



Optimal Voltage Allocation Techniques for Dynamically Variable Voltage Processors

WOO-CHEOL KWON

Samsung Electronics Co., Ltd.

and

TAEWHAN KIM

Seoul National University

This paper presents important, new results of a study on the problem of task scheduling and voltage allocation in dynamically variable voltage processors, the purpose of which was minimization of processor energy consumption. The contributions are twofold: (1) For given multiple *discrete* supply voltages and tasks with arbitrary arrival-time/deadline constraints, we propose a voltage allocation technique that produces a feasible task schedule with *optimal* processor energy consumption. (2) We then extend the problem to include the case in which tasks have *nonuniform loads* (i.e.; switched) capacitances and solve it *optimally*. The proposed technique, called Alloc-vt, in (1) is based on the prior results in [Yao, Demers and Shenker. 1995. In *Proceedings of IEEE Symposium on Foundations of Computer Science*. 374–382] (which is optimal for dynamically continuously variable voltages, but not for discrete ones) and [Ishihara and Yasuura. 1998. In *Proceedings of International Symposium on Low Power Electronics and Design*. 197–202] (which is optimal for a single task, but not for multiple tasks), whereas the proposed technique, called Alloc-vt_{cap}, in (2) is based on an efficient linear programming (LP) formulation. Both techniques solve the allocation problems optimally in polynomial time.

Categories and Subject Descriptors: D.4.1 [**Operating Systems**]: Process Management—*Scheduling*

General Terms: Algorithms, Design, Performance

Additional Key Words and Phrases: Dynamic voltage scaling, low power design, scheduling, variable voltage processor

1. INTRODUCTION

The progress of deep-submicron (DSM) technologies has enabled all of a system's functionalities to be implemented on a single chip (i.e., system-on-chip

This research was supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center (AITrc).

Authors' addresses: Woo-Cheol Kwon, CAE Center, Samsung Electronics Co., Ltd., San 24 Nongseo-Ri Kiheung-Eup, Yongin-City, 449-711 Korea; email: wc.kwon@samsung.com; Taewhan Kim (corresponding author) School of Electrical Engineering, Seoul National University, Kwanak P.O. Box 34, Seoul, 151-600 Korea; email: tkim@ssl.snu.ac.kr.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 1515 Broadway, New York, NY 10036 USA, fax: +1 (212) 869-0481, or permissions@acm.org.

© 2005 ACM 1539-9087/05/0200-0211 \$5.00

(SoC) design). One of the most important SoC design considerations is the minimization of the energy consumption. Most portable electronic devices with microprocessors require energy efficient mobile computing to extend the battery life. A major trend in reducing the amount of energy consumed by these devices is to utilize the concept of performance on demand [Sinha and Chanrakasan 2001] whose basic idea is to run the CPU at a voltage and frequency that satisfies the current performance requirement. That is, the devices use a low-supply voltage during periods of light workload and at the same time, to satisfy the timing constraints. This is because the amount of energy consumption, E_i , for task J_i in CMOS circuits typically increases quadratically with the supply voltage, as indicated in Chandrasena et al. [2001] and Ishihara and Yasuura [1998] (by simply assuming a fixed supply voltage for the task):

$$E_i = R_i \cdot C_i \cdot V_i^2, \quad (1)$$

where R_i is the total number of cycles required for the execution of task J_i , C_i is the average switched capacitance per clock cycle for the task, and V_i is the supply voltage used for the execution of the task.

On the other hand, it should be noted that, in addition to the voltage, the value of E_i is affected by the switched capacitance of the task (i.e., C_i in Eq. 1). The value of C_i is determined according to the execution characteristics of the task J_i : If J_i requires hardware components with high-switched capacitance, such as multipliers, for execution, the value of C_i will be large, and vice versa. Consequently, to reduce the total energy consumed by tasks, it is desirable to execute the tasks with low-switched capacitance using high-supply voltages while the tasks with high-switched capacitance using low-supply voltages.

However, the supply voltage scaling incurs one critical penalty: The voltage reduction increases circuit delay, which is approximately linearly proportional to the supply voltage, since the circuit delay, T_d , is expressed as [Chandrasena et al. 2001]:

$$T_d = \frac{C_L V_{dd}}{\mu C_{ox} (W/L) (V_{dd} - V_t)^\alpha}, \quad (2)$$

where C_L represents the total node capacitance, μ is the mobility, C_{ox} is the oxide capacitance, V_t is the threshold voltage, V_i is the supply voltage to the task, α is a constant satisfying $1 < \alpha < 2$, and W and L represent the width and length of transistors, respectively.

Consequently, the variable voltage allocation problem is, for given sets of supply voltages and tasks with possibly different switched capacitances, to schedule tasks and assign them to supply voltages so that the total energy consumption should be minimized while satisfying all the timing constraints of the tasks.

Since the supply voltage directly determines the processor's clock speed (as implied in Eq. 2), it is often convenient to think of the energy consumption as a function of the clock speed. Let $s_i(t)$ be the clock speed assigned to task J_i at time t and $P_i(s_i(t))$ be the energy consumed in task J_i during a period of unit

time, starting at t . Then, the total energy consumed by a voltage allocation, \mathcal{A}_i , for task J_i is given by [Yao et al. 1995]

$$E(\mathcal{A}_i) = \int_{t_{i,1}}^{t_{i,2}} P_i(s_i(t))dt, \quad (3)$$

where $t_{i,1}$ and $t_{i,2}$ are the start and ending times of the execution of task J_i . Thus, the total energy consumption E_{tot} for the N tasks (J_1, J_2, \dots, J_N) is

$$E_{tot} = \sum_{i=1}^N \int_{t_{i,1}}^{t_{i,2}} P_i(s_i(t))dt. \quad (4)$$

There are several works that have addressed the problem of static task scheduling and voltage allocation for low energy in variable voltage processor. Yao et al. [1995] proposed an optimal algorithm for finding a schedule of tasks and voltage allocation under the assumption that the number and range of supply voltages are infinitely large (i.e., continuously variable voltage), which is practically impossible. Hong et al. [1998] proposed a heuristic to schedule-mixed workloads of static and dynamic tasks, based on Yao et al.'s algorithm. Ishihara and Yasuura [1998] proposed an optimal voltage allocation technique using a discontinuously variable voltage processor. However, the optimality of the technique is confined to a single task. Furthermore, the optimality does not hold for the practical case in which each task has nonidentical switched capacitance. Shin and Choi [1999] proposed a fixed priority based on-line scheduling scheme for a continuously variable voltage processor. The technique is simple, but cannot fully exploit the fact that the arrival times and deadlines of most real-time tasks in embedded system applications are known in advance. In addition Shin et al. [2000] proposed an off-line fixed-priority based scheduling algorithm for a continuously variable voltage processor. One major limitation of the technique is that the arrival times of the tasks are assumed to be all identical. Quan and Hu [2001] proposed an off-line fixed-priority based voltage scheduling algorithm which overcomes the limitations in Shin and Choi [1999] and Shin et al. [2000]. For a set of tasks with statically assigned priorities, it produces a voltage schedule which always results in energy consumption below the minimum constant voltage and which shuts down the system when it is idle. However, the technique is not optimal, even if its application is useful in periodic tasks where the execution priorities among tasks are given. In Quan and Hu [2002], Quan and Hu proposed two optimal algorithms to the same voltage scheduling problem as in Quan and Hu [2001] in a fix-priority real-time system. Mochocki et al. [2002] addressed the problem of voltage scheduling with consideration of transition overhead and voltage level discretization. They heuristically modified the optimal scheduling algorithm of Yao et al. [1995] to consider the transition overhead and discrete voltages. Manzak and Chakrabarti [2003] extended the existing scheduling algorithms (earliest due date, earliest deadline first, rate monotonic) to minimize energy consumption for a continuously variable processor, and they developed heuristics by using the Lagrange multiplier method to determine the relation between the operating voltages for the minimum energy. Aydin et al. [2001] proposed an optimal polynomial time

solution to the problem of continuously variable voltage scheduling problem for periodic tasks. Unlike the Yao et al.'s solution [Yao et al. 1995] in which all tasks have identical power consumption functions, Aydin, et al.'s solution assumes that each task can have a different power consumption characteristic. Lin et al. [1997] solved a discretely variable voltage allocation problem using an ILP formulation. However, the target area is in the VLSI hardwired (i.e., static) data path scheduling, and not in the software-controlled (i.e., dynamic) CPU speed scheduling by voltages. The authors in Chang and Pedram [1997], Johnson and Roy [1997], and Raje and Sarrafzadeh [1995] also proposed low-power data path scheduling techniques with multiple voltages. Since, in the techniques in Lin et al. [1997], Chang and Pedram [1997], Johnson and Roy [1997], and Raje and Sarrafzadeh [1995], the voltage is assigned statically to each functional module, the techniques would be ineffective when the timing constraints change very dynamically according to the performance requirements of the applications. Luo and Jha [2003] addressed the problem of variable voltage static scheduling in a heterogeneous distributed embedded systems. They proposed a power-efficient variable voltage scheduling heuristic based on execution order optimization and power-profile and timing-constraint guided slack allocation. The works in Bambha et al. [2001] and in Schmitz and Al-Hashimi [2001] also addressed the problem of variable voltage scheduling for distributed systems. Recently, Shang et al. [2003] proposed a variable voltage scheduling policy to minimize the network (i.e., interconnect) power rather than the processor (i.e., computation) power. They used past network utilization to predict future traffic and tune link frequency and voltage dynamically.

In this paper, we provide a set of new research results on the problem of static task scheduling and voltage allocation in a dynamically variable voltage processor with an objective of minimizing the processor energy consumption. (A preliminary analysis of the research is found in Kwon and Kim [2003].) Specifically, the contributions are twofold: (1) for given multiple discrete supply voltages and tasks with arbitrary arrival-time/deadline constraints, we propose a voltage allocation technique which produces a feasible (task) schedule with optimal processor energy consumption; (2) We then extend the problem to include the case in which the tasks have nonuniform switched capacitances, and solve the problem optimally. The technique in (1) is based on the prior results in Yao et al. [1995] (which is optimal for continuously variable voltages, but not for discrete ones) and Ishihara and Yasuura [1998] (which is optimal for a single task, but not for multiple tasks), whereas the technique in (2) is based on an efficient linear programming (LP) formulation.

2. VARIABLE VOLTAGE ALLOCATION PROBLEM

An instance of the task scheduling and voltage allocation problem consists of a set $\mathcal{J} = \{J_1, J_2, \dots, J_N\}$ of tasks (or jobs) and a set $\mathcal{V} = \{V_1, V_2, \dots, V_M\}$ of discrete supply voltages. We denote s_i , $i = 1, 2, \dots, M$, to be the clock speed corresponding to the voltage V_i . N is the number of tasks, and M is the number of discrete voltages available for use.

Each task $J_i \in \mathcal{J}$ is associated with the following parameters:

- a_i : the arrival time of J_i .
- d_i : the deadline of J_i ($a_i \leq d_i$),
- R_i : the number of processor cycles required to complete J_i ,
- C_i : the average switched capacitance for J_i ,
- $s_i(t)$: the processor clock speed at time t , and
- $P_i(s_i(t))$: the power consumed at time t .

Note that the values of a_i , d_i , R_i , and C_i are given for task J_i , and the values of $s_i(t)$ and $P_i(s_i(t))$ vary according to the dynamic allocation of voltages to J_i , and, thus, directly affect the amount of energy consumption. We define a *feasible schedule* of tasks as a schedule in which all the timing constraints of the tasks are satisfied. We assume that tasks can be preempted. Then, the task scheduling and voltage allocation problem is

Problem 1. Given an instance of tasks and voltages, find a feasible task schedule and voltage allocation to tasks that minimizes the quantity of E_{tot} in Eq. 4.

3. OPTIMAL VARIABLE VOLTAGE ALLOCATIONS

Let us first consider a restricted case of Problem 1 in which the average switched capacitances for tasks are all identical (Section 3.1). Then, we consider cases of Problem 1 in which the switched capacitances can be any arbitrary values (Section 3.2).

3.1 Allocation with Uniform Capacitances

The problem we want to solve is

Problem 2. Problem 1 with $C_1 = C_2 = \dots = C_N$, where C_i is the average switched capacitance of task J_i and N is the number of tasks.

There are two optimal results in the literature that are related to Problem 2: (a) Ishihara and Yasuura [1998] showed that if there is only a *single task* (i.e., Problem 2 with $N = 1$), an optimal voltage allocation is to use the two voltages in \mathcal{V} that are the immediate neighbors to the (ideal) voltage the clock speed of which leads to a completion of the task exactly at the time of its deadline; (b) Yao et al. [1995] proposed an optimal allocation for Problem 2 with an infinite number and range of supply voltages (i.e., *continuously variable* voltages). Consequently, it is natural to examine the algorithmic procedures used in (a) and (b) to see if they are partially applicable to Problem 2. In fact, from the analysis of the procedures, we found that Problem 2 can be solved optimally in polynomial time by exploiting the procedures.

Before describing our voltage allocation technique, called Alloc-vt, and its optimality in detail, let us give a small example to show how our proposed procedure for Problem 2 is executed in conjunction with those in Ishihara and Yasuura [1998] and Yao et al. [1995].

Table I. An Example of Tasks with Timing Constraints

Task	Arrival Time a_i	Deadline d_i	Exec. Cycles (R_i in Millions)
J_1	0	11	150
J_2	3	8	120
J_3	5	8	180
J_4	9	11	80

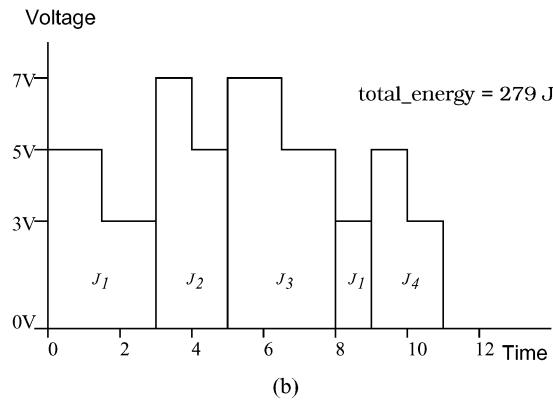
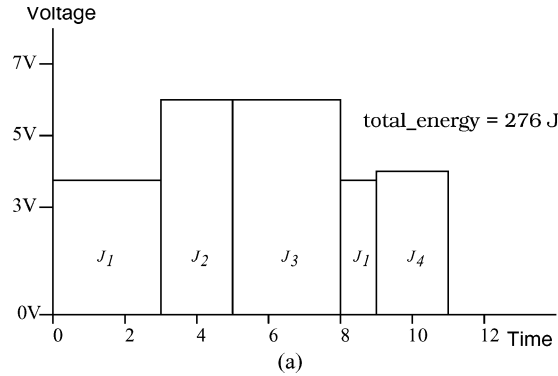


Fig. 1. An example illustrating our transformation of continuously variable voltage allocation into discontinuously variable voltage allocation. (a) A continuously variable voltage allocation for tasks in Table I; (b) A discontinuously variable voltage allocation derived from (a).

Table I shows four tasks J_1 , J_2 , J_3 , and J_4 with their timing constraints. (We assumed that the average switched capacitances of the tasks are identical, and are not shown in the table.) For simplicity, let us assume that the clock speed is linearly proportional to the supply voltage, and the energy consumption is quadratically proportional to the clock speed. Specifically, we assume that the clock speed corresponding to 5.0 V is 50 MHz, and the value of power function $P(\cdot)$ is normalized to $P(10 \text{ MHz}) = 1 \text{ J/s}$. Figure 1(a) shows an optimal voltage allocation with a feasible schedule for J_1 , J_2 , J_3 , and J_4 produced by the application of Yao et al.'s algorithm [1995]. The highest voltage used is 6.0 V

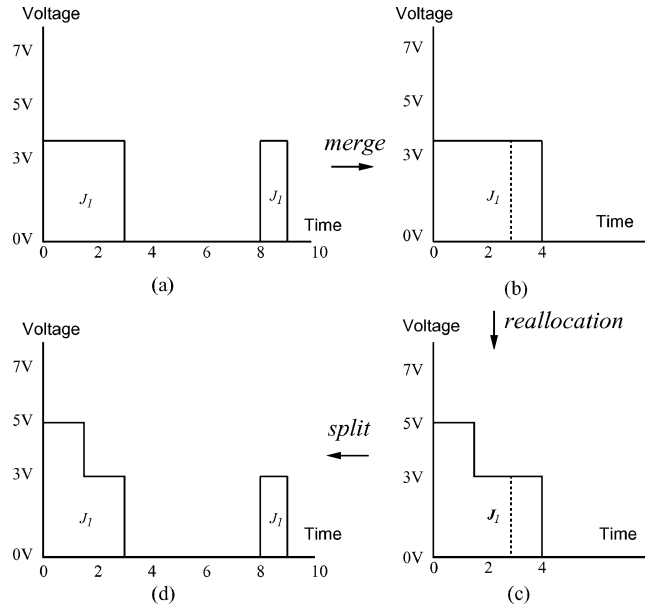


Fig. 2. The three steps of our voltage allocation procedure for task J_1 in Figure 1. (a) An initial schedule in Figure 1(a); (b) the result after merging time intervals; (c) the result after voltage reallocation; (d) the result after splitting the time interval.

and the lowest voltage is 3.75 V. The total energy consumption $E_{tot} = 276$ J. Note that task J_1 is scheduled to be executed in two separate time periods so that the deadlines of tasks J_2 and J_3 are met, while consuming minimal total (processor) energy.

Our proposed procedure starts from the results of the possibly invalid voltage allocation with the feasible task schedule obtained from Yao et al.'s algorithm [Yao et al. 1995], and transforms it into that of valid voltage allocation with a feasible schedule. The time complexity of Yao et al.'s algorithm is known to be bounded by $O(N \log^2 N)$.¹ More precisely, we will preserve the schedule of tasks during transformation, but change the voltages so that they are all valid. Then, the question is what and how the valid supply voltages are selected and used. We determine a valid voltage for each (scheduled) task by performing the following three steps: (Step 1: *Merge time intervals*) All the scheduled time intervals that were allotted to execute the task are merged into one; (Step 2: *Voltage reallocation*) The invalid supply voltage is replaced with a set of valid voltages²; (Step 3: *Split time interval*) The merged time interval is then split into the original time intervals.

For example, suppose that we have three voltages 7.0 V, 5.0 V, and 3.0 V available for use and their corresponding clock speeds are 70 MHz, 50 MHz, and 30 MHz, respectively. Then, for each scheduled task with the ideal voltage in Figure 1(a), we apply the three steps of our procedure. Figure 2 shows the

¹For details on how the algorithm works, refer to Yao et al. [1995].

²In fact, we found that at most two valid voltages are sufficient, as will be shown later.

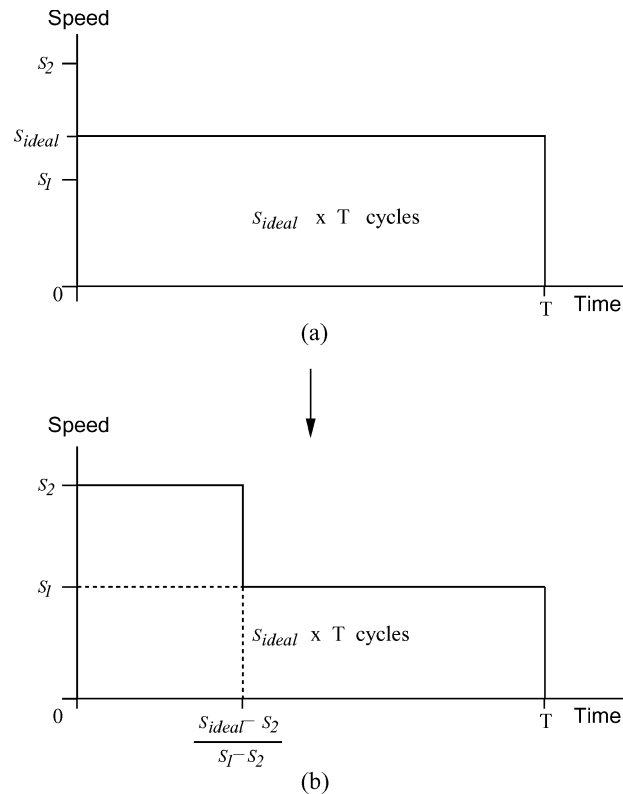


Fig. 3. Voltage reallocation (from an ideal voltage to valid voltages). (a) The ideal voltage by Yao et al. [1995]; (b) the discrete voltages by Ishihara and Yasuura [1998].

results of three steps for task J_1 . Initially, J_1 is scheduled to be executed in two time intervals $[0, 3]$ and $[8, 9]$ with the voltage being 3.75 V, as shown in Figure 2(a).³ Consequently, in Step 1 the time intervals are merged into $[0, 4]$ as shown in Figure 2(b). We then update the supply voltage in Step 2. To do this, we make use of Ishihara and Yasuura's results [1998]: For a given ideal (optimal) voltage for a task, the valid (optimal) voltage allocation is to use the two immediately neighboring valid voltages to the ideal voltage. Figure 3 shows how the ideal voltage is replaced with two immediately neighboring valid voltages where s_{ideal} represents the clock speed corresponding to the ideal voltage, and s_1 and s_2 are the clock speed corresponding to the two immediately neighboring valid voltages. (For details on how to find the time point at which the clock speed changes, see Ishihara and Yasuura [1998].) Figure 2(c) shows the result of voltage reallocation where two voltages 3.0 V and 5.0 V are used because the ideal voltage ($=3.75$ V) is in between 3.0 V and 5.0 V, and no other valid voltages are in the interval. Finally, in Step 3 we restore the time intervals while preserving the voltage reallocation obtained in Step 2, as shown in Figure 2(d). By repeating these three steps for J_2 , J_3 , and J_4 in Figure 1(a),

³According to the results in Yao et al. [1995] each task always uses the same voltage.


```

Alloc-vt: Algorithm for discretely variable voltage
         allocation and scheduling with multiple tasks
• Generate an optimal continuously variable voltage allocation
  and schedule using the method in Yao et al. [1995];
Voltage_Set = {V1, ..., VM};
if (there is a task whose ideal voltage is higher /* Corner case 2 */
    than max{V1, ..., VM})
  return (there is no feasible schedule);
endif
if (there is a task whose ideal voltage is lower /* Corner case 1 */
    than min{V1, ..., VM})
  Voltage_Set = {0.0V, V1, ..., VM};
endif
foreach (task Ji, i = 1, ..., N) {
  • Merge the time intervals used for the execution
    of Ji into one interval; /* Step 1 */
  • Reallocate voltage for the single task (Ji) /* Step 2 */
    using the method in Ishihara and Yasuura [1998] with Voltage_Set;
  • Split time interval into the original intervals; /* Step 3 */
endifor
return (the voltage allocation and schedule);

```

Fig. 4. A summary of the proposed voltage allocation procedure.

we obtain a voltage allocation for all tasks with a feasible schedule, as shown in Figure 1(b). Note that because we used only a number of discrete voltages, the energy consumption, which is 279 J, increases from that in Figure 1(a), which is 276 J. However, as it will be claimed later the amount of the increase is minimal.

Figure 4 summarizes the flow of the proposed three-step procedure, Alloc-vt. In the procedure, we should check out two corner cases in the voltage allocation result initially generated by Yao et al. [1995]: (corner case 1) When there is a task whose (ideal) voltage is lower than any of the valid voltages (i.e., lower than $\min\{V_1, \dots, V_M\}$), the two voltages used for the task are 0.0 V and $\min\{v_1, \dots, v_M\}$; (corner case 2) When there is a task whose (ideal) voltage is higher than any of the valid voltages (i.e., higher than $\max\{v_1, \dots, v_M\}$), we can safely conclude that there is no feasible schedule using voltages V_1, \dots, V_M from the fact in Yao et al. [1995] that if there is an optimal continuously variable voltage allocation with a feasible schedule in which the highest voltage used is V_h , there is no optimal continuously variable voltage allocation with a feasible schedule in which the highest voltage used is lower than V_h . The following theorem claims that Alloc-vt is optimal:

THEOREM 3.1. *Alloc-vt finds a voltage allocation with a feasible schedule, if one exists, for discretely variable voltages and multiple tasks with timing constraints, that minimizes the quantity of E_{tot} in Eq. 4.*

PROOF. Suppose that task J_k is executed for the time period of T_k in any optimal discretely variable voltage allocation. Let $\mathcal{V}_k = \{V_{k,1}, V_{k,2}, \dots\}$ be the set of voltages applied to J_k by the optimal discretely variable voltage allocation.

The clock speed corresponding to the ideal voltage is $s_{ideal} = \frac{R_k}{T_k}$. (It should be noted that an optimal energy consumption by task J_k can be achieved when the clock speed s_{ideal} is applied to J_k for the whole time period of T_k [Ishihara and Yasuura [1998] and Quan and Hu [2002]].) Consequently, if $|\mathcal{V}_k| = 1$ (i.e., $\mathcal{V}_k = \{V_{k,1}\}$) the ideal voltage V_{ideal} corresponding to s_{ideal} should be $V_{k,1}$. Otherwise, from the results in [Ishihara and Yasuura [1998]], the voltages applied for optimal energy consumption by task J_k can be only the two immediate neighbor voltages to V_{ideal} (i.e., $\mathcal{V}_k = \{V_{k,1}, V_{k,2}\}$). Thus, the two clock speeds for J_k are those immediate s_{i-1} and s_i such that $s_{i-1} < s_{ideal} < s_i$, where $s_0 = 0$. Then, the energy consumption by task J_k , when a discretely variable voltage processor is used, will increase from $P(s_{ideal}) \cdot T_k$ to $(P(s_{i-1}) \left(\frac{s_{ideal}-s_i}{s_{i-1}-s_i}\right) + P(s_i) \left(\frac{s_{ideal}-s_{i-1}}{s_i-s_{i-1}}\right)) \cdot T_k$.

We define a new power consumption function $P'(s)$:

$$P'(s) = \begin{cases} P(s_{i-1}) \left(\frac{s-s_i}{s_{i-1}-s_i}\right) + P(s_i) \left(\frac{s-s_{i-1}}{s_i-s_{i-1}}\right), & \text{if } s_{i-1} < s < s_i \\ P(s_i), & \text{if } s = s_i. \end{cases} \quad (5)$$

Let us consider a continuously variable voltage allocation in which we use the modified power consumption function P' rather than P . Then, we want to show that the discrete voltage allocation with power function $P(s)$ is identical to the continuously variable voltage allocation with power function $P'(s)$. Note that Alloc-vt with power function $P(s)$ produces exactly the same amount of power consumption as that in Yao et al. [1995] with power function $P'(s)$. The equivalence between Alloc-vt with power function $P(s)$ and Yao et al. [1995] with power function $P'(s)$ comes from the fact that the determination of the interval of each task produced by Yao et al. [1995] has nothing to do with the power function used only if the function is convex. This is because the procedure in Yao et al. [1995] inherently determines the tasks' intervals with no intervention of the power function.

Now, we show the equivalence between the discrete voltage allocation with $P(s)$ and the continuously variable voltage with $P'(s)$ in the following way: Let \mathcal{A} be an instance of optimal *discretely* variable voltage allocation with the restrictions that task $J_i, i = 1, 2, \dots, N$, should be executed for the time period of T_i , and $P(s)$ be used as a power consumption function. Let \mathcal{E} be the corresponding total amount of energy consumption. Then, we can easily check that \mathcal{A} can be a solution that is produced by an optimal *continuously* variable voltage allocation with the restriction that task $J_i, i = 1, 2, \dots, N$, should be executed for the time period of T_i , and $P'(s)$ (not $P(s)$) be used as a power consumption function. Further, the corresponding total energy consumption is exactly the same as \mathcal{E} . We can easily check that the converse argument is also hold. This implies that an optimal solution for discretely variable voltage allocation with $P(s)$ is equivalent to an optimal solution for continuously variable voltage allocation with $P'(s)$. Note that the algorithm in Yao et al. [1995] always generates an optimal solution if the power function used is convex. Thus, we need to show that $P'(s)$ is a convex function.

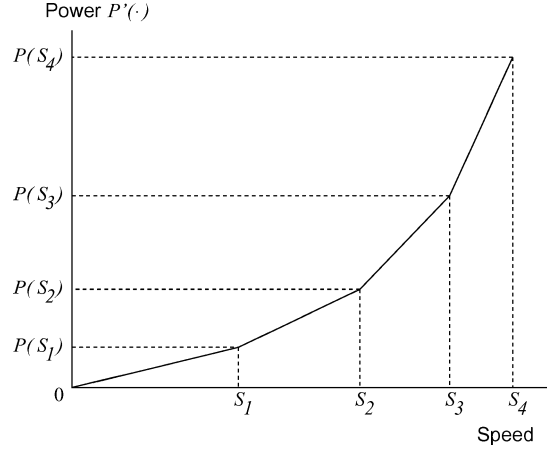


Fig. 5. The speed-power graph for power function $P'(s)$.

We restate Eq. 5 as follows:

$$P'(s) = \begin{cases} \frac{P(s_i) - P(s_{i-1})}{s_i - s_{i-1}} \cdot s + g_i, & \text{if } s_{i-1} < s < s_i \\ P(s_i), & \text{if } s = s_i \end{cases}$$

where $g_i = \frac{P(s_{i-1})s_i - P(s_i)s_{i-1}}{s_i - s_{i-1}}$. $P'(s)$ is a continuous piecewise-linear function. The slope of $P'(s)$ increases as the value of s increases, and, thus, $P'(s)$ is a convex function, as shown in Figure 5.

Consider three consecutive discrete speeds s_{i-1} , s_i , and s_{i+1} of linear subfunctions. By the definition of convexity of $P(s)$,

$$P(s_i) \leq \alpha P(s_{i-1}) + (1 - \alpha)P(s_{i+1}) \text{ for } 0 \leq \alpha \leq 1.$$

By substituting α with $\frac{s_{i+1} - s_i}{s_{i+1} - s_{i-1}}$,

$$P(s_i) \leq \frac{s_{i+1} - s_i}{s_{i+1} - s_{i-1}} P(s_{i-1}) + \frac{s_i - s_{i-1}}{s_{i+1} - s_{i-1}} P(s_{i+1})$$

or

$$\frac{P(s_i) - P(s_{i-1})}{s_i - s_{i-1}} \leq \frac{P(s_{i+1}) - P(s_i)}{s_{i+1} - s_i} \quad \square$$

3.2 Allocation with Nonuniform Capacitances

In this section, we consider Problem 1 in which the average switched capacitances of tasks are not identical. Note that the energy consumed per unit time for task J_i with clock speed s can be computed by $C_i \cdot P(s)$, where C_i is the average switched capacitances of task J_i . If the power function $P(s)$ is a simple quadratic function (e.g., $P(s) = s^2$), we can apply Yao et al.'s algorithm with a slight modification in cost function (e.g., by setting R_i , which is the required number of clock cycles for J_i , to $\sqrt{C_i} \cdot R_i$) to find an optimal *continuously variable* voltage allocation with feasible schedule, from which we can derive an

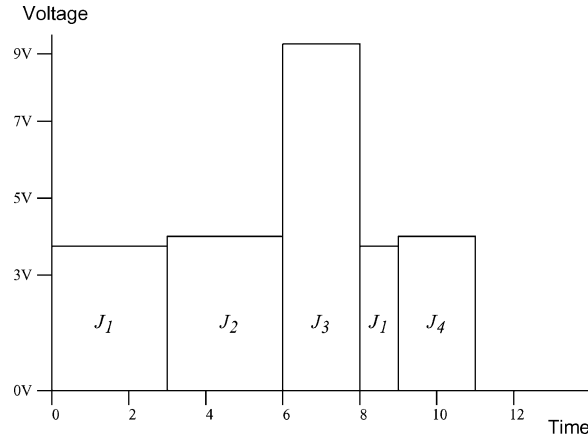


Fig. 6. The results of an optimal continuously variable voltage allocation for the tasks in Table I with $C_1 = 1.0$, $C_2 = 1.0$, $C_3 = 0.2$, and $C_4 = 1.0$. Note that task J_3 uses a relatively high voltage since its capacitance is low.

optimal discretely variable voltage allocation with feasible schedule using the procedure Alloc-vt.⁴

For example, Figure 6 shows an optimal (continuously variable) voltage allocation for the tasks in Table I when the capacitances of tasks J_1 , J_2 , J_3 , and J_4 are $C_1 = 1.0$, $C_2 = 1.0$, $C_3 = 0.2$, and $C_4 = 1.0$, respectively. Note that task J_3 uses 9.0 V which is higher than any available discrete (i.e., valid) voltages. In this case, the algorithm in Yao et al. [1995] fails in finding an optimal discretely variable voltage allocation with feasible schedule. Consequently, we propose a new voltage allocation procedure, called Alloc-vt_{cap}, to support the tasks with non-uniform switched capacitances. Specifically, we formulate the discretely variable voltage allocation problem into a linear programming (LP) problem and solve it optimally (in polynomial time).⁵

Let us first clarify notations and variables used in the LP formulation:

- N : the number of tasks,
- M : the number of discrete voltages,
- $a_k, d_k, R_k, C_k, s_j, P(s_j)$: defined at the beginning of Section 2.

For a given set of N tasks J_1, J_2, \dots, J_N , we sort the tasks' arrival times and deadlines together in nondecreasing order. Let t_1, t_2, \dots, t_{2N} denote the ordered time sequence.

- x_{jk}^i : the total time spent at executing task J_k with supply voltage V_j during the time interval $[t_i, t_{i+1}]$.

⁴The processor clock speed for each task is scaled to a factor of $\frac{1}{\sqrt{C_i}}$.

⁵There are well-known polynomial time algorithms for LP, such as ellipsoid method and interior point method [Bertsimas and Tsitsiklis 1997].

Then, the LP formulation⁶ to find an optimal discretely variable voltage allocation is

$$\text{Minimize } \sum_{i=1}^{2N-1} \sum_{k=1}^N \sum_{j=1}^M C_k \cdot P(s_j) \cdot x_{jk}^i \quad (6)$$

subject to

$$\sum_{k=1}^N \sum_{j=1}^M x_{jk}^i \leq t_{i+1} - t_i, \quad i = 1, \dots, 2N - 1 \quad (7)$$

$$x_{jk}^i = 0, \quad \text{if } a_k \geq t_{i+1} \text{ or } d_k \leq t_i \quad (8)$$

$$\sum_{i=1}^{2N-1} \sum_{j=1}^M s_j \cdot x_{jk}^i \geq R_k, \quad k = 1, \dots, N \quad (9)$$

$$0 \leq x_{jk}^i, \quad \forall i, j, k \quad (10)$$

Constraint (7) ensures that the total time consumed by all tasks at time interval $[t_i, t_{i+1}]$ should not exceed the time interval. Constraint (8) indicates that each task should be executed only within the interval of its arrival time and deadline. Constraint (9) ensures the voltage allocation and time schedule for each task must satisfy the given total cycle constraint for the task. Constraint (10) follows from the definition of variable x_{jk}^i .

The following is a segment produced by our LP formulation Alloc-vt_{cap} for the tasks with timing constraints in Figure 1 and nonuniform switched capacitances used in Figure 6.⁷ For example, the second inequality indicates that during the second time interval (i.e., $[3, 5]$) tasks J_1 and J_2 can be scheduled for execution. Thus, the total time spent by the tasks with various voltages should not be greater than 2 ($= 5 - 3$). The last inequality indicates that the total sum of the number of clock cycles at voltage V_1 for task J_4 for the entire time period $[0, 11]$ (i.e., $\sum_{i=1}^5 30 \cdot x_{14}^i$), the number of clock cycles at voltage V_2 (i.e., $\sum_{i=1}^5 50 \cdot x_{24}^i$), and the number at V_3 (i.e., $\sum_{i=1}^5 70 \cdot x_{34}^i$) should not be less than 80, which is the total number of clock cycles required for J_4 in Table I.

$$\begin{aligned} x_{11}^1 + x_{21}^1 + x_{31}^1 &\leq 3 - 0 \\ x_{11}^2 + x_{21}^2 + x_{31}^2 + x_{12}^2 + x_{22}^2 + x_{32}^2 &\leq 5 - 3 \\ &\dots \\ x_{11}^4 + x_{21}^4 + x_{31}^4 + x_{13}^4 + x_{23}^4 + x_{33}^4 &\leq 9 - 8 \\ x_{11}^5 + x_{21}^5 + x_{31}^5 + x_{14}^5 + x_{24}^5 + x_{34}^5 &\leq 11 - 9 \end{aligned}$$

⁶In a strict sense, the proposed formulation is not an LP formulation because the values of $x_{j,k}^i$ (in term of execution cycles) are not continuous. However, in practice we can consider it to be an LP formulation because the execution cycles specified by $x_{j,k}^i$ are significantly large, meaning that the rounding is sufficiently tolerable.

⁷We assume that the clock speeds at the three different voltages are 30, 50, and 70, respectively.

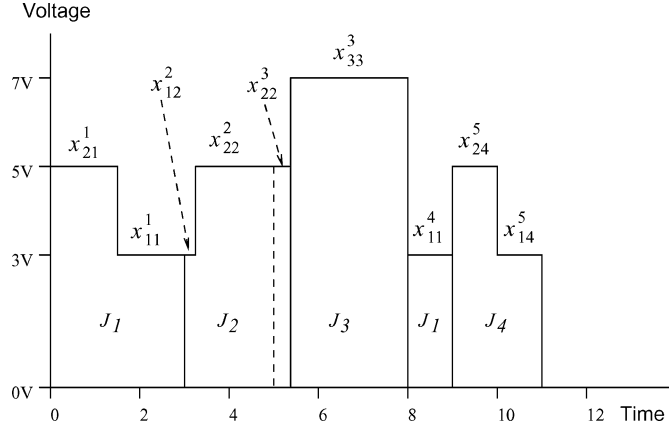


Fig. 7. The results of an optimal discretely variable voltage allocation corresponding to our LP solution by Alloc-vt_{cap} for the example in Figure 6.

$$\begin{aligned} \sum_{i=1}^5 (30 \cdot x_{11}^i + 50 \cdot x_{21}^i + 70 \cdot x_{31}^i) &\geq 150 \\ \sum_{i=1}^5 (30 \cdot x_{12}^i + 50 \cdot x_{22}^i + 70 \cdot x_{32}^i) &\geq 120 \\ &\dots \\ \sum_{i=1}^5 (30 \cdot x_{14}^i + 50 \cdot x_{24}^i + 70 \cdot x_{34}^i) &\geq 80 \end{aligned}$$

By solving the formulation, we obtain

$$\begin{aligned} x_{11}^1 &= 1.5, & x_{21}^1 &= 1.5, \\ x_{12}^2 &= 0.07, & x_{22}^2 &= 1.93, \\ x_{22}^3 &= 0.43, & x_{33}^3 &= 2.57, \\ x_{11}^4 &= 1, & x_{14}^5 &= 1, & x_{24}^5 &= 1, \text{ and} \\ x_{jk}^i &= 0 \text{ for the rest.} \end{aligned}$$

Specifically, $x_{11}^1 = 1.5$ means that task J_1 spends 1.5 units of time in the first interval $[0, 3]$ with voltage $V_1 (=3.0 \text{ V})$, and $x_{21}^1 = 1.5$ means that task J_1 spends 1.5 units of time in the first interval $[3, 5]$ with voltage $V_2 (=5.0 \text{ V})$ and so on. Figure 7 shows the voltage allocation and task schedule corresponding to the LP solution.

4. EXPERIMENTAL RESULTS

Our proposed algorithms were implemented in C++ and executed on an Intel Pentium IV computer. (The linear programming in Alloc-vt_{cap} was solved using ILOG CPLEX 7.0 [ilo].) In the experiments, we used virtual discretely variable voltage processors, and applied our techniques to a set of randomly generated tasks of moderate size. Our experiments are carried out in two respects:

Table II. A Set of Processors with a Number of Discretely Variable Clock Speeds

Processor	Available Speeds (MHz)
\mathcal{P}_1	{300, 700}
\mathcal{P}_2	{300, 500, 700}
\mathcal{P}_3	{300, 400, 500, 600, 700}
\mathcal{P}_4	{300, 333, 367, 400, 433, 467, 500, 533, 567, 600, 633, 667, 700}

Table III. Specification of Sample Task Sets

Task Set	(arr.time, deadline, exe.cycles (100M), cap (μ F))
\mathcal{J}_1	(8, 87, 144, 2) (33, 67, 40, 3) (51, 55, 8, 1) (63, 79, 32, 3) (69, 168, 117, 4) (72, 103, 56, 2) (86, 150, 123, 2) (102, 197, 183, 4) (103, 122, 18, 3) (114, 155, 52, 4)
\mathcal{J}_2	(8, 87, 139, 2) (33, 67, 38, 3) (51, 55, 7, 1) (55, 96, 3800, 2) (63, 79, 31, 3) (69, 168, 113, 4) (72, 103, 54, 2) (82, 99, 30, 1) (86, 150, 120, 2) (88, 192, 119, 1) (96, 172, 79, 3) (102, 197, 178, 4) (103, 122, 17, 3) (107, 152, 84, 2) (114, 155, 51, 4)
\mathcal{J}_3	(0, 17, 11, 2) (2, 47, 33, 2) (3, 129, 121, 1) (8, 109, 67, 4) (16, 127, 131, 2) (24, 46, 32, 2) (35, 104, 97, 1) (40, 81, 48, 3) (45, 122, 108, 3) (58, 149, 80, 2) (70, 181, 115, 4) (85, 164, 93, 3) (91, 99, 10, 4) (99, 123, 24, 2) (108, 233, 129, 1) (121, 149, 33, 2) (126, 242, 171, 4) (128, 154, 32, 3) (134, 186, 69, 3) (138, 281, 158, 1)
\mathcal{J}_4	(0, 23, 12, 2) (2, 62, 36, 2) (4, 172, 132, 1) (11, 146, 73, 4) (14, 99, 103, 3) (21, 169, 143, 2) (33, 62, 35, 2) (35, 73, 41, 2) (45, 75, 25, 4) (47, 139, 105, 1) (54, 109, 53, 3) (60, 163, 118, 3) (78, 200, 88, 2) (93, 241, 125, 4) (100, 239, 84, 4) (100, 131, 31, 2) (114, 220, 102, 3) (121, 132, 12, 4) (121, 268, 133, 2) (131, 164, 28, 1) (132, 165, 28, 2) (145, 312, 141, 1) (161, 199, 36, 2) (168, 323, 187, 4) (171, 205, 35, 3) (175, 205, 32, 3) (179, 249, 76, 3) (181, 209, 28, 2) (185, 376, 173, 1) (197, 300, 112, 2)

(1) to check the effectiveness of Alloc-vt for tasks with uniform switched capacitances and (2) to check the effectiveness of Alloc-vt_{cap} for tasks with nonuniform capacitances. Note that since both Alloc-vt and Alloc-vt_{cap} are optimal in terms of energy consumption, the comparisons (in the tables below) are reference only to show how much the energy consumptions can be reduced further if our techniques are used. We assumed that power function P is a simple quadratic function, and is normalized to $P(100 \text{ MHz}) = 1 \text{ J/s}$ at which the average switched capacitance is $1 \mu\text{F}$.

Voltage Allocation for Tasks with Uniform Switched Capacitances. Table II shows a set of processors, each of which has a number of discretely variable clock speeds, controlled by voltage. Table III specifies a set of task sets that were tested in our experiments using the processors in Table II. Table IV shows the comparisons of the energy consumptions for the task sets in Table III, in which the switched capacitances of tasks are all identical, produced by an optimal continuously variable voltage allocation Yao et al. [1995] with a subsequent *greedy* discrete voltage reallocation, and Alloc-vt. Here, the term *greedy* indicates the reassignment of the ideal voltages obtained by the technique in Yao et al. [1995] to the immediately higher valid voltages. On the other hand, Figure 8 shows

Table IV. Energy Consumptions for Tasks with Uniform Capacitances Produced by an Optimal Continuously Variable Voltage Allocation by Yao et al. [1995] with Greedy Discrete Voltage Reallocation, and Alloc-vt

Task Set	Processor	Energy Consumption (unit: J)		Reduction(%)
		Yao et al. [1995] + <i>greedy</i>	Alloc-vt	
\mathcal{J}_1	\mathcal{P}_1	54.1	37.6	30.5
	\mathcal{P}_2	38.6	33.4	13.5
	\mathcal{P}_3	36.7	32.3	12.0
	\mathcal{P}_4	32.2	31.9	1.9
\mathcal{J}_2	\mathcal{P}_1	76.8	70.1	8.8
	\mathcal{P}_2	72.4	67.7	6.5
	\mathcal{P}_3	70.2	66.7	5.9
	\mathcal{P}_4	67.2	66.4	1.2
\mathcal{J}_3	\mathcal{P}_1	109.3	97.1	11.3
	\mathcal{P}_2	106.1	90.5	14.8
	\mathcal{P}_3	92.1	88.2	4.3
	\mathcal{P}_4	90.0	88.0	2.3
\mathcal{J}_4	\mathcal{P}_1	162.8	153.7	5.6
	\mathcal{P}_2	159.4	151.3	5.1
	\mathcal{P}_3	157.5	150.1	4.0
	\mathcal{P}_4	156.4	149.3	4.6
Average				8.3

$(\{ \text{Alloc-vt} \} / [\text{Yao et al. 1995}]) \times 100$

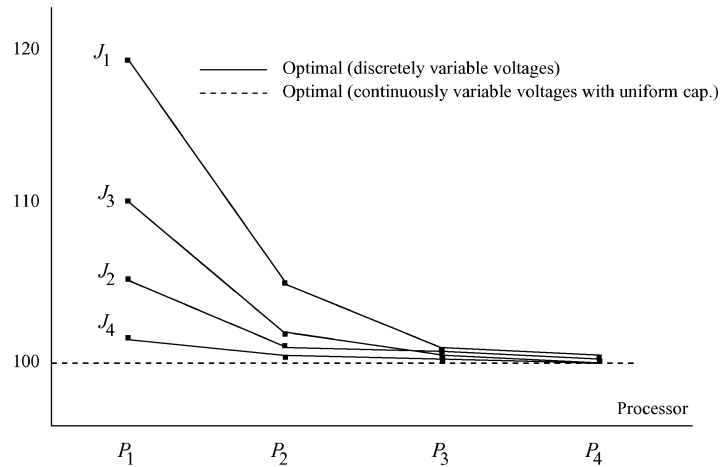


Fig. 8. Curves showing the difference between optimal discretely variable and optimal continuously variable voltage allocations under uniform capacitances of tasks.

how much the optimal results using discretely variable voltages by Alloc-vt differ from the optimal results (i.e., the dashed line in the figure) obtained by using continuously variable voltages from Yao et al. [1995]. The curves show that as the number of available voltages increases, the allocation solutions approach to those of Yao et al. [1995].

Voltage Allocation for Tasks with Nonuniform Switched Capacitances. Table V shows the comparisons of the energy consumptions for the task sets in

Table V. Energy Consumptions for Tasks with Nonuniform Capacitances Produced by an Optimal Continuously Variable Voltage Allocation by Yao et al. [1995] with Greedy Discrete Voltage Reallocation, and Alloc- vt_{cap}

Task Set	Processor	Energy Consumption (unit: J)		Reduction (%)
		Yao et al. [1995] + <i>Greedy</i>	Alloc- vt_{cap}	
\mathcal{J}_1	\mathcal{P}_1	163.2	107.5	34.2
	\mathcal{P}_2	116.6	100.1	14.2
	\mathcal{P}_3	112.4	96.1	14.5
	\mathcal{P}_4	98.2	95.8	2.5
\mathcal{J}_2	\mathcal{P}_1	202.2	183.8	9.1
	\mathcal{P}_2	192.7	176.9	8.2
	\mathcal{P}_3	187.8	174.2	7.2
	\mathcal{P}_4	179.5	173.9	3.1
\mathcal{J}_3	\mathcal{P}_1	258.6	220.5	14.7
	\mathcal{P}_2	255.5	205.3	19.6
	\mathcal{P}_3	220.1	203.8	7.4
	\mathcal{P}_4	216.3	202.8	6.2
\mathcal{J}_4	\mathcal{P}_1	392.4	373.8	4.7
	\mathcal{P}_2	389.0	365.0	6.2
	\mathcal{P}_3	387.0	361.9	6.5
	\mathcal{P}_4	385.8	361.4	6.3
Average				10.3

$$(\{ \text{Alloc-}vt_{cap} \} / [\text{Yao et al. 1995}]) \times 100$$

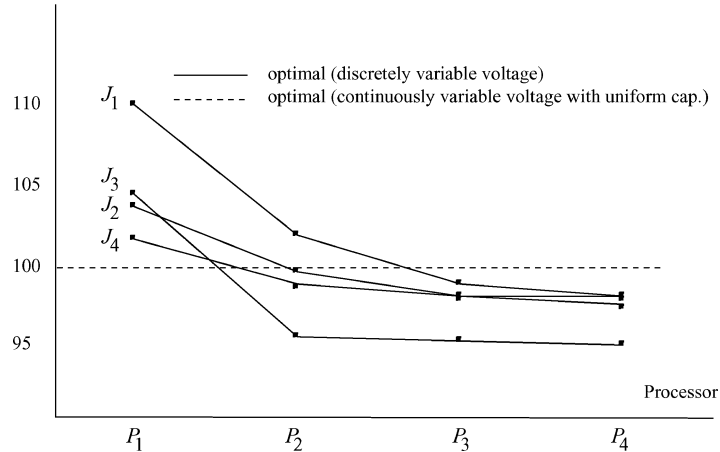


Fig. 9. Curves showing the difference between optimal discretely variable and optimal continuously variable voltage allocations under nonuniform capacitances of tasks.

Table III produced by an optimal continuously variable voltage allocation Yao et al. [1995] with a subsequent *greedy* discrete voltage reallocation, and Alloc- vt_{cap} . Still, the results produced by Alloc- vt_{cap} is optimal, and consumes about 10% less energy than Yao et al. [1995] with greedy voltage reassignment. Figure 9 shows how much the optimal results using discretely variable voltages by Alloc- vt_{cap} differ from the results (i.e., the dashed line in the figure) obtained by Yao et al. [1995]. The design points below the dashed line indicate that Alloc- vt_{cap} outperforms Yao et al. [1995]. This is mainly due to

the fact that Yao et al. [1995] is not optimal for continuously variable voltage allocation when nonuniform capacitances are assumed, while Alloc- vt_{cap} is optimal.

It should be noted that as indicated by Sinha and Chandrakasan [2001] in their experiments, the values of averaged switched capacitances of tasks in many programs have little variations. Consequently, our optimal voltage allocation technique for tasks with nonuniform switched capacitances would not be much effective in most embedded applications in practice. Nevertheless, in case there are applications with nonuniform switched capacitances and energy consumption is critical, our technique, which is neat and simple, will be useful to find a theoretically optimal voltage allocation and task schedule.

5. CONCLUSIONS

We studied a set of optimization problems which have not been addressed extensively in the literature although their importance is growing in the area of low-energy embedded system design. We addressed, in this paper, the problem of static task scheduling and voltage allocation in dynamically variable voltage processors for minimization of the total processor energy consumption. Specifically, the main contributions are (1) for given multiple voltages and tasks with arrival-time/deadline constraints, we propose an optimal voltage allocation technique; (2) We then extend the problem to include the more realistic situation in which tasks have nonuniform load capacitances, and solve this problem optimally.

Note that the contribution of our work in the field of dynamically variable voltage processors is to show that the problem of offline task and voltage scheduling in a discretely variable voltage processor, which has been tackled heuristically in the literature, is *solvable optimally in polynomial time* for hard real-time tasks either with uniform or nonuniform load capacitances. However, the overhead incurred by voltage transition is ignored in the work. Generally, there are three types of transition overhead encountered for a voltage transition: *transition cycle*, *transition interval*, and *transition energy*. A *transition cycle* is the number of additional cycles for the transition itself, which will be a constant (denoted as Δcyc). A *transition interval* is the time taken during voltage transition, which is also a constant (denoted as $\Delta time$) [Mochocki et al. 2002] in most variable voltage processors. Since it has been known that our optimal voltage allocation uses at most two voltage levels at each task, our technique can be used by assuming that the number of processor cycles required to execute task J_i ($j = 1, \dots$) is set to $R_i + 2\Delta cyc$ and the deadline of task J_i is set to $d_i - 2\Delta time$. Even though this assumption destroys the optimality, the resulting allocations will be very close to the optimum, taking a tight and predictable error bound. Finally, *transition energy* is the amount of energy consumed during the transition interval. The value of transition energy may vary depending on the starting and ending voltage levels in transition. Thus, the problem induced by the transition energy is much more difficult to solve optimally or even near-optimally. This issue should be investigated in the future.

REFERENCES

- <http://www.ilog.com>. ILOG, Inc.
- AYDIN, H., MELHEM, R., MOSSE, D., AND MEJIA-ALVAREZ, P. 2001. Determining optimal processor speeds for periodic real-time tasks with different power characteristics. In *Proceedings of Euromicro Conference on Real-Time Systems*. 225–232.
- BAMBHA, N., BHATTACHARYA, S. S., TELCH, J., AND ZITZLER, E. 2001. Hybrid search strategies for dynamically voltage scaling in embedded multiprocessors. In *Proceedings of International Workshop on Hardware/Software Co-Design*. 243–248.
- BERTSIMAS, D. AND TSITSIKLIS, J. N. 1997. *Introduction to Linear Programming*. Athena Scientific Publisher, Chs. 8 and 9.
- CHANDRASENA, L. H., CHANDRASENA, P., AND LIEBELT, M. J. 2001. An energy efficient rate selection algorithm for voltage quantized dynamic voltage scaling. In *Proceedings of International Symposium on System Synthesis*. 124–129.
- CHANG, J.-M. AND PEDRAM, M. 1997. Energy minimization using multiple supply voltages. *IEEE Trans. VLSI Syst.* 5, 4 (Dec.), 436–443.
- HONG, I., KIROVSKI, D., QU, G., POTKONJAK, M., AND SRIVASTAVAZ, M. B. 1998. Vpower optimization of variable voltage core-based systems. In *Proceedings of Design Automation Conference*. 176–181.
- ISHIHARA, T. AND YASUURA, H. 1998. Voltage scheduling problem for dynamically variable voltage processors. In *Proceedings of International Symposium on Low Power Electronics and Design*. 197–202.
- JOHNSON, M. C. AND ROY, K. 1997. Datapath scheduling with multiple supply voltages and level converters. *ACM Trans. Design Automation Electronic Syst.* 2, 3 (July), 227–248.
- KWON, W. AND KIM, T. 2003. Optimal voltage allocation techniques for dynamically variable voltage processors. In *Proceedings of Design Automation Conference*. 125–130.
- LIN, Y.-R., HWANG, C.-T., AND WU, A. C. 1997. Scheduling techniques for variable voltage low power designs. *ACM Trans. Design Automation of Electronic Syst.* 2, 2 (Apr.), 81–97.
- LUO, J. AND JHA, N. K. 2003. Power-profile driven variable voltage scaling for heterogeneous distributed real-time embedded systems. In *Proceedings of International Conference on VLSI Design*. 369–375.
- MANZAK, A. AND CHAKRABARTI, C. 2003. Variable voltage task scheduling algorithms for minimizing energy/power. *IEEE Trans. VLSI Syst.* 11, 2 (Apr.), 270–276.
- MOCHOCKI, B., HU, X. S., AND QUAN, G. 2002. A realistic variable voltage scheduling model for real-time applications. In *Proceedings of International Conference on Computer-Aided Design*. 726–731.
- QUAN, G. AND HU, X. S. 2001. Energy efficient fixed-priority scheduling for real-time systems on variable voltage processors. In *Proceedings of Design Automation Conference*. 828–833.
- QUAN, G. AND HU, X. S. 2002. Minimum energy fixed-priority scheduling for variable voltage processors. In *Proceedings of Design Automation and Test in Europe Conference and Exhibition*. 782–787.
- RAJE, S. AND SARRAFZADEH, M. 1995. Variable voltage scheduling. In *Proceedings of International Symposium on Low Power Electronics and Design*. 9–14.
- SCHMITZ, M. T. AND AL-HASHIMI, B. M. 2001. Considering power variables of dvs processing elements for energy minimization in distributed systems. In *Proceedings of International Symposium on System Synthesis*. 250–255.
- SHANG, L., PEH, L.-S., AND JHA, N. K. 2003. Dynamic voltage scaling with links for power optimization of interconnection networks. In *Proceedings of International Symposium on High-Performance Computer Architecture*. 91–102.
- SHIN, Y. AND CHOI, K. 1999. Power conscious fixed priority scheduling for hard real-time systems. In *Proceedings of Design Automation Conference*. 134–139.
- SHIN, Y., CHOI, K., AND SAKURAI, T. 2000. Power optimization of real-time embedded systems on variable speed processors. In *Proceedings of International Conference on Computer-Aided Design*. 365–368.
- SINHA, A. AND CHANDRAKASAN, A. P. 2001. Jouletrack—A web based tool for software energy profiling. In *Proceedings of Design Automation Conference*. 220–225.

- SINHA, A. AND CHANRAKASAN, A. P. 2001. Energy efficient real-time scheduling. In *Proceedings of International Conference on Computer-Aided Design*. 458–463.
- YAO, F., DEMERS, A., AND SHENKER, S. 1995. A scheduling model for reduced cpu energy. In *Proceedings of IEEE Symposium on Foundations of Computer Science*. 374–382.

Received March 2003; revised August 2003, December 2003; accepted June 2004