Optimal Voltage Allocation Techniques for Dynamically Variable Voltage Processors

Woo-Cheol Kwon
CAE Center
Samsung Electronics Co.,Ltd.
San 24, Nongseo-Ri, Kiheung-Eup, Yongin-City,
Kyounggi-Do, Korea

Taewhan Kim
Dept. of EECS and Advanced Information
Technology Research Center(AITrc)
Korea Advanced Institute of Science &
Technology, Korea

ABSTRACT

This paper presents a set of new important results for the problem of task scheduling and voltage allocation in dynamically variable voltage processor for minimizing the total processor energy consumption. The contributions are two folds: (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 tasks have *non-uniform load* (i.e., switched) capacitances, and solve it *optimally.*

Categories and Subject Descriptors

B.7.1 [**INTEGRATED CIRCUITS**]: Types and Design Styles

General Terms

Algorithms, Design, Performance

Keywords

low power design, variable voltage processor, scheduling

1. INTRODUCTION

The progress of deep-submicron (DSM) technologies has enabled a system's entire functionalities to be implemented in a single chip (i.e., system-on-chip (SoC) design). One of the most important SoC design considerations is to minimize the energy consumption. Most portable electronic devices with micro-processor require energy efficient mobile computing to extend the battery life. A major trend in saving energy consumption on the devices is to utilize the concept of performance on demand [1]. The idea of energy saving is to use a low supply voltage during the periods of light workload and at the same time, to satisfy the timing constraints. This is because of the fact that the amount of energy consumption, E_i, for task J_i in CMOS circuits typically increases quadratically with the supply voltage, as indicated in [2, 3] (by simply assuming a fixed supply voltage for the task):

$$E_i = R_i \cdot C_i \cdot V_i^2 \tag{1}$$

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 voltage supplied to 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 the hardware components with high switched capacitance, such as multiplier, for execution, the value of C_i will be large, and conversely. Consequently, to reduce the total energy consumed by tasks, it is desirable to execute the tasks with low switched capacitance by using high supply voltages while the tasks with high switched capacitance by using low 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, from the fact that the circuit delay, T_d, is expressed as ([2]):

$$T_d = \frac{C_L V_{dd}}{\mu C_{ox}(W/L)(V_{dd} - V_t)^2} \approx 1/V_{dd} \tag{2}$$

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, 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 (or power) consumed in task 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 ([5])

DAC 2003, June 2–6, 2003, Anaheim, California, USA.
Copyright 2003 ACM 1-58113-688-9/03/0006 ...$5.00.

$$E(\mathcal{A}_i) = \int_{t_1}^{t_2} P_i(s_i(t))dt \quad (3)$$

where t_1 and t_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 $(J_1, J_2, \cdots, J_N)$ is

$$E_{tot} = \sum_{i=1}^{N} \int_{t_1}^{t_2} P_i(s_i(t))dt. \quad (4)$$

There are several works which addressed the problem of static task schedule and voltage allocation for low power in variable voltage processor. Yao *et al.* [5] 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.* [4] proposed a heuristic to schedule mixed workloads of static and dynamic tasks, based on the Yao *et al.*'s algorithm. Ishihara and Yasuura [3] proposed an optimal voltage allocation technique using discontinuously variable voltage processor. However, the optimality of the technique is confined to a single task. Further, the optimality does not hold for the practical case in which each task has non-identical switched capacitance. Lin *et al.* [6] 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 [7, 8, 9] also proposed low-power data path scheduling techniques with multiple voltages. Since for the techniques in [6, 7, 8, 9] 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. Recently, Mochocki *et al.* [10] proposed a heuristic voltage scheduling algorithm which takes into account the transition overhead and voltage level discretization.

In this paper, we provide a set of new important research results on the problem of task scheduling and voltage allocation in dynamically variable voltage processor for minimizing the processor energy consumption. Specifically, the contributions are two folds: (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 tasks have *non-uniform switched capacitances*, and solve it *optimally*. The technique in (1) is based on the prior results in [5] (which is optimal for continuously variable voltages, but not for discrete ones) and [3] (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 task scheduling and voltage allocation problem consists of a set $\mathcal{J} = \{J_1, J_2, \cdots, J_N\}$ of tasks (or jobs) and a set $\mathcal{V} = \{V_1, V_2, \cdots, V_M\}$ of discrete supply voltages. We assume that $(V_1, V_2, \cdots, V_M)$ is a non-decreasing sequence. We denote s_i, $i = 1, 2, \cdots, M$, 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 to 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 to be 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 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 $C_1 = C_2 = \cdots = C_N$ 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 [3] 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 in which its clock speed leads to a completion of the task exactly at the time of its deadline; (b) Yao *et al.* [5] 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 [3] and [5].

Table 1 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

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

Table 1: An example of tasks with timing constraints.

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 5.0V is 50MHZ, and the value of power function $P(\cdot)$ is normalized to $P(10\text{MHZ}) = 1J/\text{sec}$. Figure 1(a) shows an optimal voltage allocation with feasible schedule for J_1, J_2, J_3 and J_4 produced by the application of Yao *et al.*'s algorithm [5]. The highest voltage used is 6.0V and the lowest voltage is 3.75V. The total energy consumption $E_{tot} = 276J$. 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 to be met, while consuming minimal total (processor) energy.

Our proposed procedure starts from the result of the possibly invalid voltage allocation with feasible task schedule obtained from the Yao *et al.*'s algorithm [5], and transforms it into that of valid voltage allocation with feasible schedule. The time complexity of the 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 voltages[2]; (Step 3: *Split time interval*) The merged time interval is then split to be the original ones.

For example, suppose that we have three voltages 7.0V, 5.0V and 3.0V available to use and their corresponding clock speeds are 70MHZ, 50MHZ and 30MHZ, 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 results of three steps for task J_1. Initially, J_1 is scheduled to be executed in two time interval [0,3] and [8,9] with voltage being 3.75V, 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 [3]: For a given ideal (optimal) voltage for a task, the valid (optimal) voltage allocation is to use the two immediate valid voltages to the ideal voltage. Figure 3 shows how the ideal voltage is replaced with two immediate 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 immediate valid voltages. (For details on how to find the time point at which the clock speed changes, see [3].) Figure 2(c) shows the result of voltage reallocation where two voltages 3.0V and 5.0V are used because the ideal voltage (=3.75V) is in between 3.0V and 5.0V, and no other valid voltage is 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 this three steps for each task of J_2, J_3 and J_4 in Figure 1(a), we obtain a voltage allocation for all tasks with feasible schedule, as shown in Figure 1(b). Note that because we used only a number of discrete voltages, the energy consumption, which is $279J$, increases from that in Figure 1(a), which is $276J$. However, as it will be claimed later the amount of the increase is at the minimum.

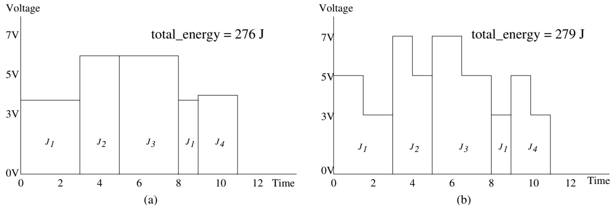

Figure 1: An example illustrating our transformation of continuously variable voltage allocation into discontinuously variable voltage allocation. (a) A continuously variable voltage allocation for tasks in Table 1; (b) A discontinuously variable voltage allocation derived from (a).

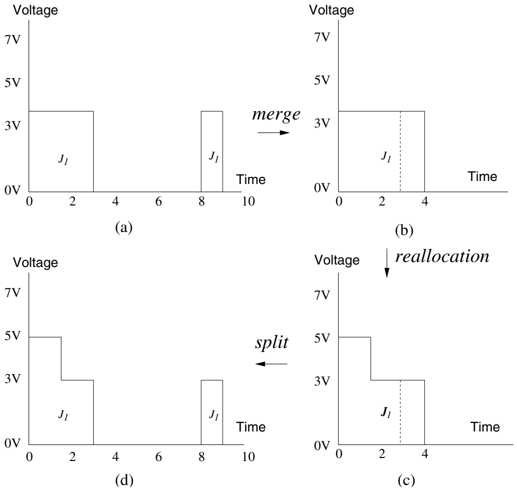

Figure 2: The three steps of our voltage allocation procedure for task J_1 in Figure 1. (a) An initial schedule in Figure 1(a); (b) The result after merging time intervals; (c) The result after voltage reallocation; (d) The result after splitting the time interval.

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 [5]: (Case 1) When there is a task whose (ideal) voltage is lower than any of the valid voltages (i.e., lower than $min\{V_1, \cdots, V_M\}$), the two voltages used for the task are 0.0V and $min\{v_1, \cdots, v_M\}$; (Case 2) When there is a task whose (ideal) voltage is higher than any of the valid voltages (i.e., higher than $max\{v_1, \cdots, v_M\}$), we can safely

[1] For details on how the algorithm works, refer to [5].

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

[3] According to the results in [5] each task always uses the same voltage.

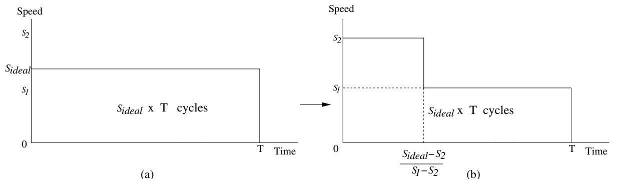

Figure 3: Voltage reallocation (from an ideal voltage to valid voltages). (a) The ideal voltage by [5]; (b) The discrete voltages by [3].

```
Alloc-vt: Algorithm for discretely variable voltage
          allocation and scheduling with multiple tasks)
  • Generate an optimal continuously variable voltage
    allocation and schedule by using the method in [5];
  if (there is a task whose ideal voltage is higher
        than $max\{V_1, \cdots, V_M\}$)
    return (there is no feasible schedule);
  } endif
  foreach (task $J_i$, $i = 1, \cdots, N$) {
    • Merge the time intervals used for the execution
        of $J_i$ into one interval; /* Step 1 */
    • Reallocate voltage for the single task ($J_i$) using
        the method in [3]; /* Step 2 */
    • Split the time interval into the original intervals;
        /* Step 3 */
  } endfor
  return (the voltage allocation and schedule);
```

Figure 4: A summary of the proposed voltage allocation procedure.

conclude that there is no feasible schedule using voltages $V_1, \cdots, V_M$ from the fact in [5] that if there is an optimal continuously variable voltage allocation with feasible schedule with the highest voltage used being V_h, there is no optimal continuously variable voltage allocation with feasible schedule in which its highest voltage used is lower than V_h. The *if-statement* in Alloc-vt checks the second corner case. The following theorem claims that Alloc-vt is optimal.

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

Proof Sketch: Suppose that task J_k is executed for the time period of T in in *any* optimal discretely variable voltage allocation. The two clock speeds must be those immediate s_{i-1} and s_i such that $s_{i-1} < s_{ideal} < s_i$ where $s_0 = 0$. The clock speed corresponding to the ideal voltage is $s_{ideal} = \frac{R_k}{T}$. Hence, the energy consumption, will increase from $P(s_{ideal}) \cdot T$ to $(P(s_{i-1})(\frac{s_{ideal}-s_i}{s_{i-1}-s_i}) + P(s_i)(\frac{s_{ideal}-s_{i-1}}{s_i-s_{i-1}})) \cdot T$.

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

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

Let us consider a continuously variable voltage allocation in which we use the modified power consumption function P' rather than P. Then, the discrete voltage allocation with power function $P(s)$ is identical to the continuously variable voltage allocation with power function $P'(s)$. Note that Yao *et al.*'s algorithm is optimal for any convex power function, and $P'(s)$ is indeed convex. Due to the space limitation, we omit technical details here.

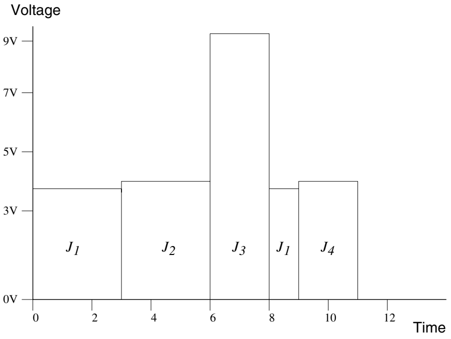

Figure 5: The results of an optimal continuously variable voltage allocation for the tasks in Table 1 with $C_1 = 1.0$, $C_2 = 1.0$, $C_3 = 0.2$ and $C_4 = 1.0$. Note that task J_3 uses a relatively high voltage since its capacitance is low.

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

For example, Figure 5 shows an optimal (continuously variable) voltage allocation for the tasks in Table 1 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 9.0V which is higher than any available discrete (i.e., valid) voltages. In this case, contrary to the second corner case (i.e., Case 2) mentioned in section 3.1, the algorithm in [5] 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 nonuniform switched capacitances. Specifically, we formulate the discretely variable voltage allocation problem into a linear programming (LP) problem and solve it optimally (in polynomial-time).

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

- N : the number of tasks,
- M : the number of discrete voltages,
- a_k : the arrival time of task J_k,
- d_k : the deadline of task J_k,
- R_k : the number of processor cycles required to complete J_k,
- C_k : the average switched capacitance for J_k,
- s_j : the processor clock speed when voltage V_j is applied,

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

- $P(s_j)$: the energy consumed per unit time when the clock speed is s_j.

For a given set of N tasks $J_1, J_2, \cdots, J_N$, we sort the tasks' arrival times and deadlines altogether in non-decreasing order. Let $t_1, t_2, \cdots, t_{2N}$ denote the ordered time sequence.

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

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

$$Minimize \quad \sum_{i=1}^{2N-1} \sum_{k=1}^{N} \sum_{j=1}^{M} C_k \cdot P(s_j) \cdot x^i_{jk} \qquad (5)$$

subject to

$$\sum_{k=1}^{N} \sum_{j=1}^{M} x^i_{jk} \leq t_{i+1} - t_i, \quad i = 1, \cdots, 2N-1 \qquad (6)$$

$$x^i_{jk} = 0, \quad \text{if} \;\; a_k \geq t_{i+1} \;\; \text{or} \;\; b_k \leq t_i \qquad (7)$$

$$\sum_{i=1}^{2N-1} \sum_{j=1}^{M} s_j \cdot x^i_{jk} \geq R_k, \quad k = 1, \cdots, N \qquad (8)$$

$$0 \leq x^i_{jk} \leq t_{i+1} - t_i, \quad {}^\forall \; i, \; j, \; k \qquad (9)$$

Constraint 6 ensures that the total time consumed by all tasks at time interval $[t_i, t_{i+1}]$ should not exceed the time interval. Constraint 7 indicates that each task should be executed only within the interval of its arrival time and deadline. Constraint 8 ensures the voltage allocation and time schedule for each task must satisfy the given total cycle constraint for the task. Constraint 9 follows from the definition of variable x^i_{jk}.

In the following is shown a segment produced by our LP formulation Alloc-vt$_{cap}$ for the tasks with timing constraints in Figure 1 and non-uniform switched capacitances used in Figure 5.[5] 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., $\sum_{i=1}^{5} 30 \cdot x^i_{14}$), the number of clock cycles at voltage V_2 (i.e., $\sum_{i=1}^{5} 50 \cdot x^i_{24}$), and the number at V_3 (i.e., $\sum_{i=1}^{5} 70 \cdot x^i_{34}$) should not be less than 80, which is the total number of clock cycles required for J_4 in Table 1.

$$x^1_{11} + x^1_{21} + x^1_{31} \leq 3 - 0$$
$$x^2_{11} + x^2_{21} + x^2_{31} + x^2_{12} + x^2_{22} + x^2_{32} \leq 5 - 3$$
$$\cdots$$
$$x^4_{11} + x^4_{21} + x^4_{31} + x^4_{13} + x^4_{23} + x^4_{33} \leq 9 - 8$$
$$x^5_{11} + x^5_{21} + x^5_{31} + x^5_{14} + x^5_{24} + x^5_{34} \leq 11 - 9$$

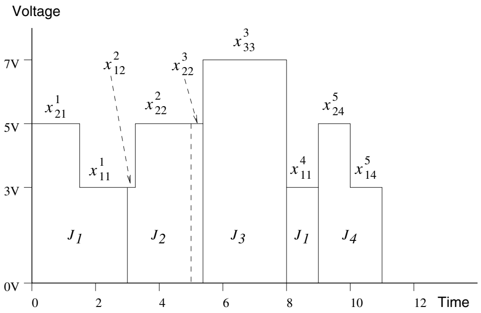

Figure 6: The results of an optimal discretely variable voltage allocation corresponding to our LP solution by Alloc-vt$_{cap}$ **for the example in Figure 5.**

$$\sum_{i=1}^{5} (30 \cdot x^i_{11} + 50 \cdot x^i_{21} + 70 \cdot x^i_{31}) \geq 150$$
$$\sum_{i=1}^{5} (30 \cdot x^i_{12} + 50 \cdot x^i_{22} + 70 \cdot x^i_{32}) \geq 120$$
$$\cdots$$
$$\sum_{i=1}^{5} (30 \cdot x^i_{14} + 50 \cdot x^i_{24} + 70 \cdot x^i_{34}) \geq 80$$

By solving the formulation, we obtain
$x^1_{11} = 1.5, \quad x^1_{21} = 1.5,$
$x^2_{12} = 0.07, \quad x^2_{22} = 1.93,$
$x^3_{22} = 0.43, \quad x^3_{33} = 2.57,$
$x^4_{11} = 1, \quad x^5_{14} = 1, \quad x^5_{24} = 1$, and
$x^i_{jk} = 0$ for the rest.

Specifically, $x^1_{11} = 1.5$ means that task J_1 spends 1.5 units of time in the first interval [0,3] with voltage V_1 (=3.0V), and $x^1_{21} = 1.5$ means that task J_1 spends 1.5 units of time in the first interval [3,5] with voltage V_2 (=5.0V) and so on. Figure 6 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 [11].) In the experiments, we used the 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: (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 non-uniform 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 $P(100MHZ) = 1J/sec.$ at which the average switched capacitance is $1\mu F$.

- **Voltage allocation with uniform switched capacitances:** Table 2 shows a set of processors, each of which has a number of discretely variable clock speeds, controlled by voltage.

We prepared four task sets $\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3$, and $\mathcal{J}_4$, which contain 10, 20, 30, and 40 tasks, respectively. Each task is spec-

[5] We assume that the clock speeds at three different voltages are 30, 50, 70, respectively.

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 2: A set of processors with a number of discretely variable clock speeds.

ified by randomly chosen (a, d, R, C)-tuple, where a (arrival time) and d (deadline) range from 0 to 400 ($sec.$), R (execution cycles) ranges from 1 to 400 ($Millions$), and C (average switched capacitance) ranges from 1 to 4 (μF).

Table 3 shows the comparisons of the energy consumptions for the task sets, in which the switched capacitances of tasks are all identical, produced by an optimal continuously variable voltage allocation by [5] 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 [5] to the immediately higher valid voltages.

Task Set	Processor	Energy Consumption (unit:J)		Reduction
		[5]+*greedy*	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%
Avg.				8.3%

Table 3: Energy consumptions for tasks with uniform capacitances produced by an optimal continuously variable voltage allocation by [5] with greedy discrete voltage reallocation, and Alloc-vt.

- **Voltage allocation with non-uniform switched capacitances:** Table 4 shows the comparisons of the energy consumptions for the task sets, produced by an optimal continuously variable voltage allocation by [5] 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 [5] with greedy voltage reassignment.

Task Set	Processor	Energy Consumption (unit:J)		Reduction
		[5]+*greedy*	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%
Avg.				10.3%

Table 4: Energy consumptions for tasks with nonuniform capacitances produced by an optimal continuously variable voltage allocation by [5] with greedy discrete voltage reallocation, and Alloc-vt$_{cap}$.

5. CONCLUSIONS

We studied a set of important optimization problems which have not been addressed extensively in the literature, although their importance is growing in the area of low-power embedded system design. We addressed, in this paper, the problem of static task scheduling and voltage allocation in dynamically variable voltage processor for minimizing the total processor energy consumption. Specifically, the main contributions are: (1) For given multiple *discrete* 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 *non-uniform load capacitances*, and solve it *optimally.* The technique in (1) is based on the prior results in [5] (which is optimal for continuously variable voltages, but not for discrete ones) and [3] (which is optimal for a single task, but not for multiple tasks), whereas the technique in (2) is based on an efficient linear programming formulation. Both techniques solve the allocation problems optimally in polynomial time. We provided a set of experimental data to show the effectiveness of the proposed techniques in saving energy consumption over the existing methods.

6. ACKNOWLEDGMENTS

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

7. REFERENCES

[1] A. Sinha and A. P. Chanrakasan, “Energy Efficient Real-Time Scheduling,” *ICCAD*, 2001.

[2] L. H. Chandrasena, P. Chandrasena, and M. J. Liebelt, “An Energy Efficient Rate Selection Algorithm for Voltage Quantized Dynamic Voltage Scaling,” *ISSS*, 2001.

[3] T. Ishihara and H. Yasuura, “Voltage Scheduling Problem for Dynamically Variable Voltage Processors,” *ISLPED*, 1998.

[4] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. B. Srivastavaz “Power Optimization of Variable Voltage Core-Based Systems,” *DAC*, 1998.

[5] F. Yao, A. Demers, and S. Shenker, “A Scheduling Model for Reduced CPU Energy,” *IEEE FOCS*, 1995.

[6] Y-R. Lin, C-T. Hwang, and A. C. Wu, “Scheduling Techniques for Variable Voltage Low Power Designs,” *ACM TODAES*, 1997.

[7] J-M. Chang and M. Pedram, “Energy Minimization using Multiple Supply Voltages,” *IEEE TVLSI*, 1997.

[8] M. C. Johnson and K. Roy, “Datapath Scheduling with Multiple Supply Voltages and Level Converters,” *ACM TODAES*, 1997.

[9] S. Raje and M. Sarrafzadeh, “Variable Voltage Scheduling,” *ISLPED*, 1995.

[10] B. Mochocki, X. S. Hu, and G. Quan, “A Realistic Variable Voltage Scheduling Model for Real-Time Applications,” *ICCAD*, 2002.

[11] http://www.ilog.com.