

Run-time Power Control Scheme Using Software Feedback Loop for Low-Power Real-time Applications

Seongsoo Lee

Takayasu Sakurai

Center for Collaborative Research and Institute of Industrial Science, University of Tokyo

7-22-1 Roppongi, Minato-ku, Tokyo, Japan, 106-8558

Tel: +81-3-3402-6226

Fax: +81-3-3402-6227

e-mail: {cupid, tsakurai}@iis.u-tokyo.ac.jp

Abstract – A novel hardware-software cooperative power control scheme, namely a software feedback loop, is proposed to lower power consumption of VLSI systems. The proposed run-time voltage and frequency control scheme guarantees the real-time execution of applications. It avoids interface problems and also provides “binary-code compatibility”. Using a software analysis environment, the power control scheme is shown to achieve more than 90% power reduction for real-time MPEG-4 SP@L1 video encoding, taking into consideration the transition delay between voltage and frequency levels.

I. INTRODUCTION

Over the past several years, reduction of power consumption is taking on significant importance in VLSI system design, especially for portable, battery-powered devices such as a digital cellular phone and a personal digital assistant (PDA).

One promising approach in power reduction is dynamic voltage scaling [1][2]. In the scheme, supply voltage is reduced to the lowest possible level to achieve the lowest power consumption, since the power consumption of CMOS circuits is proportional to the square of the supply voltage. As the supply voltage decreases, the speed of CMOS circuits also decreases. Therefore, the supply voltage should be dynamically controlled based on the workload variation.

Recently, extensive studies have been carried out on the dynamic voltage scaling [1]-[5], based on compile-time supply voltage scheduling. These approaches, however, are not suitable for real-time applications, because they do not guarantee for the program to be finished in a given time interval. Neither, it does provide the optimum supply voltage,

because the workload variation of each task is not exactly known at compile-time and it is data dependent.

Moreover, in the previously published approaches, system clock frequency can have arbitrary values, which may cause interface problems to exchange data with external memories and peripheral LSI's. Especially, this interface problem becomes serious for external devices such as a cathode-ray tube (CRT), a liquid-crystal display (LCD), and a radio frequency (RF) front-end. Usually, these devices operate at a constant clock frequency so that complicated interface circuits may be needed.

A dynamic voltage scaling scheme often employs a DC-DC converter and a frequency synthesizer to control the supply voltage and clock frequency. DC-DC converters have rather slow transition delay compared with processor cycle time. The transition delay should be also taken into account, but most of the conventional approaches do not.

Voltage-frequency relationship is important in dynamic control of the supply voltage. In the conventional approaches, the relationship is stored in the form of dedicated hardware and/or customized software. The voltage-frequency relationship, however, depends strongly on the process technology and architecture, which means that hardware redesign or software recompilation is required when the generation of a processor changes.

In this paper, a novel hardware-software cooperative scheme is proposed to lower power consumption of VLSI systems. It consists of (1) a simple power control chip with an on-chip DC-DC converter and a frequency generator, (2) a simple run-time power control algorithm using software feedback loop, and (3) a device driver for considering

voltage-frequency relationship of a target processor.

The dynamic software feedback scheme proposed in the paper guarantees real-time execution of the application for the first time and achieves more power saving over compile-time supply voltage scheduling. It is because the supply voltage is controlled adaptively in run-time and the scheme takes the transition delay into account. The power control chip generates f_{CLK} , $f_{CLK}/2$, $f_{CLK}/3$, ... to avoid interface problems, where f_{CLK} is the master clock frequency. Voltage-frequency relationship is given in the form of device driver program being separate from the application program. Therefore, neither recompilation nor reprogramming of the application program is needed even when a processor is upgraded.

Since the power control chip is separate from the processor, proposed scheme can be directly applied to many existing commercial processor systems.

In order to show the effectiveness of the scheme, performance evaluation is conducted using Pentium II code set and MPEG-4 video encoding [6], which is going to be the killer application for portable systems that require real-time execution.

II. ARCHITECTURE

The outline of the proposed run-time power control scheme is shown in Fig. 1. The power control algorithm is embedded in a small part of the application program, and is executed on target processor. It administrates the power control chip by sending control codes through I/O ports of a target processor.

The power control chip has simple hardware architecture. It does not need to be redesigned for specific target processor,

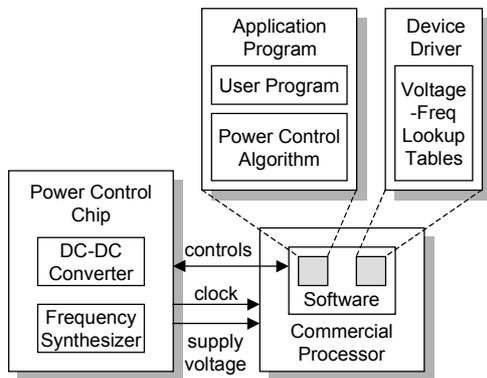


Fig. 1. Proposed run-time power control scheme.

because all processor-dependent parameters are stored in a separate device driver program. By using the device driver that can be modified easily when the version of the target processor is changed, the proposed system guarantees the “binary-code compatibility” for any versions of the upward-compatible processor. On the contrary, in the conventional approaches, the application program should be reprogrammed or recompiled. Moreover, in these cases, “binary-code compatibility” is not guaranteed.

Fig. 2 shows the supply voltages and clock frequencies for the target system. The power control chip generates the variable supply voltage V_{VAR} and the variable clock frequency f_{VAR} from the master supply voltage V_{DD} and master clock frequency f_{CLK} . Since voltage-frequency relationship of each chip is different, variable supply voltage V_{VAR} is applied only to core logic of the target processor that consumes most of the power.

To avoid interface problems, all chips on the target system operate at the same clock frequency f_{VAR} , except for I/O chips. Moreover, f_{VAR} has only discrete values of f_{CLK} , $f_{CLK}/2$, $f_{CLK}/3$, ... to simplify the interface scheme. The I/O chips should operate at a constant clock frequency f_{CLK} to exchange data with external devices such as CRT, LCD, and RF front-end.

Most real-time applications have some given time intervals in which a certain amount of task should be executed. For example, real-time MPEG-4 SP@L1 video coding should process a picture in 1/15 second. For the rest of this paper, we refer this time interval as a sync frame.

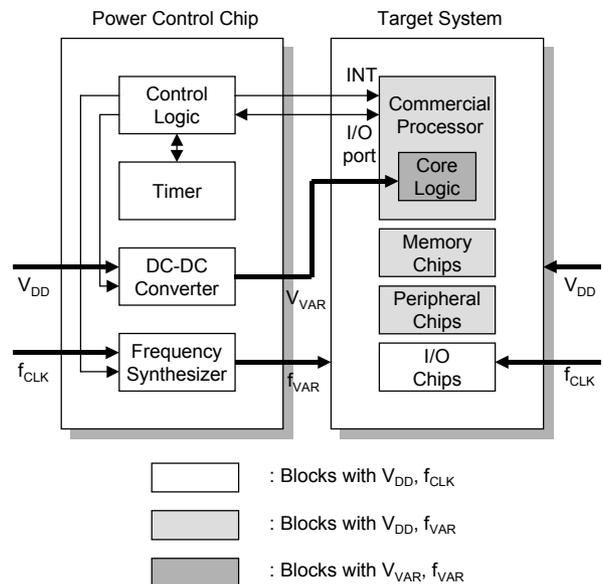


Fig. 2. Supply voltages and clock frequencies for the target system.

In the proposed power control scheme, every sync frame is divided into several timeslots. For each timeslot, target execution time T_{TAR} is calculated, and f_{VAR} is determined to finish the task of a given timeslot within T_{TAR} to guarantee the real-time execution. V_{VAR} is determined from f_{VAR} , based on the voltage-frequency relationship of the target processor.

The proposed power control algorithm for real-time application is shown in Fig. 3. Application program has two loops: one for the sync frame and the other for the timeslot. The device driver program has two lookup tables: one for voltage-frequency relationship of target processor and the other for transition delay of the power control chip. These lookup tables are made after the measurement of the target processor and the power control chip.

III. SOFTWARE ALGORITHM AND HARDWARE OPERATION

A. $f_{VAR_V_{VAR_calculate()}$

Fig. 4 describes the algorithm how to determine V_{VAR} and f_{VAR} for each timeslot. The important point is that even for the worst case, all the tasks assigned in a sync frame should be done within that sync frame to achieve the real-time execution. The following algorithm guarantees this point.

- (1) For the i -th timeslot, Current time T_{Ci} is taken from the internal timer in the power control chip. Target time T_{TARi} is calculated as $T_{TARi} = \sum T_{Si} - T_{Ci}$, where T_{Si} is execution time limit of the timeslot at maximum clock frequency.
- (2) Estimated worst-case execution time T_{ik} is calculated as $T_{ik} = T_{TD} + T_{Si} \times (f_{CLK}/f_k)$ if a clock frequency is chosen

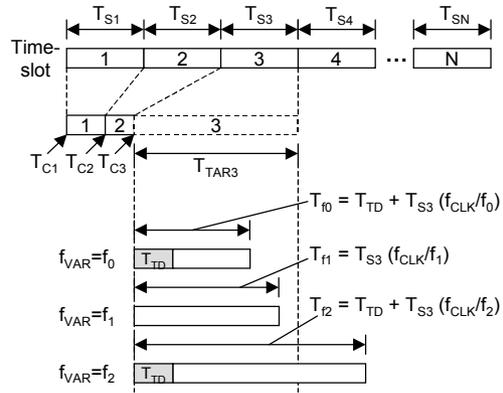
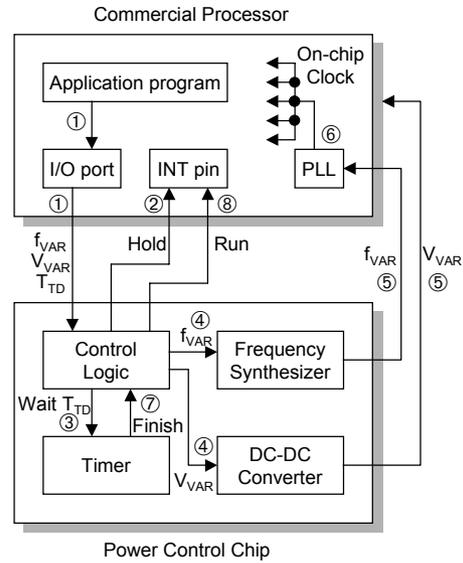


Fig. 4. Determination of V_{VAR} and f_{VAR} .



(a) Block diagram.

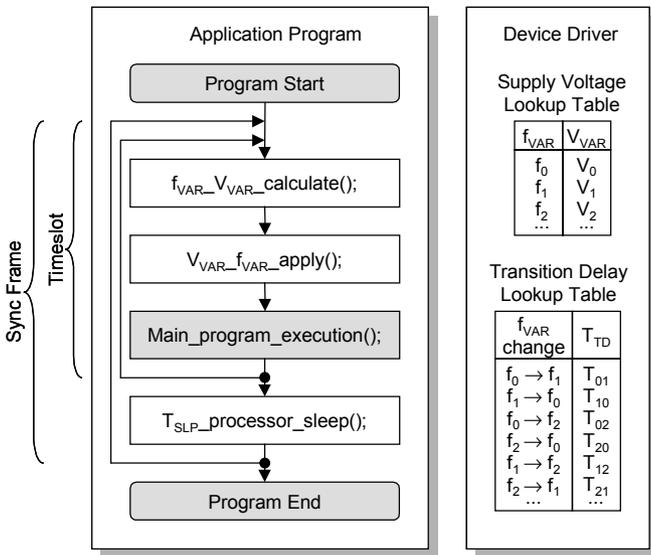
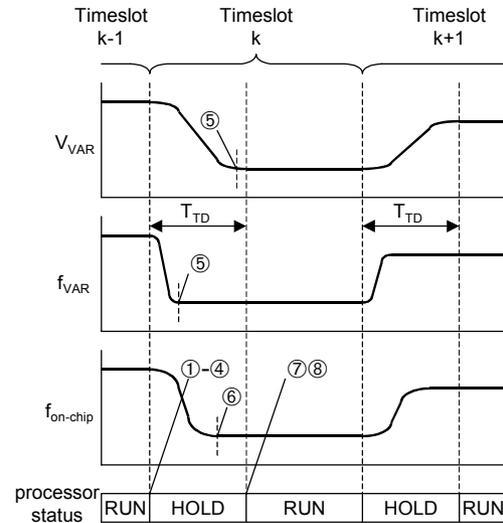


Fig. 3. Application program and embedded power control algorithm.

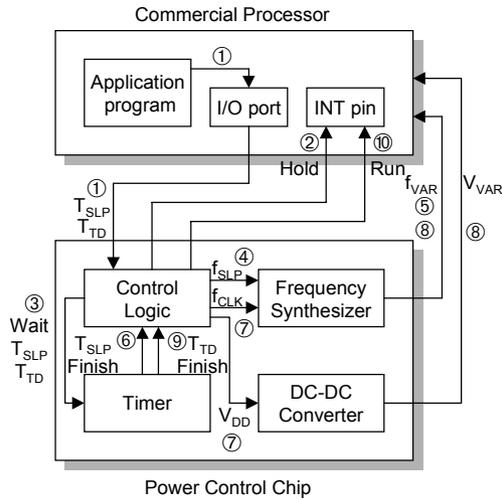


(b) Timing diagram.

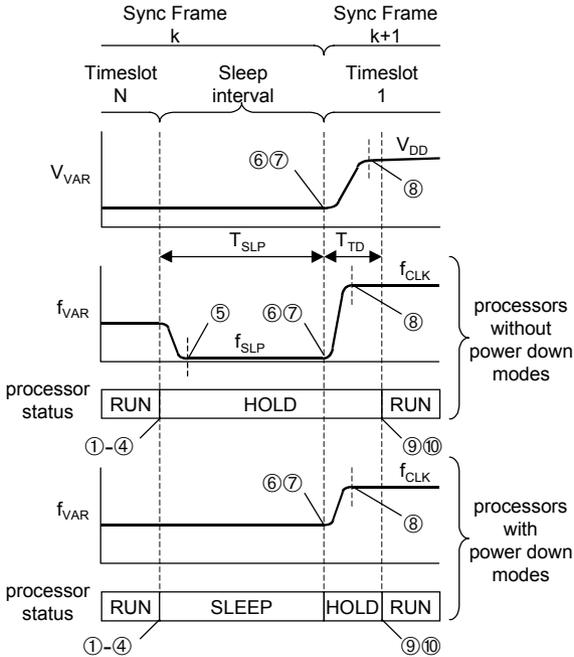
Fig. 5. Applying f_{VAR} and V_{VAR} to the target processor.

as f_k ($k=1,2,3\dots$). Note that there is no transition delay if the clock frequency f_k does not change from a timeslot to the next timeslot. In Fig. 4, $T_{fl} = T_{S3} \times (f_{CLK}/f_i)$ because $f_{VAR} = f_i$ for the previous timeslot.

- (3) Clock frequency f_{VAR} is determined as the minimum clock frequency f_k whose estimated worst-case execution time T_{ik} does not exceed the target time T_{TARI} . Note that real execution time is always smaller than the target time, which guarantees the real-time execution.
- (4) Supply voltage V_{VAR} is determined from f_{VAR} , using the lookup table in the device driver.



(a) Block diagram.



(b) Timing diagram.

Fig. 6. Making the target processor idle at the end of a sync frame.

B. $f_{VAR} V_{VAR} apply()$

After f_{VAR} and V_{VAR} are determined in the application program, these values are transferred into the power control chip using I/O instructions as shown in Fig. 5.

During the transition delay T_{TD} , the clock frequency and the supply voltage are unstable, which may cause invalid operation of the target processor. To avoid this problem, the power control chip makes the target processor in a HOLD state, where the target processor stops running in this interval. The power control chip waits for T_{TD} using an internal counter and then issues the interrupt for the target processor to wake up. Without this internal timer in the power control chip, there is no way for the processor to restart.

Many commercial processors have an internal phase-lock loop (PLL) for clock generation. In this case, settling time of an internal PLL should be taken into account in measuring T_{TD} .

C. $T_{SLP} processor_sleep()$

In real-time applications, there is some idle time at the end of each sync frame, because the real execution time of each sync frame is always smaller than the worst-case execution time. Even in this case, V_{VAR} cannot be reduced to zero lest all internal data in the processor should be lost and the interrupt signal should be neglected. The clock frequency, however, can be zero, which results in no power consumption during the idle time. For some processors in a dynamic design style that has the minimum operating frequency, the f_{VAR} should be set equal to the minimum frequency f_{SLP} that the power control chip can provide.

Recently, many commercial processors provide a SLEEP mode for minimum power consumption, where most part of the processor stops running and internal clock goes zero. In this case, we don't need to control the clock frequency. The power control chip makes the target processor in a SLEEP state, instead of a HOLD state. Fig. 6 shows how to make the target processor idle at the end of sync frame.

IV. PERFORMANCE EVALUATION

In order to evaluate the proposed power control scheme, we applied it to MPEG-4 video encoding, which is one of the typical real-time portable low-power applications for mobile equipments.

We created real-time MPEG-4 SP@L1 video encoding

software running on Intel Pentium II [7] processor, and measured the execution time of every timeslot and every function module. Effect of operating system (OS) was measured and eliminated, using Intel VTune Performance Analyzer [8] program. Execution cycles of each timeslot were calculated from this timing information.

Power consumption was calculated using Eq. (1), and voltage-frequency relationship was obtained from Eq. (2), based on the alpha-power delay model [9].

$$P = \sum P_{SL}, \quad P_{SL} \propto V_{SL}^2 N_{SL} \quad (1)$$

$$\frac{1}{f_{SL}} \propto \frac{V_{SL}}{(V_{SL} - V_T)^\alpha}, \quad (2)$$

where P is total power consumption, and P_{SL} , V_{SL} , N_{SL} , and f_{SL} signify power consumption, supply voltage, execution cycles, and clock frequency of a given timeslot, respectively, V_T is the threshold voltage of the processor, and α is the velocity saturation index.

Simulation conditions are summarized in Table 1. In the modeling of the voltage-frequency relationship, V_{DD} , V_T , and α are assumed to be 2.5V, 0.5V, and 1.3, respectively, but the normalized power consumption is not sensitive to these values if the parameters are chosen in a practical range.

Fig. 7 shows the power consumption using the proposed power control scheme. This power consumption is normalized by P_{FIX} , which is the power consumption with a fixed supply voltage and a fixed clock frequency.

The theoretical limit of the power consumption is also calculated using a post-simulation analysis. This theoretical

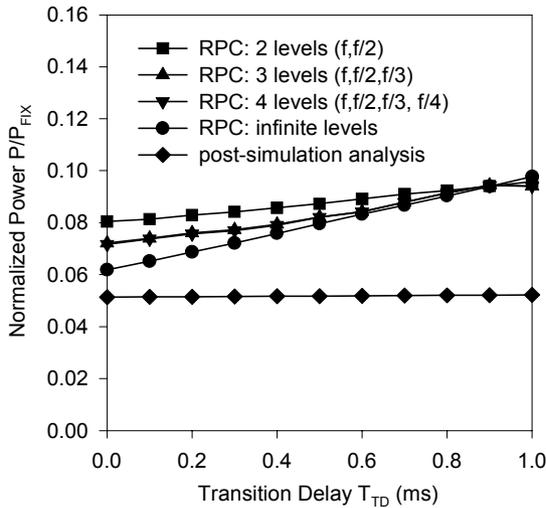
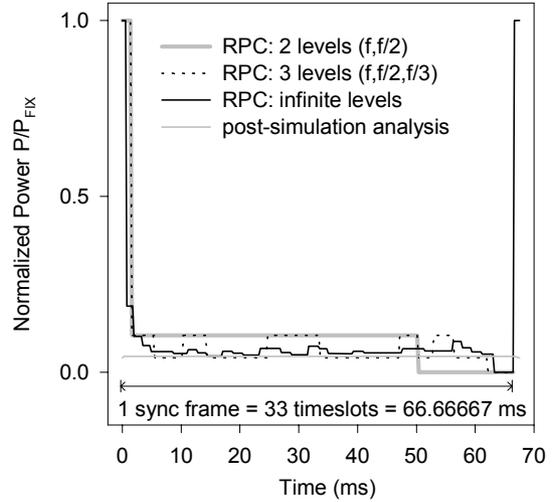
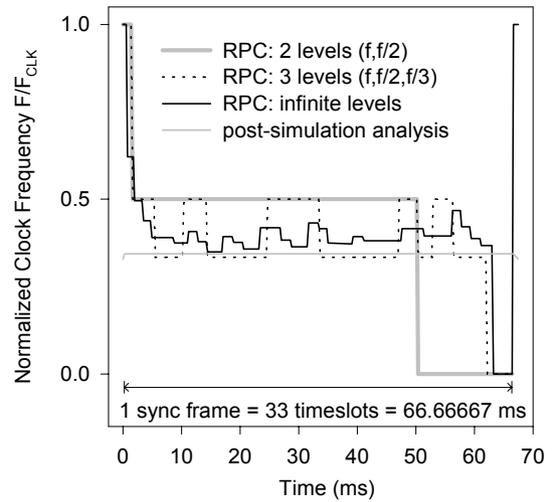


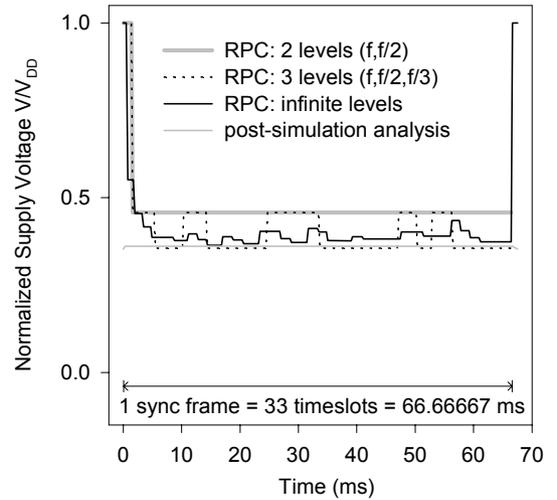
Fig. 7. Normalized power consumption.



(a) Power.



(b) Clock frequency.



(c) Supply voltage.

Fig. 8. Power, clock frequency, and supply voltage of the 194th sync frame

limit is optimal in energy saving but can not be realized, because V_{VAR} and f_{VAR} are determined knowing exact number of cycles required for the sync frame. This can be only achieved by a two-pass process, which is unfeasible in real environments.

From Fig. 7, it is seen that the proposed power control scheme has about 90~94% power reduction, while the theoretical limit has 95% power reduction. Also, only two ($= f, f/2$) or three ($= f, f/2, f/3$) discrete levels of the clock frequency are sufficient. Power efficiency decreases as the transition delay of the power control chip increases, because the target processor stops its execution during the transition delay.

Fig. 8 shows the normalized transient curves of power, f_{VAR} , and V_{VAR} of the 194th sync frame when the transition delay $T_{TD} = 0.5$ milliseconds.

TABLE I
SIMULATION CONDITIONS

Target application	
Real-time algorithm	MPEG-4 SP@L1 video encoding
Input picture format	QCIF (176 pels \times 144 pels), 15 Hz
Number of macroblocks in a picture	99
Distance between I-pictures	5 (IPPPP IPPPP IPPPP...)
Method of motion estimation	Full search
Search ranges of motion estimation	$\pm 7 \times \pm 7$
Voltage-delay modeling	
External supply voltage	2.5V
Threshold voltage	0.5V
α	1.3
Power control algorithm	
Length of sync frame	66.67 ms
Length of timeslot	2.020 ms
Number of timeslots in a sync frame	33

V. CONCLUSION

A novel run-time power control scheme is proposed using a software feedback loop suited for real-time applications. It employs a power control chip with an on-chip DC-DC converter and a frequency synthesizer, and an embedded run-time power control algorithm using the software feedback loop.

The proposed power control scheme guarantees the real-time operation, and optimizes the supply voltage in run-time while the conventional approaches do in a compile time. It can be directly applicable to many existing processors without hardware redesign, and also has the “binary-code

compatibility” for generations of a processor series. It avoids interface problems with external memories, peripheral chips and external devices, by exploiting discrete clock frequency $f_{CLK}, f_{CLK}/2, f_{CLK}/3, \dots$

When applied to real-time MPEG-4 SP@L1 video encoding, the proposed power control scheme is shown to achieve more than 90% power reduction compared with the fixed frequency and voltage scheme, while guaranteeing the real-time operation. Currently, hardware implementation of the proposed power control scheme is in progress.

ACKNOWLEDGEMENTS

This research was partly supported by Mirai-Kaitaku Project.

REFERENCES

- [1] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos, “Data driven signal processing: an approach for energy efficient computing,” *Proceedings of International Symposium on Low Power Electronics and Design (ISLPED '96)*, pp. 347-352, 1996.
- [2] A. Chandrakasan and R. Brodersen, *Low Power Digital CMOS Design*, Kluwer Academic Publishers, 1995.
- [3] T. Ishihara and H. Yasuura, “Power-Pro: programmable power management architecture,” *Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC '98)*, pp. 321-322, 1998.
- [4] T. Ishihara and H. Yasuura, “Voltage scheduling problem for dynamically variable voltage processors,” *Proceedings of International Symposium on Low Power Electronics and Design (ISLPED '98)*, pp. 197-202, 1998.
- [5] T. Pering, T. Burd, and R. Brodersen, “The simulation and evaluation of dynamic voltage scaling algorithms,” *Proceedings of International Symposium on Low Power Electronics and Design (ISLPED '98)*, pp. 76-81, 1998.
- [6] ISO/IEC JTC1/SC29/WG11 N2202, “Coding of audio-visual objects,” May 1998.
- [7] <http://www.intel.com/PentiumII>
- [8] <http://www.intel.com/vtune/analyzer>
- [9] T. Sakurai and A. Newton, “Alpha-power law MOSFET model and its application to CMOS inverter delay and other formulas,” *IEEE Journal of Solid State Circuits*, vol. 25, no. 2, pp. 584-594, Apr. 1990.