Energy-Aware Clock-Frequency Assignment in Microprocessors and Memory Devices for Dynamic Voltage Scaling

Youngjin Cho, *Student Member, IEEE*, and Naehyuck Chang, *Senior Member, IEEE*

***Abstract*—Dynamic supply-voltage scaling (DVS) can reduce the energy consumption of microprocessors, but most DVS schemes only scale the clock frequency of the microprocessor and ignore the memory system. In this paper, we show how more energy can be saved by changing the clock frequency of the memory as well as that of the microprocessor in a coordinated fashion. The contributions of this paper include: 1) consideration of both the energy and access time of the memory; 2) derivation of a mathematical formulation of a system-wide energy model as a function of the clock frequencies of the microprocessor and memory; 3) derivation of analytic solutions of system-wide energy-optimal clock-frequency pairs for the microprocessor and the memory, and, finally, 4) extension of the frequency-assignment technique to handle discrete voltages and frequencies. Cycle-accurate system-level energy simulation shows that the proposed scheme can save up to 50% more energy than previous DVS schemes. Our approach can also be applied to other synchronous peripheral devices.**

***Index Terms*—Energy management, power management, real-time systems.**

I. Introduction

DYNAMIC supply-voltage scaling (DVS) is one of the most efficient ways to reduce the dynamic energy consumption of supply-voltage-clock-scalable (SVCS) devices [2], [3]. However, many devices include current-mode circuits that do not allow supply-voltage scaling. Even if a device does allow supply-voltage scaling, there is no energy saving unless there is a superlinear relationship between the supply voltage and the power consumption. Consequently, DVS is mainly restricted to the core of SVCS microprocessors. However, the power consumption of a microprocessor represents around 10% to 25% of the total power consumption in handheld devices, as shown in Fig. 1. Furthermore, the power consumption of the memory is as important as that of the microprocessor [4]. Since most DVS schemes proposed so far are applicable to only the core of SVCS microprocessors, it is necessary to consider the power consumption of the entire system including memory and other devices.

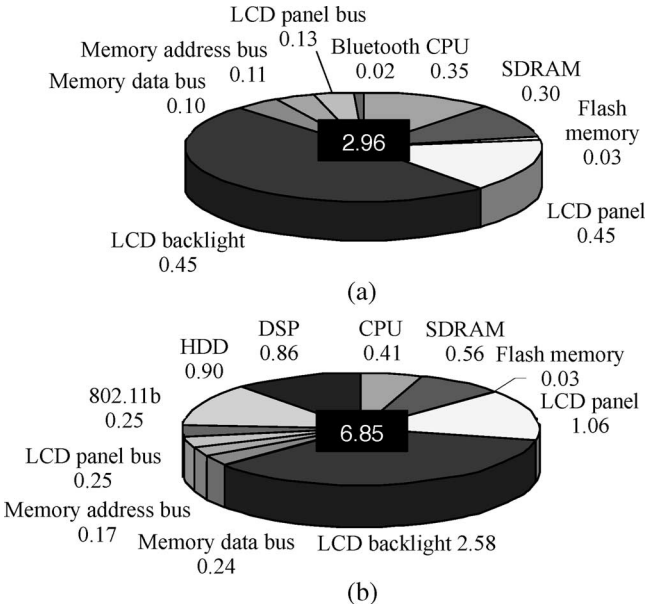


Fig. 1. Power consumption of three representative handheld devices running a streaming video application (W) [4]. (a) PDA power consumption. (b) Portable media player power consumption.

The effects of nonsupply-voltage-scalable (NSVS) memory technology on DVS have been considered by some recent works [5]–[7]. The decreased clock frequency of the microprocessor for lower supply voltage increases the execution time of a task, which may actually increase their energy consumption. This shows how existing DVS techniques can easily fail to achieve system-wide energy optimization.

It has also been shown that down-scaling the clock frequency of a microprocessor cannot reduce the total system energy when the memory bandwidth is limited by the application [5], [6]. A recent approach to DVS [7] therefore takes the memory access time into account when the microprocessor frequency is determined. Applying dynamic power management (DPM) to NSVS devices can mitigate the energy overhead of the extended execution time that results from DVS. DPM and DVS may be combined to achieve system-wide optimization in a predictive power control strategy [8]. When DVS is applied to an SVCS microprocessor, NSVS memory devices controlled by that microprocessor will have longer idle times, which can magnify the effect of DPM [9]. While most existing DVS scheduling

Manuscript received October 31, 2005; revised April 21, 2006 and August 2, 2006. This work was supported in part by the Brain Korea 21 Project, LG Yonam Foundation, and IT R&D Project funded by Korean Ministry of Information and Communications. This paper was presented in part at the International Symposium on Low-Power Electronic Design 2004 (ISLPED 04). This paper was recommended by Associate Editor M. Pedram.

The authors are with the School of Computer Science and Engineering, Seoul National University, Seoul, 151-744 Korea (e-mail: yjcho@cselab.snu.ac.kr; naehyuck@snu.ac.kr).

Digital Object Identifier 10.1109/TCAD.2006.885835

algorithms ignore the memory and peripheral devices, some recent works on task scheduling [10]–[12] have tried to minimize the energy consumption of the total system.

It has become very clear that a microprocessor and its memory affect each other in terms of both energy and delay time, and that independent consideration of either device cannot be expected to minimize the energy consumption of the whole system. However, none of the work on this topic so far accurately describes the interactions between a microprocessor and its memory system under DVS in terms of both energy and performance. The combination of DPM and DVS [8] has been analyzed in terms of total energy consumption but not at the level of the individual device. Energy models [9] have included too simple assumptions, in which constant power consumption is consumed in both active and power-down modes.

In this paper, we aim at the limited but well-defined goal of a system-wide energy-optimal frequency assignment for a system, which consists of an SVCS microprocessor and a synchronous NSVS memory. We assume that neither the CPU nor the memory enters power-down mode while a task is being executed. For this purpose, we derive the relationship between energy consumption and clock frequencies for an SVCS microprocessor and a synchronous NSVS memory based on their static and dynamic energy consumption. We prove that there exists an energy-optimal pair of clock frequencies for an SVCS microprocessor and a synchronous NSVS memory, and that these frequencies are functions of the number of microprocessor clock cycles, the number of memory transactions, and the parameterized hardware energy model. We derive a generalized analytical solution that determines the optimal frequencies under various constraints. We also demonstrate the practical use of frequency assignment on a commercially available SDRAM running real applications. Experimental results show 50% greater energy saving than traditional DVS schemes, which ignore the energy consumption of the memory.

II. Principle of Memory-Aware Frequency Assignment

A. Problem Statement

Up to the present, there has been no proper strategy for reducing the energy consumption of a system consisting of a synchronous NSVS memory controlled by an SVCS microprocessor. Most DVS schemes for an SVCS microprocessor simply ignore the memory and assume that the execution time and energy consumption of the system are solely determined by the microprocessor. One recent scheme [7] does consider the proportion of the execution time taken by memory accesses and uses this to determine a lower clock frequency and, hence, a lower supply voltage for the microprocessor. Some studies do take into account the energy consumption of the memory, but even these cannot estimate the energy accurately because of their simple energy models. Without distinction between dynamic energy and static energy, it is impossible to estimate the energy consumption of the memory that varies as the clock frequency of the microprocessor changes. But, none of the existing schemes has considered it.

In reality, the memory does contribute both to the execution time and to the energy consumption. Even if we do not scale the supply voltage of synchronous NSVS memory, its energy consumption will change with clock frequency due to changes in leakage energy during the active mode and due to changes in dynamic energy during the idle mode (details of energy consumption of a synchronous NSVS memory will be discussed in Section II-B). Thus, when we assign a clock frequency $f_{\rm c}$ to the microprocessor, we must also consider the clock frequency $f_{\rm m}$ of the memory. Since both $f_{\rm c}$ and $f_{\rm m}$ affect the execution time, energy-optimal frequency pairs $(f_{\rm c}, f_{\rm m})$ must be derived together.

Our proposed clock-frequency assignment determines a pair of frequencies $(f_{\rm c}, f_{\rm m})$ that minimizes the energy consumption of the entire system for a given application and hardware configuration. The supply voltage of the synchronous NSVS memory is fixed, and the supply voltage of the SVCS microprocessor can then be determined from $f_{\rm c}$. We assign the lowest possible supply voltage that ensures reliable operation. Our technique is applicable to any synchronous devices controlled by the microprocessor, but we focus on the most common device, which is a synchronous memory. Our frequency-assignment technique is also applicable to an NSVS microprocessor. Since the mathematics underpinning our method is quite involved, we define all the symbols used in advance in Table I.

B. Energy Consumption of a Synchronous NSVS Memory

In addressing the problem of energy-aware memory clock-frequency assignment, we start by deriving generalized energy-consumption equations for a synchronous NSVS memory. The active-mode dynamic energy of such a memory is primarily determined by the number of transactions. The total number of external memory transactions is given by

$$N_{\rm m} = \varepsilon N_{l1} \tag{1}$$

where ε is cache miss ratio and N_{l1} is the total number of transactions between the microprocessor and the L1 cache. If the microprocessor is designed to stall when there is a cache miss, we can reasonably assume that the total execution time of a task will be given by

$$\tau_{\rm exe} \simeq \frac{N_{\rm c}}{f_{\rm c}} + \frac{N_{\rm m} M_b}{f_{\rm m}} \tag{2}$$

where $N_{\rm c}$ is the number of microprocessor cycles and M_b is the number of memory clocks for a burst-mode transaction. We do not consider the effect of instruction parallelism in this, even though a consideration of parallelism might lead to a tighter estimate of execution time.

Fig. 2 shows a generalized energy model of a synchronous NSVS memory. The total idle-to-active dynamic energy is determined by the number of external memory transactions, which is independent of $f_{\rm c}$ and $f_{\rm m}$, and can be expressed as

$$E_{{\rm AD}_T} = E_{\rm AD} N_{\rm m}. \tag{3}$$

TABLE I
NOMENCLATURE

N_c	Total number of microprocessor cycles with 100% cache hit ratio
N_m	Total number of external memory transactions
M_b	Number of memory clocks for a burst-mode transaction
f_c	Clock frequency of the microprocessor
f_m	Clock frequency of the memory
V_{dd}	Supply voltage
k	Coefficient of proportionality relating the supply voltage and f_c
f_{cmax}	Maximum possible clock frequency of the microprocessor
f_{cmin}	Minimum possible clock frequency of the microprocessor
f_{mmax}	Maximum possible clock frequency of the memory
τ_{exe}	Total execution time
τ_d	Deadline for task execution
E_{AS}	Active-mode static energy of the memory
E_{IS}	Idle-mode static energy of the memory
E_{AD}	Idle-to-active dynamic energy of the memory
E_{PCD}	Precharge dynamic energy of the memory
E_{ID}	Idle-to-idle dynamic energy of the memory
E_{WD}	Wake-up dynamic energy of the memory
E_{PDD}	Idle-to-powerdown dynamic energy of the memory
P_{IS}	Idle-mode static power of the memory
P_{AS}	Active-mode static power of the memory
P_{PDS}	Powerdown-mode static power of the memory
E_{CPU}	Energy consumption of the microprocessor
E_M	Energy consumption of the memory
α	Average switching activity of the microprocessor
C_{CPU}	Average switching capacitance of the microprocessor
$I_{leakage}$	Leakage current of the microprocessor
P_{ON}	Inherent power cost in keeping the microprocessor on

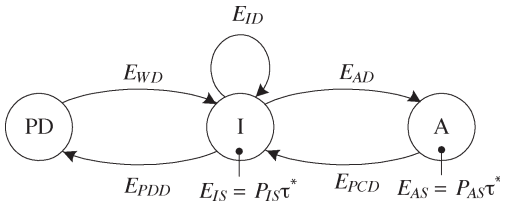


Fig. 2. Typical energy state machine representation of a synchronous memory ($*\tau$ is the clock period).

We assume the existence of autoprecharge scheme [13] that immediately sends the SDRAM to idle mode after every burst-mode access. The autoprecharge scheme is common in battery-operated systems. The total precharge energy is given by

$$E_{\mathrm{PCD}_T} = E_{\mathrm{PCD}} N_{\mathrm{m}}. \tag{4}$$

Current-mode circuits exhibit a distinct active-mode static energy but negligible leakage energy during idle mode. The active-mode static energy is directly proportional to the length of time that the circuit remains in active mode. It is determined by the number of external memory transactions, the clock period of the SDRAM, and the number of memory clock ticks required for a burst-mode transfer. Thus, the total active-mode static energy is given by

$$E_{\mathrm{AS}_T} = \frac{P_{\mathrm{AS}} N_{\mathrm{m}} M_b}{f_{\mathrm{m}}}. \tag{5}$$

Simple energy models usually consider the sum of three energy components, $E_{\mathrm{AD}_T} + E_{\mathrm{PCD}_T} + E_{\mathrm{AS}_T}$, without distinguishing them and also usually ignore the dynamic energy consumption during idle mode. However, a synchronous memory has a distinct dynamic energy consumption during idle mode, unlike an asynchronous memory, because of the necessity for clock propagation during the idle mode. The SDRAM is a slave device, and its idle mode is a ready state in which it waits for an external request. The total idle-to-idle dynamic energy consumption of an SDRAM is determined by the number of idle clock cycles, which is in turn determined by $\tau_{\mathrm{exe}} - (1/f_{\mathrm{m}}) N_{\mathrm{m}} M_b$. Thus

$$E_{\mathrm{ID}_T} = E_{\mathrm{ID}}(\tau_{\mathrm{exe}} f_{\mathrm{m}} - N_{\mathrm{m}} M_b) = \frac{E_{\mathrm{ID}} f_{\mathrm{m}} N_{\mathrm{c}}}{f_{\mathrm{c}}}. \tag{6}$$

This explains why unnecessarily high f_{m} increases the number of idle clock cycles and thus the total idle-to-idle dynamic energy. As f_{c} increases, the total execution time τ_{exe} decreases, and idle-to-idle dynamic energy consumption reduced. On the other hand, the total idle-mode static energy is solely determined by the duration of the idle mode

$$E_{\mathrm{IS}_T} = P_{\mathrm{IS}} \frac{1}{f_{\mathrm{m}}} (\tau_{\mathrm{exe}} f_{\mathrm{m}} - N_{\mathrm{m}} M_b) = \frac{P_{\mathrm{IS}} N_{\mathrm{c}}}{f_{\mathrm{c}}}. \tag{7}$$

As f_{c} increases, the total idle-mode energy decreases.

If the total execution time of a task τ_{exe} is less than a task deadline τ_d, then both the microprocessor and the memory should be in power-down mode to avoid unnecessary energy consumption. The energy overhead for power down and wake up ($E_{\mathrm{PDD}} + E_{\mathrm{WD}}$) is proportional to the number of power-down operations. We assume that the slack time $\tau_d - \tau_{\mathrm{exe}}$ occurs at the end of task execution, and that there is one power-down and one wake-up operation. The dynamic energy required to enter and leave the power-down mode can then be simplified

$$E_{\mathrm{PDD}_T} = E_{\mathrm{PDD}} \text{ and } E_{\mathrm{WD}_T} = E_{\mathrm{WD}}. \tag{8}$$

TABLE II
ENERGY VARIATION VERSUS f_c AND f_m FOR A GIVEN TASK

Energy component	CPU clock		Memory clock	
	⇑	⇓	⇑	⇓
E_{AD_T}	constant	constant	constant	constant
E_{PCD_T}	constant	constant	constant	constant
E_{ID_T}	⇓	⇑	⇑	⇓
E_{IS_T}	⇓	⇑	constant	constant
E_{AS_T}	constant	constant	⇓	⇑

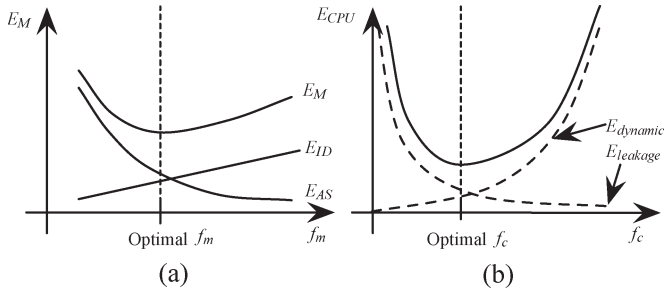


Fig. 3. Energy consumption of a synchronous NSVS memory and an SVCS microprocessor. (a) Memory. (b) Microprocessor.

During the slack time $\tau_d - \tau_{\rm exe}$, the SDRAM stays in power-down mode and consumes only a small amount of static energy

$$E_{{\rm PDS}_T} = P_{\rm PDS}(\tau_d - \tau_{\rm exe}) = P_{\rm PDS}\left(\tau_d - \frac{N_{\rm c}}{f_{\rm c}} - \frac{N_{\rm m}M_b}{f_{\rm m}}\right). \tag{9}$$

Note that $E_{{\rm PDD}_T} + E_{{\rm WD}_T} + E_{{\rm PDS}_T}$ is not significant.

Finally, the total energy consumption of the SDRAM is given as follows:

$$\begin{aligned} E_M = &\frac{E_{\rm ID} f_{\rm m} N_{\rm c}}{f_{\rm c}} + (E_{\rm AD} + E_{\rm PCD})N_{\rm m} \\ &+ \frac{P_{\rm AS} N_{\rm m} M_b}{f_{\rm m}} + \frac{P_{\rm IS} N_{\rm c}}{f_{\rm c}} + E_{\rm PDD} + E_{\rm WD} \\ &+ P_{\rm PDS}\left(\tau_d - \frac{N_{\rm c}}{f_{\rm c}} - \frac{M_b N_{\rm m}}{f_{\rm m}}\right). \end{aligned} \tag{10}$$

Table II summarizes how the energy-consumption behavior of a synchronous NSVS memory is affected by $f_{\rm c}$ and $f_{\rm m}$. In some circumstances, energy consumption increases with one or both of these frequencies, while in others, the relationship is inverted.

III. ENERGY-OPTIMAL FREQUENCY ASSIGNMENT

A. Energy-Frequency Relationship in a Synchronous NSVS Memory

We will now see how a synchronous NSVS memory behaves as the clock-frequency changes. As we vary $f_{\rm m}$ for a given $f_{\rm c}$, E_M responds as shown in Fig. 3(a). This variation is primarily caused by a tradeoff between $E_{\rm ID}$ and $E_{\rm AS}$. The total energy consumption E_M is convex, and thus there exists an energy-optimal $f_{\rm m}$.

We will now derive the clock frequency of a synchronous NSVS memory that corresponds to the least energy consumption for given values of $N_{\rm c}$, $N_{\rm m}$, and $f_{\rm c}$.

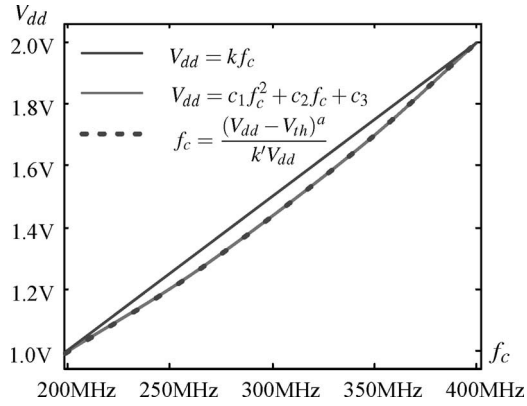


Fig. 4. Lowest allowable supply voltage corresponding to the microprocessor frequency.

Theorem 1: A synchronous NSVS memory has an energy-optimal clock frequency $f_{\rm m}$ such that

$$f_{\rm m} = \sqrt{\frac{(P_{\rm AS} - P_{\rm PDS}) f_{\rm c} N_{\rm m} M_b}{E_{\rm ID} N_{\rm c}}}. \tag{11}$$

Proof: Since E_M is a convex function, we can obtain the optimal value of $f_{\rm m}$ by taking the derivative of (10) for $f_{\rm m}$. ■

B. Energy Model of an SVCS Microprocessor

The power consumption of a microprocessor can be divided into dynamic power and static power, as shown in Fig. 3(b). We assume that the clock and power gating is well designed, so that the microprocessor does not consume appreciable energy during a stall state. However, the device leakage-power consumption increases with each technology generation advances. Recently, several groups have published papers about leakage-aware DVS [14], [15]. Different leakage sources that contribute to the total static-power consumption include subthreshold, reverse bias junction current, and gate direct tunneling current. To contain the scope of this paper, we have simplified microprocessor energy models by assuming that the leakage-power consumption is a linear function of the supply voltage. A simplified microprocessor power consumption is then given by

$$P_{\rm CPU} = \alpha C_{\rm CPU} V_{\rm dd}^2 f_{\rm c} + I_{\rm leakage} V_{\rm dd} + P_{\rm ON} \tag{12}$$

where α and $C_{\rm CPU}$ are the average switching activity and capacitance, respectively, $V_{\rm dd}$ is the supply voltage, $I_{\rm leakage}$ is the leakage current, and $P_{\rm ON}$ is the inherent power cost to keep the microprocessor ON. A maximum allowable frequency of the microprocessor can be expressed as

$$f = \frac{(V_{\rm dd} - V_{\rm th})^a}{k' V_{\rm dd}} \tag{13}$$

where $V_{\rm th}$ is the threshold voltage, k' and a are constants for a given technology process, and $1 < a \leq 2$ [16].

To derive an analytic solution, we simplified (13) using curve fitting. Fig. 4 shows an approximate solution where a, $V_{\rm th}$, and k' are 1.413, 0.4 V and $2.429E-9$, respectively. The delay and frequency model of (13) can be accurately modeled as a

second-order polynomial function. Although the second-order approximation is more accurate and we can use the second-order model without any problem, we will use a first-order model in the following treatment to make our analysis easier to understand.

We will assume that the lowest allowable supply voltage is linearly proportional to $f_{\rm c}$

$$V_{\rm dd} = k f_{\rm c} \tag{14}$$

where k is a coefficient of proportionality relating the supply voltage and $f_{\rm c}$. Using (12) and (14), the energy consumption of the microprocessor can be expressed as a function of $f_{\rm c}$ and $N_{\rm c}$, as follows:

$$\begin{aligned} E_{\rm CPU} &= P_{\rm CPU}\frac{N_{\rm c}}{f_{\rm c}} \\ &= \alpha C_{\rm CPU} k^2 f_{\rm c}^2 N_{\rm c} + I_{\rm leakage} k N_{\rm c} + P_{\rm ON}\frac{N_{\rm c}}{f_{\rm c}}. \end{aligned} \tag{15}$$

C. Frequency Assignment Without Considering the Energy Consumption of the Memory

Fig. 5 summarizes three different frequency-assignment schemes for the DVS of an SVCS microprocessor. Fig. 5(a) shows a conventional frequency-assignment scheme for the DVS. Here, $\tau_{\rm exe} = (1/2)\tau_d$ at $f_{\rm c} = f_{\rm c\,max}$, where $f_{\rm c\,max}$ is the maximum possible clock frequency of the microprocessor. Since this frequency-assignment scheme ignores both the energy consumption and the access time of a synchronous NSVS memory, $f_{\rm c}' = (1/2) f_{\rm c\,max}$ is applied over the period $\tau_{\rm exe}'$. But, this means that $\tau_{\rm exe}' < \tau_d$, because only $N_{\rm c}/f_{\rm c}$ is doubled by the assignment [7].

This approach does not make best use of the slack time, and Zhang and Chanson presented an improved frequency assignment, which takes into account the execution time of the memory [7]. As shown in Fig. 5(b), this assignment yields $\tau_{\rm exe}^{\#} = \tau_d$ and thus a lower value of $f_{\rm c}^{\#}$, making it feasible to satisfy the inequality $f_{\rm c}^{\#} < f_{\rm c}'$. However, eliminating the slack time is not always optimal. An excessively low microprocessor clock frequency increases the execution time of the microprocessor and the idle time of the memory, and thus increases the leakage energy consumption of the microprocessor and the idle-mode energy consumption of the memory, which can cancel out the energy saving in the microprocessor.

D. Frequency Assignment Considering the Energy Consumption of the Memory

The frequency-assignment schemes shown in Fig. 5(a) and (b) only try to minimize $E_{\rm CPU}$. In addition, $\tau_{\rm exe} = \tau_d$ does not correspond to an energy-optimal frequency assignment due to the convex energy behavior [9]. Since $E_{\rm CPU}$ and E_M are functions of $f_{\rm c}$ and $f_{\rm m}$, and $\tau_{\rm exe}$ is a function of $f_{\rm c}$ and $f_{\rm m}$ [(2), (10), and (15)], we can prove that there exists a system-wide energy-optimal frequency pair $(f_{\rm c}, f_{\rm m})$. Since $f_{\rm c}$ and $f_{\rm m}$ are cross-coupled, we need to derive $f_{\rm c}$ and $f_{\rm m}$ together, as shown in Fig. 5(c).

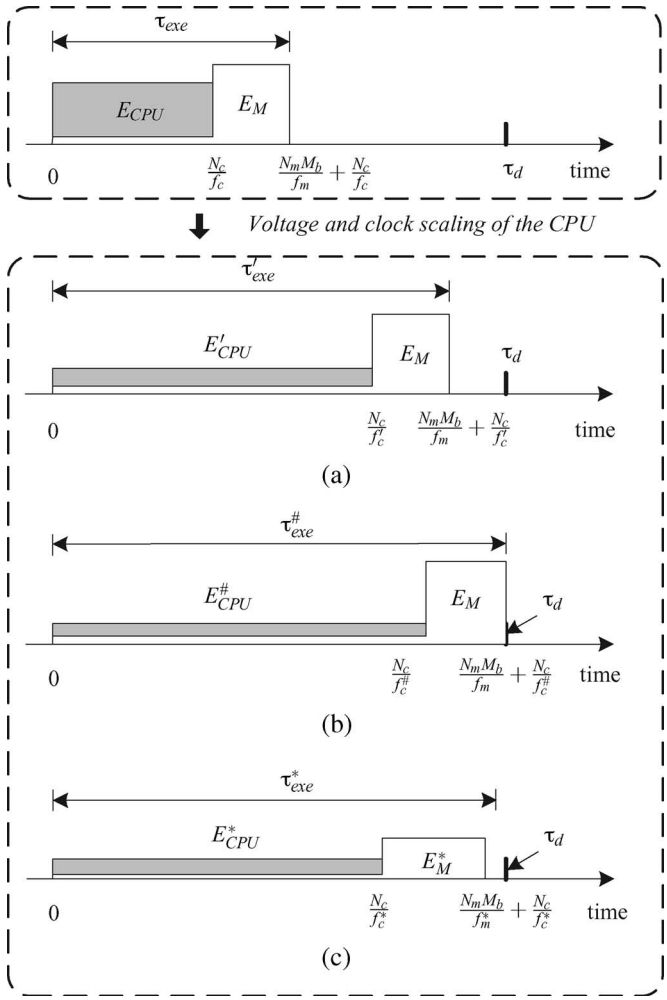


Fig. 5. Frequency assignment of a microprocessor and its memory. (a) Voltage assignment ignoring memory energy and access time. (b) Voltage assignment considering memory access time. (c) Voltage assignment considering memory energy and access time.

The total system energy required by the microprocessor and the memory E_T is given by

$$\begin{aligned} E_T &= E_{\rm CPU} + E_M \\ &= \alpha C_{\rm CPU} k^2 f_{\rm c}^2 N_{\rm c} + I_{\rm leakage} k N_{\rm c} \\ &\quad + \frac{N_{\rm c}}{f_{\rm c}} P_{\rm ON} + \frac{E_{\rm ID} f_{\rm m} N_{\rm c}}{f_{\rm c}} + (E_{\rm AD} + E_{\rm PCD}) N_{\rm m} \\ &\quad + \frac{(P_{\rm AS} - P_{\rm PDS}) N_{\rm m} M_b}{f_{\rm m}} + \frac{(P_{\rm ON} + P_{\rm IS} - P_{\rm PDS}) N_{\rm c}}{f_{\rm c}} \\ &\quad + P_{\rm PDS}\tau_d + E_{\rm PDD} + E_{\rm WD}. \end{aligned} \tag{16}$$

Note that E_T is convex because a positive weighted sum of convex functions is convex [17]. Fig. 6 shows convex contours of E_T in the space of $f_{\rm c}$ and $f_{\rm m}$.

There exists a unique pair $(f_{\rm c}, f_{\rm m})$ corresponding to the minimum value of E_T. But, we need to verify that this pair is feasible. The constraints on feasibility are summarized as follows:

$$\begin{aligned} &f_{\rm c}\text{ constraint:} && f_{\rm c\,min} \le f_{\rm c} \le f_{\rm c\,max}, \\ &f_{\rm m}\text{ constraint:} && f_{\rm m} \le f_{\rm m\,max}, \\ &\text{Deadline constraint:} && \tau_{\rm exe} \le \tau_d. \end{aligned} \tag{17}$$

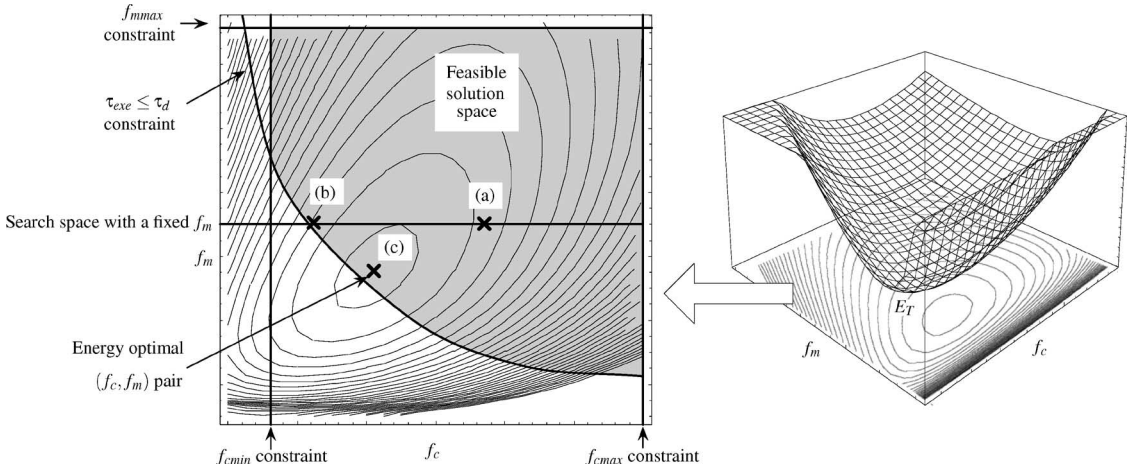


Fig. 6. Feasible solution space of the energy-optimal clock frequency. Contours denote E_T and are convex. The supply voltage of the microprocessor is the lowest possible that will ensure reliable operation.

The constraints on $f_{\mathrm{c\,max}}$ and $f_{\mathrm{m\,max}}$ come from the device data sheet. A lower bound on f_{m} may be determined by the dynamic logic structure, but it is too low to offset the feasibility of the optimal solution. However, an SVCS microprocessor has quite a high $f_{\mathrm{c\,min}}$, since the supply voltage of the microprocessor cannot be reduced beyond a certain limit, which is determined by its logic structure.

Optimization problems with constraints can in general be solved by a Lagrange multiplier [18], but we do not consider this approach because of its complexity. Instead, we derive the solution by dividing the problem into several subproblems. The constraints form four boundary conditions, as shown in Fig. 6. Except for the deadline constraints, the boundary conditions are trivial. If the optimal $(f_{\mathrm{c}}, f_{\mathrm{m}})$ pair exists in the feasible solution space, Theorem 2 determines the optimal solution, and the corresponding $(f_{\mathrm{c}}, f_{\mathrm{m}})$ can be obtained by solving a differential equation.

Theorem 2: A system equipped with an SVCS microprocessor and a synchronous NSVS memory has a system-wide energy-optimal $(f_{\mathrm{c}}, f_{\mathrm{m}})$ pair such that

$$4(\alpha C_{\mathrm{CPU}}k^2)^2 N_{\mathrm{c}} f_{\mathrm{c}}^6 - 4\alpha C_{\mathrm{CPU}}k^2 N_{\mathrm{c}}(P_{\mathrm{ON}} + P_{\mathrm{IS}} - P_{\mathrm{PDS}}) f_{\mathrm{c}}^3 - E_{\mathrm{ID}} N_{\mathrm{m}} M_b (P_{\mathrm{AS}} - P_{\mathrm{PDS}}) f_{\mathrm{c}} + N_{\mathrm{c}}(P_{\mathrm{ON}} + P_{\mathrm{IS}} - P_{\mathrm{PDS}})^2 = 0 \tag{18}$$

and

$$f_{\mathrm{m}} = \sqrt{\frac{(P_{\mathrm{AS}} - P_{\mathrm{PDS}}) f_{\mathrm{c}} N_{\mathrm{m}} M_{\mathrm{b}}}{E_{\mathrm{ID}} N_{\mathrm{c}}}} \tag{19}$$

if both frequencies are included in the feasible solution space determined by (17).

Proof: The optimal $(f_{\mathrm{c}}, f_{\mathrm{m}})$ pair is derived from the simultaneous equations $\partial E_T/\partial f_{\mathrm{c}} = 0$ and $\partial E_T/\partial f_{\mathrm{m}} = 0$, which can be rewritten

$$\frac{\partial E_T}{\partial f_{\mathrm{c}}} = 2\alpha C_{\mathrm{CPU}} k^2 N_{\mathrm{c}} f_{\mathrm{c}} - \frac{E_{\mathrm{ID}} N_{\mathrm{c}} f_{\mathrm{m}}}{f_{\mathrm{c}}^2} - \frac{(P_{\mathrm{ON}} + P_{\mathrm{IS}} - P_{\mathrm{PDS}}) N_{\mathrm{c}}}{f_{\mathrm{c}}^3} = 0 \tag{20}$$

and

$$\frac{\partial E_T}{\partial f_{\mathrm{m}}} = \frac{E_{\mathrm{ID}} N_{\mathrm{c}}}{f_{\mathrm{c}}} - \frac{N_{\mathrm{m}} M_b (P_{\mathrm{AS}} - P_{\mathrm{PDS}})}{f_{\mathrm{m}}^2} = 0. \tag{21}$$

We can derive the optimal f_{m} in (19) from the partial derivative of (21). Replacing f_{m} in (20) by the solution of (21), we obtain

$$\frac{\partial E_T}{\partial f_{\mathrm{c}}} = 2\alpha C_{\mathrm{CPU}} f_{\mathrm{c}} k^2 N_{\mathrm{c}} - \sqrt{\frac{E_{\mathrm{ID}}(P_{\mathrm{AS}} - P_{\mathrm{PDS}}) N_{\mathrm{m}} M_b N_{\mathrm{c}}}{f_{\mathrm{c}}^3}} - \frac{N_{\mathrm{c}}(P_{\mathrm{ON}} + P_{\mathrm{IS}} - P_{\mathrm{PDS}})}{f_{\mathrm{c}}^2} = 0. \tag{22}$$

The open-form solution of (22) is given by (18), and the positive real root of (18) is the optimal f_{c}. ■

Since (18) and the following polynomial equations are too complicated to have closed-form solutions, we solve them numerically.

Unfortunately, we may often encounter the situation that the pair of frequencies $(f_{\mathrm{c}}, f_{\mathrm{m}})$ obtained from Theorem 2 is not included in the feasible solution space. In this case, a suboptimal pair has to be found on the boundary of the feasible solution space. We obtain this solution by generating locally optimal pairs at intervals along all the boundary conditions and choosing the global optimum.

Corollary 1: A system equipped with an SVCS microprocessor and a synchronous NSVS memory has a system-wide energy-optimal $(f_{\mathrm{c}}, f_{\mathrm{m}})$ such that

$$f_{\mathrm{m}} = \sqrt{\frac{(P_{\mathrm{AS}} - P_{\mathrm{PDS}}) f_{\mathrm{c\,min}} N_{\mathrm{m}} M_b}{E_{\mathrm{ID}} N_{\mathrm{c}}}} \tag{23}$$

or

$$f_{\mathrm{m}} = \sqrt{\frac{(P_{\mathrm{AS}} - P_{\mathrm{PDS}}) f_{\mathrm{c\,max}} N_{\mathrm{m}} M_b}{E_{\mathrm{ID}} N_{\mathrm{c}}}} \tag{24}$$

if it lies on the boundary condition corresponding to $f_{\mathrm{c\,min}}$ or $f_{\mathrm{c\,max}}$. ■

Theorem 3: A system equipped with an SVCS microprocessor and a synchronous NSVS memory has a system-wide energy-optimal $(f_{\rm c}, f_{\rm m})$ such that

$$f_{\rm c} = \left(\frac{E_{\rm ID} f_{\rm m\,max} + P_{\rm ON} + P_{\rm IS} - P_{\rm PDS}}{2\alpha C_{\rm CPU} k^2} \right)^{\frac{1}{3}}$$

and

$$f_{\rm m} = f_{\rm m\,max} \tag{25}$$

if the optimum lies on the boundary condition corresponding to $f_{\rm m\,max}$.

Proof: The optimal $f_{\rm c}$ occurs when

$$\frac{\partial E_T}{\partial f_{\rm c}} = N_{\rm c} \left(2\alpha C_{\rm CPU} f_{\rm c} k^2 - \left(\frac{E_{\rm ID} f_{\rm m}}{f_{\rm c}^2} + \frac{P_{\rm ON} + P_{\rm IS} - P_{\rm PDS}}{f_{\rm c}^2} \right) \right) = 0. \tag{26}$$

■

Frequency assignment of an SVCS microprocessor using conventional DVS implies a fixed $f_{\rm m}$. The frequency-assignment schemes shown in Fig. 5(a) and (b) address this problem. However, neither of these schemes arrives at system-wide energy optima, but the value of $f_{\rm c}$ corresponding to the lowest energy consumption with a fixed $f_{\rm m}$ can be achieved by the following strategy:

$$\begin{aligned} &\text{minimize} && E_T \\ &\text{subject to} && f_{\rm c\,min} \le f_{\rm c} \le f_{\rm c\,max} \\ &&& \text{with a constant } f_{\rm m}. \end{aligned} \tag{27}$$

Corollary 2—Frequency assignment in conventional DVS*:* A system equipped with an SVCS microprocessor and a synchronous NSVS memory has a system-wide energy-optimal $f_{\rm c}$ with a fixed $f_{\rm m}$ such that

$$f_{\rm c} = \left(\frac{E_{\rm ID} f_{\rm m} + P_{\rm ON} + P_{\rm IS} - P_{\rm PDS}}{2\alpha C_{\rm CPU} k^2} \right)^{\frac{1}{3}}. \tag{28}$$

Proof: Since $f_{\rm m}$ is a given constant value, we can obtain the optimal $f_{\rm c}$ by replacing $f_{\rm m\,max}$ with $f_{\rm m}$ in Theorem 3. ■

Sometimes, an energy-optimal $(f_{\rm c}, f_{\rm m})$ is not feasible due to a deadline miss. In this case, a suboptimal $(f_{\rm c}, f_{\rm m})$ can be obtained by the following procedure:

$$\begin{aligned} &\text{minimize} && E_T \\ &\text{subject to} && f_{\rm c\,min} \le f_{\rm c} \le f_{\rm c\,max} \\ &&& f_{\rm m} \le f_{\rm m\,max} \\ &&& \tau_{\rm exe} = \tau_d. \end{aligned} \tag{29}$$

The resulting energy-suboptimal $(f_{\rm c}, f_{\rm m})$ is on the boundary condition of the deadline constraint $\tau_{\rm exe} \le \tau_d$ in the feasible solution space.

Theorem 4: A system equipped with an SVCS microprocessor and a synchronous NSVS memory has a system-wide energy-optimal $(f_{\rm c}, f_{\rm m})$ such that

$$\begin{aligned} &2\alpha C_{\rm CPU} k^2 \tau_d^2 f_{\rm c}^5 - 4\alpha C_{\rm CPU} k^2 N_{\rm c} \tau_d f_{\rm c}^4 + 2\alpha C_{\rm CPU} k^2 N_{\rm c}^2 f_{\rm c}^3 \\ &+ \left((P_{\rm AS} - P_{\rm IS} - P_{\rm ON}) \tau_d^2 - E_{\rm ID} N_{\rm m} M_b \tau_d \right) f_{\rm c}^2 + 2N_{\rm c} \\ &\times (P_{\rm IS} + P_{\rm ON} - P_{\rm AS}) \tau_d f_{\rm c} + N_{\rm c}^2 (P_{\rm AS} - P_{\rm IS} - P_{\rm ON}) = 0 \end{aligned} \tag{30}$$

and

$$f_{\rm m} = \frac{N_{\rm m} M_b f_{\rm c}}{\tau_d f_{\rm c} - N_{\rm c}} \tag{31}$$

if the optimum lies on the boundary condition corresponding to the inequality $\tau_{\rm exe} \le \tau_d$.

Proof: On the deadline boundary condition, $\tau_{\rm exe}$ is given by

$$\tau_d = \frac{N_{\rm c}}{f_{\rm c}} + \frac{N_{\rm m} M_b}{f_{\rm m}} \tag{32}$$

and we have

$$f_{\rm m} = \frac{N_{\rm m} M_b f_{\rm c}}{\tau_d f_{\rm c} - N_{\rm c}}. \tag{33}$$

Thus, E_T is given by

$$\begin{aligned} E_T = {} & \alpha C_{\rm CPU} k^2 f_{\rm c}^2 N_{\rm c} + I_{\rm leakage} k N_{\rm c} + P_{\rm ON} \frac{N_{\rm c}}{f_{\rm c}} \\ & + (E_{\rm AD} + E_{\rm PCD}) N_{\rm m} + \frac{E_{\rm ID} N_{\rm c} N_{\rm m} M_b}{\tau_d f_{\rm c} - N_{\rm c}} \\ & + P_{\rm AS} \left(\tau_d - \frac{N_{\rm c}}{f_{\rm c}} \right) + \frac{P_{\rm IS} N_{\rm c}}{f_{\rm c}}. \end{aligned} \tag{34}$$

The optimal $f_{\rm c}$ corresponds to the solution of $\partial E_T / \partial f_{\rm c} = 0$. ■

IV. Discrete Frequency Assignment

It is too expensive to make an embedded system with DVS that allows continuous voltage and frequency changes. Discrete voltage and frequency changes can achieve a near-optimal configuration while significantly reducing implementation complexity [19]. With the existing DVS techniques, discrete voltage and frequency assignments do not require a significantly different approach from continuous frequency assignment: The next available discrete frequency above the calculated continuous frequency is selected.

However, using memory-aware DVS, we cannot select $f_{\rm c}$ in this way, because the convexity property is no longer valid in the discrete solution space. Although the continuous cost function remains convex, the space of discrete frequency pairs cannot be convex, as shown in Fig. 7. In other words, the globally optimal discrete frequency pair cannot be found by greedy search.

In general, this sort of problem can be solved by simulated annealing, branch and bound techniques, or genetic algorithms

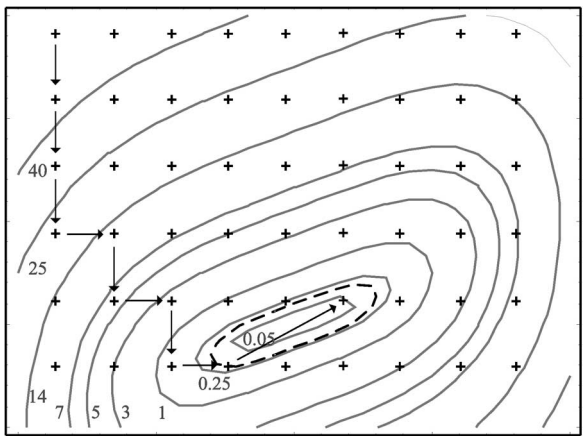


Fig. 7. Example of a discrete solution space.

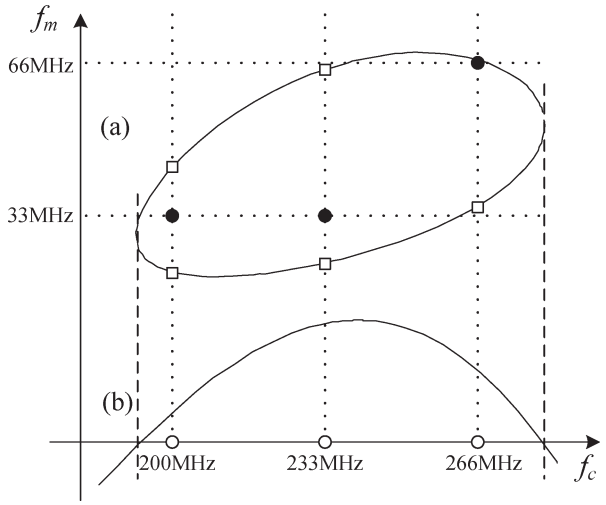


Fig. 8. Reduction of the solution space by a contour, which includes the local minimum.

[20]. However, the size of the solution space is not large enough to justify such methods. There are fewer than a hundred frequency pairs in most cases. Instead, we introduce a new approach that is more suitable for this problem. First, we find a local optimum by greedy search, beginning from a random starting point, and then we find the contour that has the same altitude as the local optimum. Finally, we determine the globally optimal frequency pair by examining the area inside this contour.

A. Symbolic Contour Equation

The key idea of this approach is to reduce the search space by finding a contour that has the same altitude as the local optimum. Because the continuous solution space is convex, the global optimum must lie within that contour. Points representing discrete frequency pairs that are outside the contour correspond to higher energy consumption than those inside. We can afford to perform exhaustive search for the global optimum within the dramatically reduced search space enclosed by this contour.

We set up the contour equation with the local minimum energy value $E_{\rm local}$ as follows:

$$E_T(f_{\rm m}, f_{\rm c}) = E_{\rm local}. \tag{35}$$

This contour equation can be expressed as a second-order polynomial function of $f_{\rm m}$ by multiplying both sides by $f_{\rm m}f_{\rm c}$ as follows:

$$\begin{aligned} E_T \cdot f_{\rm m}f_{\rm c} =& \alpha C_{\rm CPU}k^2N_{\rm c}f_{\rm c}^3f_{\rm m} + I_{\rm leakage}kN_{\rm c}f_{\rm c}f_{\rm m} \\ &+ N_{\rm c}P_{\rm ON}f_{\rm m} + E_{\rm ID}N_{\rm c}{f_{\rm m}}^2 \\ &+ (E_{\rm AD}+E_{\rm PCD})N_{\rm m}f_{\rm c}f_{\rm m} \\ &+ (P_{\rm AS}-P_{\rm PDS})N_{\rm m}M_bf_{\rm c} \\ &+ (P_{\rm ON}+P_{\rm IS}-P_{\rm PDS})N_{\rm c}f_{\rm m} \\ &+ (P_{\rm PDS}\tau_d + E_{\rm PDD}+E_{\rm WD})f_{\rm c}f_{\rm m} \\ =& E_{\rm local}\cdot f_{\rm m}f_{\rm c}. \end{aligned} \tag{36}$$

TABLE III
ENERGY CONSUMPTION OF A 32-BIT MEMORY WITH FOUR MICRON MT48LC16M8A2 SDRAMs ($M_b = 9$)

E_{AD}	110.8 (nJ/access)	E_{PCD}	15.08 (nJ/access)
E_{ID}	3.14 (nJ/cycle)	E_{WD}	0 (nJ/access)
E_{PDD}	0 (nJ/access)	P_{IS}	0.072 (nJ/ns)
P_{AS}	0.151 (nJ/ns)	P_{PDS}	0.0116 (nJ/ns)

We can rewrite this quadratic equation in a standard form

$$af_{\rm m}^2 + bf_{\rm m} + c = 0 \tag{37}$$

where

$$\begin{aligned} a =& E_{\rm ID}N_{\rm c} \\ b =& \alpha C_{\rm CPU}k^2N_{\rm c}f_{\rm c}^3 + I_{\rm leakage}kN_{\rm c}f_{\rm c} \\ &+ ((E_{\rm AD}+E_{\rm PCD})N_{\rm m} + P_{\rm PDS}\tau_d + E_{\rm PDD} + E_{\rm WD} - E_{\rm local})f_{\rm c} \\ &+ (P_{\rm ON}+P_{\rm IS}-P_{\rm PDS})N_{\rm c} \\ c =& (P_{\rm AS}-P_{\rm PDS})N_{\rm m}M_b. \end{aligned} \tag{38}$$

This has the well-known closed-form solution

$$f_{\rm m}(f_{\rm c}) = \frac{-b \pm \sqrt{g(f_{\rm c})}}{2a} \tag{39}$$

where

$$g(f_{\rm c}) = b^2 - 4ac. \tag{40}$$

Real roots exist only when $g(f_{\rm c}) \geq 0$, and so the globally optimal $f_{\rm c}$ must exist within the boundary defined by $g(f_{\rm c}) = 0$. Equation (40) is a sixth-order polynomial in $f_{\rm c}$ and would need to be solved numerically. But, we only need to find out whether each discrete point satisfies $g(f_{\rm c}) \geq 0$ [marked (b) in Fig. 8]. Then, we can use (39) to derive the upper and lower limits on $f_{\rm m}$ for each $f_{\rm c}$ [marked (a) in Fig. 8]. Finally, we can determine the global optimum $(f_{\rm c}, f_{\rm m})$ from all the points inside the contour. This does require exhaustive search, but it is relatively fast, owing to the reduced search space.

TABLE IV
FREQUENCY ASSIGNMENT AGAINST SYSTEM-WIDE ENERGY REDUCTION
(f_c AND f_m ARE IN MEGAHERTZ, AND E_{CPU}, E_M, AND E_T ARE IN MICROJOULES)

MPEG4 decoder: $N_c = 7.4 \times 10^6$, $N_m = 81,208$ and $\tau_d = 50$ms

Frequency assignment	f_c	f_m	E_{CPU}	E_M	E_T	Reduction	τ_{exe}
No DVS	400	66	13,690	17,058	30,748		29.6
(a)	237	66	6,631	20,626	27,257	11.4%	42.4
(b)	200	66	5,735	22,224	27,959	9.1%	48.1
(c)	241	38	6,756	18,997	25,753	16.2%	50.0

JPEG decompressor: $N_c = 24.4 \times 10^6$, $N_m = 369,782$ and $\tau_d = 115$ms

Frequency assignment	f_c	f_m	E_{CPU}	E_M	E_T	Reduction	τ_{exe}
No DVS	400	66	44,400	70,907	115,307		110.0
(a)	384	66	41,530	71,601	113,130	1.9%	113.0
(b)	372	66	39,380	72,184	111,564	3.2%	115.0
(c)	325	81	32,459	76,881	108,865	5.6%	115.0

MP3 decoder: $N_c = 2 \times 10^6$, $N_m = 1,377$ and $\tau_d = 25$ms

Frequency assignment	f_c	f_m	E_{CPU}	E_M	E_T	Reduction	τ_{exe}
No DVS	400	66	3,700	1,598	5,298		5.2
(a)	200	66	1,550	2,994	4,544	14.2%	10.2
(b)	200	66	1,550	2,994	4,544	14.2%	10.2
(c)	200	8	1,550	1,378	2,928	44.7%	11.6

(a) Voltage assignment ignoring memory energy and access time
(b) Voltage assignment considering only memory access time
(c) Voltage assignment considering memory energy and access time (our scheme)

As described above, our algorithm consists of two parts: a greedy search and an exhaustive search. The greedy search has a linear complexity $O(n_{\rm m} + n_{\rm c})$, where $n_{\rm c}$ is the number of microprocessor frequency steps and $n_{\rm m}$ is the number of memory clock-frequency steps. Subsequently, we can find the global optimum within the contour by means of an exhaustive search. There are usually just one or two discrete points within the contour, although that cannot be guaranteed.

We could also find the global optimum using only an exhaustive search algorithm. The number of the discrete voltage pairs is often less than a hundred, when DVS is implemented using an embedded clock generator in a microprocessor. However, depending on the system configuration, each resolution of CPU and memory clock frequency can be greater than a hundred when we use external clock generators, because typical off-chip clock generators have high-resolution frequency synthesizers. Furthermore, our proposed algorithm has a linear complexity, while the exhaustive search has a quadratic complexity $O(n_{\rm m} n_{\rm c})$.

V. EXPERIMENTAL RESULTS

We evaluated the effect of the different frequency-assignment schemes on system-wide energy consumption for DVS [shown in Fig. 5(a)–(c)] and also compared them with the result of frequency assignment without DVS. The target system consists of a 32-bit RISC microprocessor and Micron 128-Mbit SDRAMs [21] but no other peripherals. The microprocessor is SVCS, and we assume that its energy consumption conforms to (15). We also know that it operates at a maximum frequency of 400 MHz and a minimum frequency of 200 MHz, consuming 740 mW when $V_{\rm dd} = 2$ V and 155 mW when $V_{\rm dd} = 1$ V, such that $\alpha C_{\rm CPU} k^2 = 1 \times 10^{-17}$ nJ/Hz^2. There must be a leakage current and an inherent power cost of keeping the microprocessor ON, and for these, we assume conservative values of $I_{\rm leakage} = 25$ mA and $P_{\rm ON} = 50$ mW. The microprocessor has the ARM instruction-set architecture and separate 8KB I-cache and 8KB D-cache. The memory has a 32-bit data width composed of four Micron SDRAMs. The default $f_{\rm m}$ is 66 MHz, which is the most popular setting for battery-operated systems. Frequency assignment using previous DVS methods is achieved by changing $f_{\rm c}$ while keeping $f_{\rm m} = 66$ MHz, and frequency assignment without DVS uses the settings $f_{\rm c} = f_{\rm c\,max}$ and $f_{\rm m} = 66$ MHz.

Table III summarizes the energy consumption of the Micron SDRAMs using the data acquired by cycle-accurate measurement [22]. We use the values in Table III to derive the energy-optimal $(f_{\rm c}, f_{\rm m})$ pairs using the theorems and corollaries presented earlier in this paper. Once we have derived the energy-optimal $(f_{\rm c}, f_{\rm m})$ pairs, we can use them to perform trace-driven cycle-by-cycle energy simulations with real applications. The $(f_{\rm c}, f_{\rm m})$ pairs may not be true optima due to modeling errors, but the benchmarks still represent a fair comparison.

We selected three embedded applications: an MPEG4 decoder running at 20 frames/s, a JPEG decompressor with 96% utilization, and an MP3 decoder streaming at 320 Kb/s. Table IV summarizes the performance of the frequency-assignment techniques in a continuous solution space.

In the case of the MPEG4 decoder (shown in Table IV), our optimal frequency assignment arrives at a higher microprocessor clock frequency but a lower memory clock frequency than conventional DVS. This results in 10% more energy consumption in the microprocessor but 20% lower energy consumption in the memory. As a result, this reduces the total energy consumption by 10% compared with the conventional

TABLE V
DISCRETE FREQUENCY ASSIGNMENT AGAINST SYSTEM-WIDE ENERGY REDUCTION (f_c AND f_m ARE IN MEGAHERTZ, AND E_{CPU}, E_M, AND E_T ARE IN MICROJOULES)

MPEG4 decoder: $N_c = 7.4 \times 10^6$, $N_m = 81,208$ and $\tau_d = 50$ms

Frequency assignment	f_c	f_m	E_{CPU}	E_M	E_T	Reduction	τ_{exe}
No DVS	400	66	13,690	17,058	30,748		29.6
(a)	266	66	7,575	19,641	27,216	11.5%	38.8
(b)	200	66	5,735	22,224	27,959	9.1%	48.1
(c)	266	33	7,575	18,431	26,006	15.4%	49.7

JPEG decompressor: $N_c = 24.4 \times 10^6$, $N_m = 369,782$ and $\tau_d = 115$ms

Frequency assignment	f_c	f_m	E_{CPU}	E_M	E_T	Reduction	τ_{exe}
No DVS	400	66	44,400	70,907	115,307		110.0
(a)	400	66	44,400	70,907	115,307	0%	110.0
(b)	400	66	44,400	70,907	115,307	0%	110.0
(c)	300	100	28,600	82,447	111,047	3.7%	113.0

MP3 decoder: $N_c = 2 \times 10^6$, $N_m = 1,377$ and $\tau_d = 25$ms

Frequency assignment	f_c	f_m	E_{CPU}	E_M	E_T	Reduction	τ_{exe}
No DVS	400	66	3,700	1,598	5,298		5.2
(a)	200	66	1,550	2,994	4,544	14.2%	10.2
(b)	200	66	1,550	2,994	4,544	14.2%	10.2
(c)	200	33	1,550	1,996	3,546	33.1%	10.4

(a) Voltage assignment ignoring memory energy and access time
(b) Voltage assignment considering only memory access time
(c) Voltage assignment considering memory energy and access time (our scheme)

DVS. In contrast, the JPEG application has only 4% slack time before DVS is applied, and thus conventional DVS can only achieve 2% to 3% energy saving. In this example, our optimal frequency assignment increases the memory clock frequency to 81 MHz, allowing further reduction of the microprocessor clock frequency by about 50 MHz. This achieves about 2× energy saving compared to the conventional DVS. The third, the MP3 application involves distinctly small number of memory transactions, so the optimal memory clock frequency is extremely low, which allows our optimal frequency assignment to achieve 45% saving in the total system energy. In summary, the optimal memory clock frequency can be either higher or lower than the initial setting, depending on the memory utilization. When the memory utilization is relatively low in applications, such as MPEG and MP3 in this paper, lower memory clock frequency is better. Our optimal frequency assignment systematically derives the energy optimal frequency pair according to system energy models and application characteristics.

Table V summarizes the performance of our technique for discrete frequency assignment, which shows a generally similar performance to the continuous DVS. The previous DVS techniques are, in general, not capable of finding the optimal setting, although optima were achieved in some cases.

VI. CONCLUSION

We have presented a new way to derive the energy-optimal frequency assignment for a system consisting of SVCS microprocessor and a synchronous NSVS memory device. We derived energy-optimal microprocessor and memory clock frequencies based on accurate energy models. We formulated analytical models of energy consumption by the microprocessor and the memory, and derived generalized solutions under various feasibility constraints, including limits on clock frequencies and task deadlines. We also presented a heuristic algorithm for the case of discrete frequencies, which guarantees achieving a globally optimal solution. Our frequency assignment enhances the previous DVS techniques and can save 50% more system-wide energy than those schemes.

ACKNOWLEDGMENT

The Institute of Computer Technology at Seoul National University provided research facilities for this paper. No part of this report may be read without permission of MIC and the IT R&D Project assessment and management.

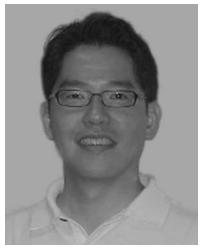
REFERENCES

[1] Y. Cho and N. Chang, "Memory-aware energy-optimal frequency assignment for dynamic supply voltage scaling," in *Proc. Int. Symp. Low Power Electron. Des.*, 2004, pp. 387–392.
[2] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," in *Proc. 36th Symp. Foundations Comput. Sci.*, 1995, pp. 374–382.
[3] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," in *Proc. Int. Symp. Low Power Electron. and Des.*, 1998, pp. 197–202.
[4] H. Shim, Y. Cho, and N. Chang, "Power saving in hand-held multimedia systems using MPEG-21 digital item adaptation," in *Proc. IEEE Workshop ESTIMedia*, 2004, pp. 13–18.
[5] T. L. Martin and D. P. Siewiorek, "Nonideal battery and main memory effects on CPU speed-setting for low power," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 1, pp. 29–34, Feb. 2001.
[6] B. Egger, J. Lee, and H. S. Shin. (2004). "An application-specific and adaptive power management technique," in *Proc. 1st Int. Workshop Power-Aware Real-Time Comput.* [Online]. Available: http://www.cs.pitt.edu.PARC/
[7] F. Zhang and S. T. Chanson, "Processor voltage scheduling for real-time tasks with non-preemptible sections," in *Proc. Real-Time Syst. Symp.*, 2002, pp. 235–245.

[8] T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. D. Micheli, "Dynamic voltage scaling and power management for portable systems," in *Proc. 38th Des. Autom. Conf.*, 2001, pp. 524–529.
[9] X. Fan, C. S. Ellis, and A. R. Lebeck, "The synergy between power-aware memory systems and processor voltage," in *Proc. Power-Aware Comput. Syst.*, 2003, pp. 164–179.
[10] R. Jejurikar and R. Gupta, "Dynamic voltage scaling for systemwide energy minimization in real-time embedded systems," in *Proc. Int. Symp. Low Power Electron. and Des.*, 2004, pp. 78–81.
[11] J. Zhuo and C. Chakrabarti, "System-level energy-efficient dynamic task scheduling," in *Proc. 42nd Des. Autom. Conf.*, 2005, pp. 628–631.
[12] Y. Choi, N. Chang, and T. Kim, "DC–DC converter-aware power management for battery-operated embedded systems," in *Proc. 42nd Des. Autom. Conf.*, 2005, pp. 895–900.
[13] H. Shim, Y. Joo, Y. Choi, H. G. Lee, and N. Chang, "Low-energy off-chip SDRAM memory systems for embedded applications," *Trans. Embedded Comput. Syst.*, vol. 2, no. 1, pp. 98–130, Feb. 2003.
[14] R. Jejurikar, C. Pereira, and R. Gupta, "Leakage aware dynamic voltage scaling for real-time embedded systems," in *Proc. 41st Des. Autom. Conf.*, 2004, pp. 275–280.
[15] L. Niu and G. Quan, "Reducing both dynamic and leakage energy consumption for hard real-time systems," in *Proc. Int. Conf. Compilers, Architecture, and Synthesis Embedded Syst.*, 2004, pp. 140–148.
[16] J. L. Wong, G. Qu, and M. Potkonjak, "Power minimization in QoS sensitive systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 6, pp. 553–561, Jun. 2004.
[17] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[18] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 2nd ed. Hoboken, NJ: Wiley, 1997.
[19] F. Xie, M. Martonosi, and S. Malik, "Compile-time dynamic voltage scaling settings: Opportunities and limits," in *Proc. ACM SIGPLAN Conf. Program. Language Des. and Implementation*, 2003, pp. 49–62.
[20] T. Wang, "Global optimization for constrained nonlinear programming," Ph.D. dissertation, Dept. Comput. Sci., Univ. Illinois, Urbana, IL, Dec. 2000.
[21] Micron Technology Inc., *Datasheet of Micron Synchronous SDRAM (MT48LC16M8A2)*, 2003.
[22] Y. Joo, Y. Choi, H. Shim, H. G. Lee, K. Kim, and N. Chang, "Energy exploration and reduction of SDRAM memory systems," in *Proc. 39th Des. Autom. Conf.*, 2002, pp. 892–897.

Youngjin Cho (S'03) received the B.S. degrees in computer science and engineering from Seoul National University, Seoul, Korea, where he is currently working toward the Ph.D. degree at the School of Computer Science and Engineering.

His research interests include embedded-systems design and low-power systems design.



Naehyuck Chang (SM'05) received the B.S., M.S., and Ph.D. degrees in control and instrumentation engineering from Seoul National University, Seoul, Korea.

He is an Associate Professor in the School of Computer Science and Engineering, Seoul National University. His research interests include system-level low-power design and embedded-systems design.

Dr. Chang is a Senior Member of the Association for Computing Machinery.