, low power modes equipped to the CPU should be utilized in cooperation with FV control. We implemented an efficient FV control under these restrictions.

2. The target system

As a target system for this FV control, we use a system board with an application processor for mobile phones[3]. Newly developed voltage change circuit supplies two voltages according to the 0/1 signal from the processor’s GPIO. Clock frequency is changed by setting the frequency division ratio of the PLL output. The voltage change circuit is added between the CPU’s V_{CC} pin and the voltage regulator. We implemented 2 system calls of SetFV_Low (1/2 frequency) and SetFV_High (maximum frequency) on a real-time OS. The total system architecture is shown in Fig.1.

3. FV control in MPEG-4 video middleware

We ran MPEG-4 video decoder in this system. Programs are divided into slices and a scheduled processing time is estimated by linear programming. And the deadline for activation of the next picture frame is assigned. A checkpoint is attached to each slice, where a margin or the difference of the actual executed time and the scheduled processing time is calculated. Frequency and voltage are lowered when this margin is enough for WCET (Worst Case Execution Time) of the slice. For MPEG-4, 100 slices are created according to macro block of 16x16 pixels processing. Only several instructions are used for the calculation and judgement and their overhead is negligible.

We evaluated effect of FV control using typical moving picture decoding of QCIF sized, 10 frames/sec and 30% average workload. The result is 70% reduction for power consumption at 2 sets of frequency/voltage – 120MHz/1.5V and 60MHz/1.25V without loss of decoding quality. The prototype voltage change circuit requires 3msec for settling the output voltage. This overhead will be reduced around 100μsec, bringing power reduction to be 75%.

In most cases the execution is done before the deadline and we have spare time because the controlled frequency is maximum and a half of it. We set the CPU to sleep mode (low power consumption mode by cutting off the clock) in this spare time to get more power reduction in cooperation with FV control.

We designed a power management task in Fig.1 with lowest execution priority, which sets CPU to sleep mode. The power management task is invoked when MPEG-4 video decoding task finishes its decoding. MPEG-4 task is re-activated when

the deadline reaches. Fig.2 shows voltage and current transition at FV change. Average current is 153mA at 1.5V, 76mA at 1.25V, and 24mA at sleep mode respectively.

4. FV control for synchronized multi-tasks

In an actual situation of cellular phone applications, several running programs require real-time feature. Thus it is essential that FV control should work well in that situation. We implemented and evaluated our FV control with multi-tasking where MPEG-4 video decoder task and MP3 decoder task are running synchronously.

4.1 Application of FV control to MP3

MPEG-4 program is divided into 100 program slices by the macro block processing. MP3 program is divided into 4 slices according to the granule for audio sample data and stereo channels. Calculation and judgement schemes at the checkpoints are the same as the ones in MPEG-4.

Workload for MP3 does not vary compared to MPEG-4, and the ratio of its maximum and minimum is typically 3:2. Since the average workload is 21%, cooperative work of FV control and low-power mode saves the power consumption by 80%.

4.2 Multi-tasking of MPEG-4 and MP3

Actual application where MPEG-4 and MP3 are synchronously running under the FV control is shown in Fig. 3. In the FV control under multi-tasking situation, each task observes its deadline and is scheduled to activate the next frame by the deadline. When there is a time margin for the deadline, F and V are lowered.

The frame duration for MPEG-4 and MP3 are set to 100msec (10 frames/sec) and 26msec (44.1kHz sampling) respectively. MP3 task has higher priority to MPEG-4 task.

The initial values of the deadlines for the tasks are 100msec and 26msec respectively. As the deadlines are updated with the current time when the margin is estimated, the margin will be precise even an interruption from other tasks occurred. For its higher priority, MP3 task preempts MPEG-4 task and is activated in order to execute in its time frame.

The MP3 task with higher priority performs FV control when it is activated from a sleep mode. Otherwise, when it preempts other tasks, FV control for the MP3 task does not take place in that frame. This management of FV control is done by the power manager task. An execution scheme for the two tasks is shown in Fig. 3.

Deadline time and power manager task make the FV control valid under an MPEG-4 and MP3 multi-tasking environment and makes 60% power reduction.

5. References

[1] S. Lee and T. Sakurai, “Run-time Voltage Hopping for Low- power Real-time Systems”, DAC, pp.806-809, June 2000.

[2] K. Aisaka, et al. ”Design Rule for Frequency-Voltage Cooperative Power Control and Its Application to an MPEG-4 Decoder”, Proc. Symp. on VLSI Circuits, pp.216-217, June 2002.

[3] T. Yamada et al. “A 133MHz 170mW 10μA Standby Application Processor for 3G Cellular Phones”, ISSCC Digest of Technical Papers, pp.370-371, Feb. 2002.

Responsible author name : MISAKA Satoshi
Postal address : 1-280, Higashi-koigakubo Kokubunji-shi, Tokyo 185-8601, Japan
E-mail address : smisakas@crl.hitachi.co.jp
Phone number : +81-42-323-1111 (ext.3753)
Fax number : +81-42-327-7737

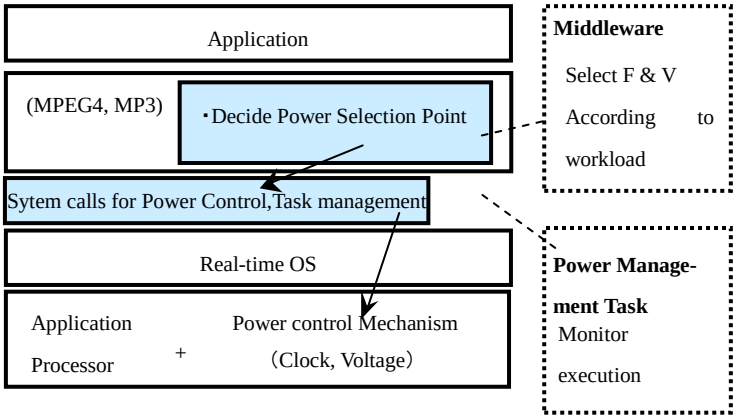

Fig.1 Power Control System Architecture

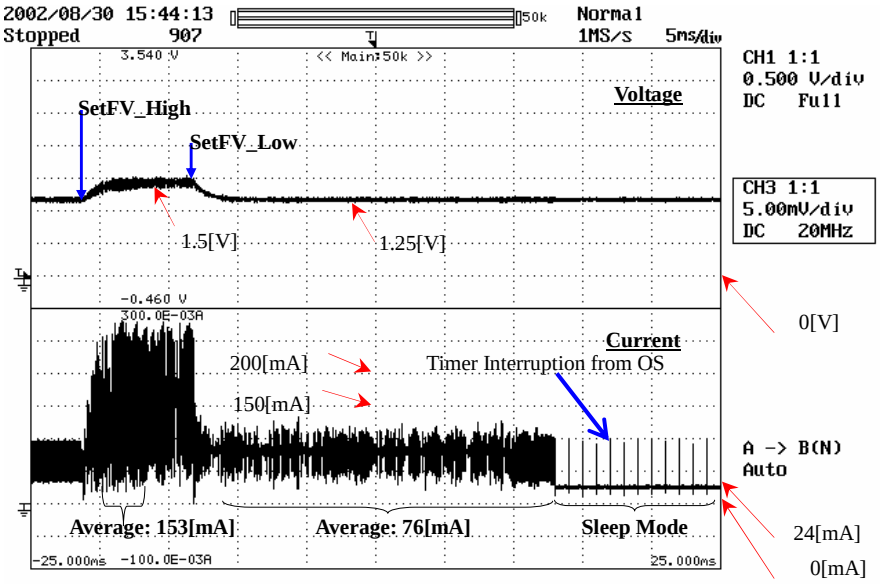

Fig.2 Measured waveforms for FV control

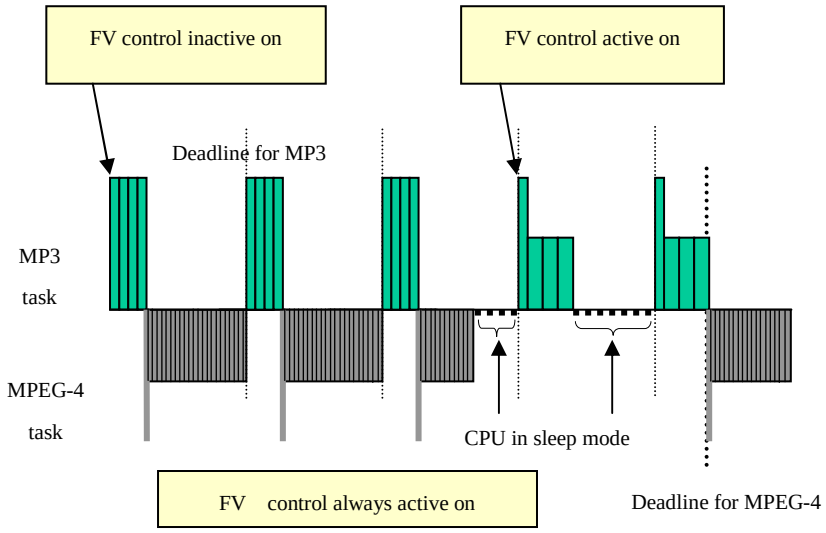

Fig.3 Scheduling example for MPEG-4 and MP3