

Memory-Aware Energy-Optimal Frequency Assignment for Dynamic Supply Voltage Scaling

Youngjin Cho and Naehyuck Chang*

[†]*School of Computer Science & Engineering, Seoul National University, Korea
naehyuck@snu.ac.kr*

ABSTRACT

Dynamic supply voltage scaling (DVS) is one of the best ways to reduce the energy consumption of a device when there is a super-linear relationship between energy and supply voltage, and a pseudo-linear relationship between delay and supply voltage. However, most DVS schemes scale the clock frequency of the supply-voltage-clock-scalable (SVCS) CPU only and do not address the energy consumption of the memory. The memory is generally non-supply-voltage-scalable (NSVS), but its energy consumption is variable to its clock frequency and the total execution time. Thus, DVS for an SVCS CPU cannot achieve an optimal system-wide energy saving without consideration of the memory, as far as it is controlled by an SVCS CPU.

We introduce an energy-optimal frequency assignment, for both an SVCS CPU and a synchronous NSVS memory, which optimizes the system-wide energy consumption. We derive the energy-optimal clock frequencies for an SVCS CPU and a synchronous NSVS memory, as a function of the number of processor clock cycles, the number of memory accesses and the hardware energy model. Our technique modifies the frequency assignment of the CPU and the memory used in previous DVS schemes, which ignore the memory energy. It enables the system-wide energy-optimal settings and achieves additional 50% energy reduction over previous DVS schemes. This technique can also be applicable to synchronous NSVS peripheral devices.

Categories and Subject Descriptors

B.8.2 [Performance and Reliability]: Performance Analysis and Design Aids; C.4 [Performance of Systems]

General Terms

Design, Experimentation, Performance

Keywords

Low power, memory system, SDRAM

*Corresponding author.

[†]The ICT at Seoul National University provides research facilities for this study. This work was partly supported by the Brain Korea 21 Project.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISLPED'04, August 9–11, 2004, Newport Beach, California, USA.

Copyright 2004 ACM 1-58113-929-2/04/0008 ...\$5.00.

1. INTRODUCTION

Lossless energy reduction for digital systems is based on power management during slack times. Dynamic Power Management (DPM) shuts down unused devices, while taking into consideration the energy overhead for shut down and wake up. Voltage-mode CMOS logic devices generally allow supply voltage scaling, with a super-linear relationship between the energy consumption and the supply voltage, and a pseudo-linear relationship between the delay and the supply voltage. Dynamic supply-voltage scaling (DVS) is one of the best ways to utilize slack time for energy saving in the case of supply-voltage-clock-scalable (SVCS) devices [1] of this sort.

Most DVS schemes proposed so far only consider SVCS CPUs. Certainly, many devices include current-mode circuits that do not allow supply voltage variation. And, even if a device allows DVS, there is no energy gain unless there is a super-linear relationship between the supply voltage and the energy consumption. Furthermore, most I/O bus standards specify the logic swing, and any scaling of the logic swing would violate the standard. Consequently, DVS is only effective on the SVCS CPUs. So far, most DVS research elaborates on novel scheduling algorithms based on a frequency assignment which ignores the memory.

The effect of non-supply voltage-scalable (NSVS) memory technology on DVS has lately been demonstrated [2]. As the complexity of systems increases, even in battery-operated applications, the energy consumption of the CPU is no longer dominant. Unfortunately, scaling down the supply voltage and thus the clock frequency of the CPU decreases the slack times of synchronous NSVS memory, which may even increase the energy consumption of those devices. Thus, existing DVS techniques can easily fail to achieve system-wide optimization of energy consumption. Applying DPM to NSVS components can mitigate the energy overhead caused by the extended execution time resulting from DVS. The merger of DPM and DVS within a predictive power control strategy has therefore been proposed as an approach to system-wide optimization [3]. More specifically, a recent approach to DVS takes the memory access time into account when the CPU frequency is determined [4], and a power-aware memory system using DPM can significantly improve the energy economy of NSVS memory when DVS is applied to an SVCS CPU [5].

Although the combination of DVS and DPM overcomes the practical limitations of DVS to an extent, it does not guarantee a system-wide optimal configuration in terms of energy consumption. In this paper, we aim at the more limited goal of a system-wide energy-optimal frequency assignment for an SVCS CPU and a synchronous NSVS memory, when both of them are awake. For this purpose, we derive the relationship between energy consumption and clock frequencies for the SVCS CPU and the synchronous NSVS memory, based on the both the static and dynamic energy consumption. We prove that there exists an energy-optimal pair of clock frequencies for the SVCS CPU and the synchronous NSVS memory, and that is a function of the number of CPU clock

Table 1: Nomenclature

N_c	Total number of CPU cycles with a 100% cache hit ratio
N_{I1}	Total number of transactions between the CPU and the L1 cache
ϵ	Cache miss ratio
N_m	Total number of external memory transactions
M_b	Number of memory clocks for a burst-mode transition
f_c	Clock frequency of the CPU
f_m	Clock frequency of the memory
V_{dd}	Supply voltage
k	Proportional coefficient between the supply voltage and f_c
f_{cmax}	Maximum possible clock frequency of the CPU
f_{cmin}	Minimum possible clock frequency of the CPU by $V_{dd} = kf_c$
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 CPU
α	Average switching activity of the CPU
C_{cpu}	Average switching capacitance of the CPU

cycles, the number of memory transactions and the parameterized hardware energy model. We derive generalized analytical solution which determines the optimal frequencies for various constraints. This clock frequency assignment technique modifies previous DVS schemes and achieves a system-wide energy optimum. We demonstrate practical use of frequency assignment, with a commercially available SDRAM running real applications. The experimental results show an additional 50% energy reduction in comparison with traditional DVS schemes that ignore the memory energy.

2. PRINCIPLE OF MEMORY-AWARE OF FREQUENCY ASSIGNMENT

2.1 Problem statement

Up to now, there has been no proper strategy introduced to determine the clock frequency of a synchronous NSVS memory which is controlled by a SVCS CPU, subject to low energy consumption. Mostly, DVS schemes for an SVCS CPU simply ignores the memory and assume that the execution time and energy consumption of the system are solely determined by the CPU. A recent scheme does consider the proportion of total execution time occupied by memory access, and use this to pursue a lower clock frequency and hence a lower supply voltage for the CPU [4]. However, existing schemes do not consider how the energy consumption of the memory varies with the clock frequency change of the CPU.

For real cases, 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 does change with clock frequency due to leakage energy during the active state and dynamic energy during the idle state. (The detailed energy consumption of a synchronous NSVS memory will be discussed in Section 2.2.) Thus, when we assign a clock frequency to the CPU, we must also consider the clock frequency of the memory. Since both the CPU and the memory clock frequencies affect the execution time, a pair of the energy-optimal frequencies must be derived together.

Our method of clock frequency assignment determines a pair of energy-optimal frequencies for the CPU and the memory that minimize the energy consumption of the entire system, with a given

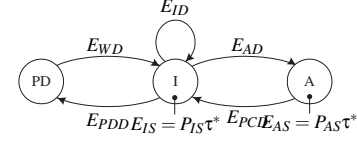


Figure 1: Typical energy state machine of a synchronous memory (* τ is the clock period).

application task and hardware configuration. The supply voltage of the synchronous NSVS memory is fixed, and the supply voltage of the SVCS CPU can then be determined by its clock frequency. We assign the lowest possible supply voltage that ensures reliable operation. Even if the energy-optimal clock frequencies are not feasible due to deadline miss, we can find feasible suboptimal values. Our technique is applicable to any synchronous devices controlled by the CPU, but we focus on synchronous memory for consistency (but without loss of generality). Our frequency assignment technique is also applicable to an NSVS CPU. Since the mathematics underpinning our method is quite involved, we will define all the symbols used in advance, in Table 1.

2.2 Energy consumption of a synchronous NSVS memory

To address the energy-aware memory clock frequency assignment, we first derive generalized energy consumption equations for a synchronous NSVS memory. The active-mode energy of the memory is primarily determined by the number of memory transactions. The total number of external memory transactions is given by

$$N_m = \epsilon N_{I1}. \quad (1)$$

If the CPU is designed to stall when there is an cache miss, we assume that the total execution time of a task is largely given by

$$\tau_{exe} \simeq \frac{N_c}{f_c} + \frac{N_m M_b}{f_m}, \quad (2)$$

without loss of generality.

Fig. 1 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 the clock frequencies of the CPU and the memory, such that

$$\sum E_{AD} = E_{AD} N_m. \quad (3)$$

We assume an auto-precharge mode [6] that immediately sends the SDRAM to idle mode after every burst-mode access. An auto-precharge scheme is common in battery-operated systems. The total precharge energy is given by

$$\sum E_{PCD} = E_{PCD} N_m. \quad (4)$$

Current-mode circuits exhibit a distinct active-mode static current but negligible leakage energy during idle mode. The active-mode static energy is directly proportional to the length of time that the circuit remain in the 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

$$\sum E_{AS} = \frac{1}{f_m} P_{AS} N_m M_b. \quad (5)$$

Simple energy models usually consider the sum of three energy components, $\sum E_{AD} + \sum E_{PCD} + \sum E_{AS}$, without distinguishing them from each other, and usually ignore the dynamic energy consumption during idle mode as well. However, a synchronous

Table 2: Energy variation versus CPU and memory clock frequencies for a given task.

Energy component	CPU clock		Memory clock	
	\uparrow	\downarrow	\uparrow	\downarrow
$\sum E_{AD}$	constant	constant	constant	constant
$\sum E_{PCD}$	constant	constant	constant	constant
$\sum E_{ID}$	\downarrow	\uparrow	\uparrow	\downarrow
$\sum E_{IS}$	\downarrow	\uparrow	constant	constant
$\sum E_{AS}$	constant	constant	\downarrow	\uparrow

memory has distinct dynamic energy consumption during idle mode, unlike an asynchronous memory, because of the necessity for clock propagation during the idle mode. SDRAM is a slave device, and thus idle mode implies ready state keeping an eye on an external request. The total idle-to-idle dynamic energy consumption is determined by the number of idle clock cycles, which is in turn determined by $\tau_{exe} - \frac{1}{f_m} N_m M_b$. Thus,

$$\sum E_{ID} = E_{ID}(\tau_{exe} f_m - N_m M_b) = \frac{E_{ID} f_m N_c}{f_c}. \quad (6)$$

This explains why an unnecessarily fast memory clock frequency increases the number of idle clock cycles and thus the total idle-to-idle dynamic energy. As the CPU clock frequency increases, the total execution time, τ_{exe} , decreases, and thus less idle-to-idle dynamic energy is consumed. On the other hand, the total idle-mode static energy is solely determined by the duration of the idle-mode:

$$\sum E_{IS} = P_{IS} \frac{1}{f_m} (\tau_{exe} f_m - N_m M_b) = \frac{P_{IS} N_c}{f_c}. \quad (7)$$

As the CPU clock frequency increases, the total idle-mode static energy decreases.

If $\tau_{exe} < \tau_d$, then both the CPU and the memory should be better to be in power-down mode to avoid unnecessary energy consumption. The energy overhead for the power down and wake up, $E_{PDD} + E_{WD}$, is proportional to the number of power-down operations. We assume that the slack time, $\tau_d - \tau_{exe}$, is located at the end of task execution, and that there is one power-down and one wake-up operation. We can then derive the dynamic energy required to enter and leave the power-down mode as follows:

$$\sum E_{PDD} = E_{PDD} \quad \text{and} \quad \sum E_{WD} = E_{WD}. \quad (8)$$

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

$$\sum E_{PDS} = P_{PDS}(\tau_d - \tau_{exe}) = P_{PDS}(\tau_d - \frac{N_c}{f_c} - \frac{N_m M_b}{f_m}). \quad (9)$$

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

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

$$\begin{aligned} E_m &= \sum E_{AD} + \sum E_{PCD} + \sum E_{AS} + \sum E_{ID} \\ &\quad + \sum E_{IS} + \sum E_{PDD} + \sum E_{WD} + \sum E_{PDS} \\ &= \frac{E_{ID} f_m N_c}{f_c} + (E_{AD} + E_{PCD}) N_m + \frac{P_{AS} N_m M_b}{f_m} + \frac{P_{IS} N_c}{f_c} \\ &\quad + E_{PDD} + E_{WD} + P_{PDS}(\tau_d - \frac{N_c}{f_c} - \frac{M_b N_m}{f_m}). \end{aligned} \quad (10)$$

Table 2 summarizes energy consumption behavior of a synchronous NSVS memory as the clock frequencies of the CPU and the memory. In some circumstances, energy consumption increases with clock frequency, while in others the relationship is inverted.

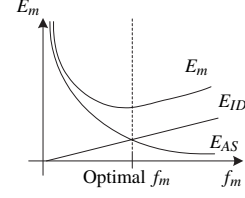


Figure 2: Energy consumption of synchronous memory against the clock frequency.

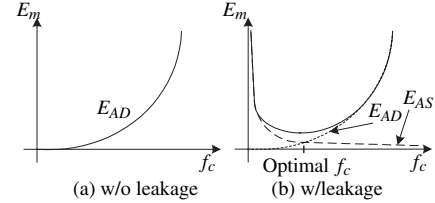


Figure 3: Energy consumption of an SVCS CPU against clock frequency.

3. ENERGY-OPTIMAL FREQUENCY ASSIGNMENT

3.1 Energy-frequency relation of a synchronous NSVS memory

In this section, we see how the synchronous NSVS memory behaves as the clock frequency changes. As we vary f_m for a given f_c , E_m responds as shown in Fig. 2. This variation is primarily caused by a trade-off between E_{ID} and E_{AS} . The total energy consumption, E_m , is convex, and thus there exists an energy-optimal clock frequency.

We will now derive the clock frequency of a synchronous NSVS memory that corresponds to the least energy consumption for a given number of CPU clock cycles, a given number of memory transactions, a given CPU clock frequency, and the parameterized energy model.

Theorem 1. *The energy consumption of a synchronous NSVS memory has a minimum, corresponding to the energy-optimal value of f_m such that*

$$f_m = \sqrt{\frac{(P_{AS} - P_{PDS}) f_c N_m M_b}{E_{ID} N_c}}. \quad (11)$$

Proof: Since E_m is convex function, we derive the optimal memory clock frequency from the derivative equation of f_m . \square

It is remarkable that the optimal f_m is proportional to root square of f_c . If CPU frequency increases four fold, we can easily achieve memory energy optimal to increase memory frequency two fold.

3.2 The SVCS CPU energy model

So far, we have examined the energy-frequency relation of a synchronous NSVS memory. This section describes the assignment of frequencies to the CPU and the memory in order to achieve system-wide energy optimum. We assume that well-designed clock gating is implemented in the CPU, and thus that the CPU consumes only negligible energy during a stall state, without loss of generality. Thus,

$$E_{cpu} = \alpha C_{cpu} k^2 f_c^2 N_c. \quad (12)$$

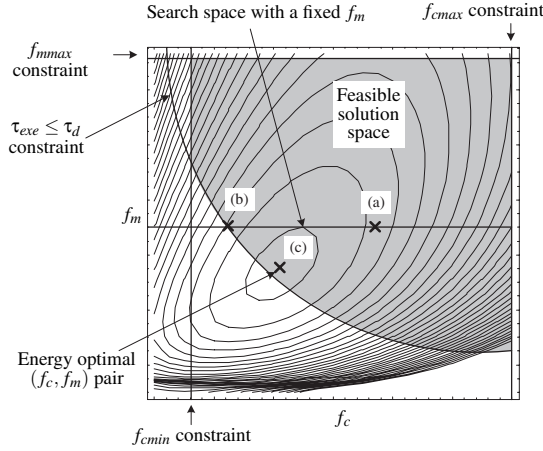


Figure 4: Feasible solution space of the energy-optimal clock frequency setting. (Contour lines denote E_m and are convex. The supply voltage of the CPU is the minimum possible values to ensure reliable operation.)

To avoid divergence of the context, we will assume that there is no leakage energy in the CPU (Fig. 3 (a)). But we can easily accommodate leakage as a function of the duration of the active state of the CPU, which makes the cost function a little more complex (Fig. 3 (b)). Allowing for leakage energy in the CPU makes the total energy, E_t , more convex, and so the optimal frequency pair (f_c, f_m) will exhibit even better energy performance compared with previous frequency assignment techniques. We will discuss E_t in Section 3.4.

3.3 Frequency assignment ignoring the memory energy

Fig. 4 summarizes three different frequency assignment schemes for the DVS of an SVCS. Fig. 4 (a) shows a conventional frequency assignment scheme for the DVS. Here, $\tau_{exe} = \frac{1}{2}\tau_d$ at $f_c = f_{cmax}$. Since the frequency assignment scheme ignores both the energy consumption and the access time of a synchronous NSVS memory, $f_c' = \frac{1}{2}f_{cmax}$ is applied over this period. However, this results in $\tau_{exe}' < \tau_d$, because only the CPU proportion of τ_{exe} , $\frac{N_c}{f_c}$, is doubled to become $2\frac{N_c}{f_c}$ [4].

This approach may not be considered to use the slack time properly, and thus a new frequency assignment is suggested, which takes into account the execution time of the memory [4], as shown in Fig. 4 (b). In this case, the frequency assignment yields $\tau_{exe}^{\#} = \tau_d$ and thus a lower value of $f_c^{\#}$, such that $f_c^{\#} < f_c'$, is feasible.

3.4 Frequency assignment considering the memory energy

The system-wide energy optimal frequency assignment is not shown in Fig. 4 (a) or in Fig. 4 (b), because both of these schemes try to minimize E_{cpu} only. In addition, $\tau_{exe} = \tau_d$ does not imply the energy-optimal frequency assignment due to the convex energy behavior of a synchronous NSVS memory [5]. As E_{cpu} and E_m are functions of f_c and f_m , and τ_{exe} is a function of f_c and f_m (Eqs. 2, 10 and 12) we can prove that there exists a system-wide energy-optimal frequency pair (f_c, f_m) . Since f_c and f_m are cross-coupled, we need to derive both f_c and f_m together, as shown in Fig. 4 (c).

The total system energy required by the CPU and the memory,

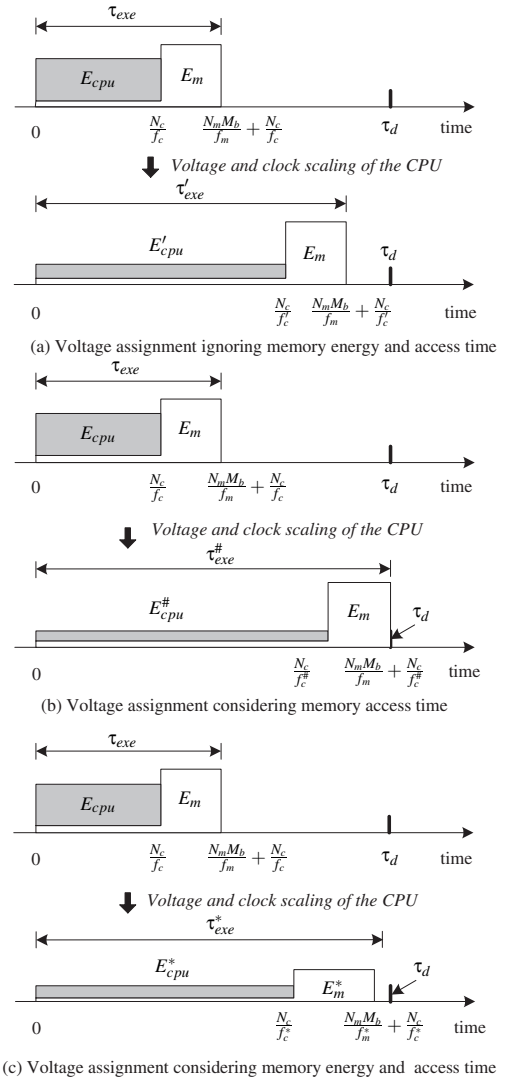


Figure 5: Frequency assignment of the CPU and the memory.

E_t , is given by

$$\begin{aligned} E_t &= E_{CPU} + E_m \\ &= \alpha C_{cpu} k^2 f_c^2 N_c + \frac{E_{ID} f_m N_c}{f_c} + (E_{AD} + E_{PCD}) N_m \\ &\quad + \frac{(P_{AS} - P_{PDS}) N_m M_b}{f_m} + \frac{(P_{IS} - P_{PDS}) N_c}{f_c} \\ &\quad + P_{PDS} \tau_d + E_{PDD} + E_{WD}. \end{aligned} \quad (13)$$

Note that, E_t is also convex because it is a linear combination of convex functions. We have convex contours of E_t over f_c and f_m , as shown in Fig. 5.

There exists a unique pair (f_c, f_m) corresponding to the minimum value of E_t . However, we also need to verify that this pair is feasible. There are constraints for a feasible (f_c, f_m) pair:

$$\begin{aligned} \text{The } f_c \text{ constraint:} & \quad f_{cmin} \leq f_c \leq f_{cmax}, \\ \text{The } f_m \text{ constraint:} & \quad f_m \leq f_{mmax}, \\ \text{The deadline constraint:} & \quad \tau_{exe} \leq \tau_d. \end{aligned} \quad (14)$$

The constraints on f_{cmax} and f_{mmax} are determined by the device data sheet. Since we only scale the clock frequency of the synchronous NSVS memory by fixing its supply voltage, there is virtually no minimum f_m . In effect, there exists a possible

minimum f_m determined by the dynamic logic structure; but is low enough to ensure the feasibility of the optimal solution. However, even an SVCS CPU has a pretty high f_{cmin} . Since we scale the supply voltage for the SVCS CPU, it is not necessary to reduce the clock frequency beyond $V_{dd} = k f_c$ as far as the total energy is concerned.

We do not consider Lagrange multiplier due to complexity though an optimization problem with constraints can be solved by Lagrange multiplier in general. Instead, we derive the solution by dividing the problem into several subproblems. The constraints must all be satisfied, and form four boundary conditions as shown in Fig. 5. Except for the deadline constraints, the boundary conditions are trivial. If the optimal (f_c, f_m) pair exists in the feasible solution space, Theorem 2 derives the optimal solution. We calculate the optimal (f_c, f_m) by solving a differential equation.

Theorem 2. *A system equipped with an SVCS CPU and a synchronous NSVS memory has a system-wide energy-optimal (f_c, f_m) pair such that*

$$4(\alpha C_{CPU} k^2)^2 N_c f_c^6 - 4\alpha C_{CPU} k^2 N_c (P_{IS} - P_{PDS}) f_c^3 - E_{ID} N_m M_b (P_{AS} - P_{PDS}) f_c + N_c (P_{IS} - P_{PDS})^2 = 0, \quad (15)$$

and

$$f_m = \sqrt{\frac{(P_{AS} - P_{PDS}) f_c N_m M_b}{E_{ID} N_c}}, \quad (16)$$

if they are included in the feasible solution space determined by Eq. 14 (Fig. 5).

Proof: The optimal (f_c, f_m) pair is derived from the simultaneous equations $\frac{\partial E_t}{\partial f_c} = 0$ and $\frac{\partial E_t}{\partial f_m} = 0$. We can derive the optimal f_m by Theorem 1, and it is given by

$$\frac{\partial E_t}{\partial f_m} = \frac{E_{ID} N_c}{f_c} - \frac{N_m M_b (P_{AS} - P_{PDS})}{f_m^3} = 0. \quad (17)$$

Replacing f_m by the solution of Eq. 17 in the first equation, we obtain

$$\frac{\partial E_t}{\partial f_c} = 2\alpha C_{CPU} f_c k^2 N_c - \sqrt{\frac{E_{ID} (P_{AS} - P_{PDS}) N_m M_b N_c}{f_c^3}} - \frac{N_c (P_{IS} - P_{PDS})}{f_c^2} = 0, \quad (18)$$

This simplifies to

$$4(\alpha C_{CPU} k^2)^2 N_c f_c^6 - 4\alpha C_{CPU} k^2 N_c (P_{IS} - P_{PDS}) f_c^3 - E_{ID} N_m M_b (P_{AS} - P_{PDS}) f_c + N_c (P_{IS} - P_{PDS})^2 = 0, \quad (19)$$

and the positive real root of Eq. 19 is the optimal f_c . ■

Since Eq. 15 and the following polynomial equations are too complex to have closed-form solutions, we use a numerical technique.

Unfortunately, we may often encounter situation that the (f_c, f_m) obtained from Theorem 2 is not included in the feasible solution space. In this case, a sub-optimal pair (f_c, f_m) can be located on the boundary of the feasible solution space. To obtain this, we need to derive local optimal (f_c, f_m) pairs on all the boundary conditions and choose the global minimum.

Corollary 1. *A system equipped with an SVCS CPU and a synchronous NSVS memory has the system-wide energy-optimal (f_c, f_m) such that*

$$f_m = \sqrt{\frac{(P_{AS} - P_{PDS}) f_{cmin} N_m M_b}{E_{ID} N_c}}, \quad (20)$$

or

$$f_m = \sqrt{\frac{(P_{AS} - P_{PDS}) f_{cmax} N_m M_b}{E_{ID} N_c}}, \quad (21)$$

if it lies on the boundary condition corresponding to f_{cmin} or f_{cmax} .

Theorem 3. *A system equipped with an SVCS CPU and a synchronous NSVS memory has the system-wide energy-optimal (f_c, f_m) such that*

$$f_c = \left(\frac{E_{ID} f_{mmax} + P_{IS} - P_{PDS}}{2\alpha C_{CPU} k^2} \right)^{\frac{1}{3}} \quad \text{and} \quad f_m = f_{mmax}, \quad (22)$$

if the optimum is included in on the boundary condition corresponding to f_{mmax} .

Proof: The optimal f_c is given by

$$\frac{\partial E_t}{\partial f_c} = N_c \left(2\alpha C_{CPU} f_c k^2 - \left(\frac{E_{ID} f_m}{f_c^2} + \frac{P_{IS} - P_{PDS}}{f_c^2} \right) \right) = 0. \quad (23)$$

Frequency assignment of the SVCS CPU in conventional DVS implies a fixed f_m . The frequency assignment schemes shown in Figs. 4 (a) and 4 (b) addresses this problem. However, neither of these schemes are system-wide energy optimal, and the energy optimal f_c is achieved by the following strategy:

$$\begin{aligned} &\text{minimize} && E_t \\ &\text{subject to} && f_{cmin} \leq f_c \leq f_{cmax}, \\ &&& f_m \text{ is a constant.} \end{aligned} \quad (24)$$

Corollary 2. Frequency assignment in conventional DVS: *A system equipped with an SVCS CPU and a synchronous NSVS memory has a system-wide energy-optimal (f_c, f_m) such that*

$$f_c = \left(\frac{E_{ID} f_m + P_{IS} - P_{PDS}}{2\alpha C_{CPU} k^2} \right)^{\frac{1}{3}}. \quad (25)$$

Proof: Since f_m is a given constant value, replacing f_{mmax} with f_m in Theorem 3, we obtain the optimal f_c for a fixed f_m . ■

Sometimes, an energy-optimal (f_c, f_m) is not feasible due to a deadline miss. In this case, a sub-optimal (f_c, f_m) is can be obtained by the following procedure:

$$\begin{aligned} &\text{minimize} && E_t \\ &\text{subject to} && f_{cmin} \leq f_c \leq f_{cmax}, \\ &&& f_m \leq f_{mmax}, \\ &&& \tau_{exe} = \tau_d. \end{aligned} \quad (26)$$

In this case, the energy-suboptimal (f_c, f_m) is on the boundary condition of the deadline constraint, $\tau_{exe} \leq \tau_d$, in the feasible solution space.

Theorem 4. *A system equipped with an SVCS CPU and a synchronous NSVS memory has a system-wide energy-optimal (f_c, f_m) such that*

$$\begin{aligned} &2\alpha C_{CPU} k^2 \tau_d^2 f_c^5 - 4\alpha C_{CPU} k^2 N_c \tau_d f_c^4 + 2\alpha C_{CPU} k^2 N_c^2 f_c^3 \\ &+ ((P_{AS} - P_{IS}) \tau_d^2 - E_{ID} N_m M_b \tau_d) f_c^2 \\ &2N_c (P_{IS} - P_{AS}) \tau_d f_c + N_c^2 (P_{AS} - P_{IS}) = 0, \end{aligned} \quad (27)$$

and

$$f_m = \frac{N_m M_b f_c}{\tau_d f_c - N_c}, \quad (28)$$

if the optimum is on the boundary condition corresponding to the inequality $\tau_{exe} \leq \tau_d$.

Proof: On the deadline boundary condition, τ_{exe} is given by

$$\tau_d = \frac{N_c}{f_c} + \frac{N_m M_b}{f_m}, \quad (29)$$

and we have

$$f_m = \frac{N_m M_b f_c}{\tau_d f_c - N_c}. \quad (30)$$

Table 3: Energy consumption of a 32-bit memory with Micron MT48LC16M8A2 SDRAMs ($M_b = 9$ and $M_m = 4$).

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

Thus, E_t is given by

$$E_t = \alpha C_{CPU} k^2 f_c^2 N_c + (E_{AD} + E_{PCD}) N_m + \frac{E_{ID} N_c N_m M_b}{\tau_d f_c - N_c} + P_{AS} \left(\tau_d - \frac{N_c}{f_c} \right) + \frac{P_{IS} N_c}{f_c}. \quad (31)$$

The optimal f_c corresponding to the solution of $\frac{\partial E_t}{\partial f_c} = 0$. ■

4. EXPERIMENTAL RESULTS

We evaluate system-wide energy differences among the frequency assignment schemes in Figs. 4 (a), 4 (b), 4 (c) and a frequency assignment without the DVS. The target system consists of a 32bit RISC processor and Micron 128Kbit SDRAMs and no other peripherals for plain benchmark. We assume that the CPU is SVCS obeying Eq. 12, and it operates at maximum 400MHz and minimum 200MHz consuming 640mW with $V_{dd} = 2V$ and 160mW with $V_{dd} = 1V$, respectively ($\alpha C_{CPU} k^2 = 1 \times 10^{-17} \text{ nJ/Hz}^2$). The CPU is assumed to have the ARM instruction set architecture and separate 8KB I-cache and D-cache. We compose 32-bit data width with four Micron SDRAMs, i.e., $M_m = 4$. The default f_m is 66MHz, which is the most popular setting for most battery-operated systems. Frequency assignment of previous DVS changes f_c while keeping $f_m = 66\text{MHz}$, and frequency assignment without the DVS implies that $f_c = f_{cmax}$ and $f_m = 66\text{MHz}$.

Table 3 summarizes energy consumption of the Micron SDRAM acquired by accurate cycle-true measurement [9]. We use Table 3 to derive energy-optimal (f_c, f_m) pairs from the theorems and the corollaries. Once we derive energy-optimal (f_c, f_m) pairs, we then perform trace-driven cycle-by-cycle energy simulation with real applications, with the derived (f_c, f_m) pairs. Thus the derived (f_c, f_m) pairs may not be true optimal due to modeling error, but this justifies that the benchmark is not biased at all.

We select three embedded applications, an MPEG4 decoder at 20 frames/sec, a JPEG de-compressor with 96% utilization and an MP3 decoder with 320Kbps streaming. Table 4 summarizes the performance of the frequency assignment techniques. Because E_m even increases over 20%, DVS with conventional frequency assignments exhibit only around 10% of the system-wide energy reduction, E_t , although they reduce E_{cpu} up to 70%. Note that the target system is equipped with only minimum numbers of SDRAMs for 32-bit data width. Our frequency assignment, however, shows additional 50% system-wide energy saving over previous frequency assignment schemes though the CPU energy consumption is higher than the conventional frequency assignments.

The optimal pair (f_c, f_m) for the MPEG decoder locates around (250MHz, 40MHz). Since the JPEG de-compressor has only 4% of slack time, DVS may not appear to be useful. In effect, DVS with conventional frequency assignments reduces f_c only by 5%. However, our scheme achieves 20% reduction of f_c by increasing f_m instead. Although the system-wide energy reduction is still minor for the JPEG de-compressor, our frequency assignment achieves 2X to 3X energy reduction over previous frequency assignments. The MP3 decoder has very low memory utilization, and thus feasible energy-optimal f_m is 7MHz. The energy-optimal f_c is f_{cmin} . We eventually have pretty much slack time even with the DVS. Our frequency assignment shows 34% additional

Table 4: Frequency assignment versus system-wide energy reduction (f_m :MHz and E_{cpu} , E_m and E_t : μJ).

MPEG4 decoder: $N_c = 7.4 \times 10^6$, $N_m = 81,208$ and $\tau_d = 47\text{ms}$

Freq. assignment	f_c	f_m	E_{cpu}	E_m	E_t	Reduction
No DVS	400	66	1,184	1,726	2,910	0%
(a)	252	66	469	2,018	2,487	14.6%
(b)	206	66	314	2,192	2,506	13.9%
(c)	246	43	448	1,902	2,350	19.3%

JPEG de-compressor: $N_c = 24.4 \times 10^6$, $N_m = 369,782$ and $\tau_d = 115\text{ms}$

Freq. assignment	f_c	f_m	E_{cpu}	E_m	E_t	Reduction
No DVS	400	66	3,904	7,123	11,027	0%
(a)	388	66	3,665	7,175	10,840	1.7%
(b)	378	66	3,484	7,218	10,702	2.9%
(c)	324	84	2,565	7,774	10,340	6.2%

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

Freq. assignment	f_c	f_m	E_{cpu}	E_m	E_t	Reduction
No DVS	400	66	320	183	503	0%
(a)	200	66	80	317	397	21.1%
(b)	200	66	80	317	397	21.1%
(c)	200	7	80	153	233	53.6%

system-wide energy reduction over previous frequency assignment schemes, for the MP3 decoder.

5. CONCLUSION AND FUTURE WORK

This paper is the first attempt to derive an energy-optimal frequency assignment of a synchronous non-supply-voltage-scalable (NSVS) memory and a supply-voltage-clock-scalable (SVCS) CPU for dynamic voltage scaling (DVS). We derive the energy optimal CPU and memory clock frequencies based on accurate energy model including active-mode leakage and idle-mode dynamic energy. We formulate analytical models of the CPU and the memory energy consumption, and derive generalized solutions for various feasibility constraints such as the minimum and the maximum clock frequencies and the deadline of a task. Our frequency assignment enhances previous DVS and results in additional 50% system-wide energy reduction over previous frequency assignment schemes.

We can improve the energy gain by adding a DPM scheme to NSCS memory derives if applicable. SDRAMs and RAMBUS DRAMs can be good candidates. We will include this complementary DPM for NSVS devices and CPU leakage power as future work.

6. REFERENCES

- [1] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," in *Proceedings of the ISLPED98*, pp. 197–202, 1998.
- [2] Thomas L. Martin and Daniel P. Siewiorek, "Nonideal battery and main memory effects on CPU speed on CPU speed-setting for low power," in *IEEE Trans. VLSI Systems*, 9(1), pp. 29–34, Feb 2001.
- [3] T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. De Micheli, "Dynamic Voltage Scaling and Power Management for Portable Systems," in *Proceedings of DAC2001*, pp. 524–529, 2001.
- [4] F. Zhang and C. S.T., "Processor voltage scheduling for real-time tasks with non-preemptible sections," in *Proceedings RTSS*, pp. 235–245, Dec 2002.
- [5] Xiaobo Fan, Carla S. Ellis and Alvin R. Lebeck, "The synergy between power-aware memory systems and processor voltage," in *Proceedings of Power-Aware Computer Systems*, 2003.
- [6] H. Shim, Y. S. Joo, Y. S. Choi, H. G. Lee and N. Chang, "Low-energy off-chip SDRAM memory systems for embedded applications," in *Trans. on Embedded Computing System*, pp 98–130, 2003.
- [7] Y.S. Joo, Y.S. Choi, H. Shim, H. G. Lee and N. Chang, "Energy exploration and reduction of SDRAM memory systems," in *Proceedings of DAC2002*, pp. 892–897, New Orleans, June, 2002.