

# A Generalized Technique for Energy-efficient Operating Voltage Set-up in Dynamic Voltage Scaled Processors

Jaewon Seo  
KAIST, Daejeon 305-701, KOREA  
jwseo@jupiter.kaist.ac.kr

Nikil D. Dutt  
University of California, Irvine, CA 92697  
dutt@ics.uci.edu

**Abstract**— Dynamic voltage scaling (DVS) which is an effective energy minimization technique has been well-studied in recent years. Yet the problem of selecting voltage levels for multiple voltage DVS systems remains an unresolved issue. In this paper, we present a novel technique for dealing with the problem of finding  $k$  operating voltages to minimize the energy consumption (voltage set-up problem). A new formulation of the voltage set-up problem is given to make our solution less dependent on the specific DVS scheme. Then it is solved optimally using dynamic programming in polynomial time. With almost the same time complexity we extend the proposed technique to explore the design space to determine the best number of voltage levels. It is confirmed from the experiments that the proposed voltage set-up solution reduces energy consumption by 19.2% on average over that of previous technique [7].

## I. INTRODUCTION

Over the past decades there have been enormous efforts to minimize the energy consumption of CMOS circuit systems. Dynamic voltage scaling (DVS) framework – involving dynamic adjustments of the supply voltage and the corresponding operating clock frequency – has emerged as one of the most effective energy minimization techniques. A one-to-one correspondence between the voltage and the clock frequency in CMOS circuits imposed an inherent constraint to DVS techniques to ensure that voltage adjustments do not violate target system timing deadlines.

There are several contemporary classifications of DVS schemes; soft real-time vs. hard real-time, inter-task vs. intra-task, online vs. offline, continuous voltage vs. discrete voltage, etc. According to the domain of the target system, the two key players, voltage adjustment and timing constraint raise various practical issues, which results in a diversity of DVS techniques. Although in recent years several instances of DVS problem with less specific constraints have been solved optimally, e.g. [1, 2, 3], many researchers are still working on more specific unsolved problems. Since most existing works assume that the available voltage set is given and focus on how to exploit the pre-defined voltages, one of the major remaining challenges – particularly for customizable Systems-on-Chips (SoCs) – is the determination of energy-efficient operating points (voltage/frequency points) that are tuned for specific applications.

To the best of our knowledge, the operating voltage set-up problem has been dealt with in the past few years, only in the

following contexts: Chen and Sarrafzadeh [4] addressed the gate-level supply voltage selection problem, where the number of available voltages is fixed to two (dual voltages). They proposed an optimal dual voltage estimation technique starting from the maximal weighted independent set-based analysis of the delay and power consumption within a circuit.

Dhar and Maksimović [5] presented a multiple voltage distribution technique for finite impulse response (FIR) filter design. For a given filter order  $N$ , the proposed algorithm finds a set of  $2N + 1$  voltages based on the method of Lagrange multipliers to minimize the average energy consumption of each module.

Recently, Hua and Qu [6] proposed an application level voltage set-up technique. They considered the problem of determining the number of voltages and each voltage value on a multiple-voltage application-specific DVS system such that the system's energy consumption is minimized. An analytical solution for dual-voltages system and a heuristic-based numerical method for the general case were proposed. One limitation of their approach is that the derivation of the solutions is based on a simple application model in which each application is specified only by execution time, deadline and probability to be executed. They also assumed that the execution time which varies dramatically due to the input data is fixed and known a priori. The intrinsic dependency on the application model of their approach makes it virtually impossible to apply to other popular DVS domains, such as intra-task DVS [3, 8].

Buss et al. [7] presented an execution profile-based operating points exploration technique targeting the outlined static intra-task voltage scheduling. In their approach, a cumulative energy distribution graph where the energy consumptions are accumulated with increase of the voltage level is constructed from the execution profile and then divided uniformly for no specific reason to determine the operating point set. Although they made no mention of its applicability, the proposed technique can be easily incorporated into other DVS domains.

The major contributions of our work – both different from, as well as complementary to related previous efforts – are: (1) *We formulate the voltage set-up problem in a new way that the proposed technique is less dependent on the specific domain making it widely applicable to various DVS schemes,* (2) *we solve the problem of finding  $k$  voltages optimally in the sense that the minimum energy consumption is obtained and* (3) *we provide a fast exploration scheme to help determine the number of operating voltages at the initial system design stage.*

The remainder of this paper is organized as follows. In Section II, we present a new formulation of the voltage set-up problem. Section III explains the proposed approach. In

Section IV, experimental results are provided to validate our methods. Finally, concluding remarks are given in Section V.

## II. PROBLEM FORMULATION

### A. Voltage set-up problem

Without loss of generality we can define the voltage set-up problem as follows.

**Problem 1 (*k*-Voltage Set-up Problem)** *For a given voltage scaled processor with  $k$  operating voltage levels and its specific application or a set of applications, determine each voltage level of  $k$  voltages to minimize the energy consumption of the system within a specified timing constraint.*

Note that depending on the domain of the target system and the corresponding voltage scheduling technique we employ, we have different specific instances of the above voltage set-up problem. For example, the timing constraint of a typical inter-task voltage scheduling is represented by the arrival time and the deadline of each real-time task and the remaining slack time caused by earlier termination of the task is reclaimed by other tasks. On the other hand the typical intra-task voltage scheduling specifies its timing constraint by simply assigning a deadline to the task and schedules it not to produce any slack time. Since the way of specifying the timing constraint is quite different from each other, the voltage set-up problems for these two voltage scheduling techniques should be treated differently. Moreover in most cases we obtain the actual energy consumption of the system only after the voltage scheduling procedure has finished, thus the quality of the solution to the voltage set-up problem is significantly affected by the choice of the voltage scheduling technique. The strong correlation between the voltage set-up problem and the employed voltage scheduling technique necessitates devising an individual voltage set-up strategy for each voltage scheduling technique, which is undesirable in the sense that it is difficult to cover the diversity of DVS techniques. One of the promising features of our method is that our voltage set-up technique is widely applicable to various voltage scheduling schemes and for that purpose we formulate the voltage set-up problem trying to minimize the dependency on the voltage scheduling technique, which will be explained in more detail later. However for the sake of illustration, we use the recently proposed optimal intra-task voltage scheduling technique [3], which achieves the minimum average energy consumption and can handle both limited and unlimited set of voltages. We begin by examining the desirable voltage schedule for our voltage set-up solution.

### B. Energy model

It is known [8] that the amount of energy consumption during the execution of a task is expressed as

$$E \propto V_{DD}^2 \times n_{total} \quad (1)$$

where  $n_{total}$  is the total number of instruction cycles executed. The relationship between clock frequency and voltage in CMOS circuits is

$$f_{CLK} \propto (V_{DD} - V_T)^\alpha / V_{DD} \approx V_{DD} \quad (2)$$

where  $V_T$  is the threshold voltage and  $\alpha$  is the velocity saturation index. Based on the one-to-one correspondence between

clock frequency  $f_{CLK}$  and supply voltage  $V_{DD}$ , we use the term *voltage* and corresponding *speed* (i.e., clock frequency) interchangeably throughout the paper.

Since many tasks exhibit different behaviors depending on input data, we cannot directly apply the simple energy model of Eq. (1). Thus, we extend the energy model of Eq. (1) so that it can handle general cases. One way to deal with the non-uniform execution path lengths is to consider the average (i.e., expected) energy consumption, which can be expressed as

$$E_{avg} = \sum_{\forall \text{execution path } \pi} (P(\pi) \cdot E(\pi)) \quad (3)$$

where  $P(\pi)$  is the probability that the execution of the task follows path  $\pi$  and  $E(\pi)$  is the energy consumption for that path. The fact that many tasks in real-time environments are executed repeatedly provides the rationale for the use of this model.

### C. Optimal intra-task voltage scheduling

We denote a real-time task  $\tau$  as having a corresponding deadline  $D_\tau$ . The task  $\tau$  is represented by its Control Flow Graph (CFG)  $G_\tau = (V, E)$ , where  $V$  is the set of basic blocks and  $E$  corresponds to the set of directed edges which impose the control dependency between basic blocks. The set of successor basic blocks of any  $v \in V$  is denoted by  $succ(v)$ . Each basic block  $b_i$  is annotated with its non-zero number of execution cycles  $n_i$  (by definition of basic block<sup>1</sup>,  $n_i$  is constant.) Each directed edge  $(b_i, b_j)$  is annotated with probability  $p_j$  that the control flow follows the edge. Given a task's CFG and its execution profile which offers the probabilities of execution paths, if we set the operating speed of basic block  $b_i$  to  $\delta_i/T_i$  (and adjust the supply voltage accordingly) then we have the minimum average energy consumption of Eq. (3), where  $\delta_i$  is defined as

$$\delta_i = \begin{cases} n_i, & \text{if } succ(b_i) = \emptyset \\ n_i + \sqrt[3]{\sum_{\forall b_j \in succ(b_i)} p_j \delta_j^3}, & \text{otherwise} \end{cases} \quad (4)$$

and  $T_i$  is the remaining time to deadline from basic block  $b_i$ . More detailed explanation including the proof of the optimality is presented in [3]. Consider for example a simple real-time task  $\tau_{simple}$  shown in Fig. 1(a). Each node represents a basic block and each edge indicates a possible flow of control between two basic blocks. The number within each node indicates its number of execution cycles and the number assigned to each edge indicates the probability. Fig. 1(b) shows the procedure of calculating each  $\delta_i$  value according to Eq. (4). Once we have computed every  $\delta$  value for each basic block, the operating speed of basic block  $b_i$  is simply obtained by dividing  $\delta_i$  by the remaining time to deadline. Note that the thick, dotted and regular arrows in Fig. 1(b) represent increase, decrease and no change of the speed, respectively and only the gray-colored basic blocks receive new operating points.

### D. Voltage distribution graph

What we can get from the previous optimal intra-task voltage scheduling technique is the minimum energy speed/voltage

<sup>1</sup>Typically, a basic block is defined as a sequence of instructions that does not contain any *jumps*.

TABLE I  
EXECUTION PROFILE OF TASK  $\tau_{\text{simple}}$  ( $D_{\tau_{\text{simple}}} = 10$  UNIT TIME)

Execution Path	Exec. Prob.	Schedule of Operating Points		Expected Number of Execution Cycles
		Speed	Voltage	
$b_0 b_1 b_5 b_6 b_8$	0.14	2.93 - 2.78 - 2.78 - 1.99 - 1.99	$v_6 v_3 v_3 v_1 v_1$	0.84 - 0.42 - 0.14 - 1.12 - 0.70
$b_0 b_1 b_5 b_7 b_8$	0.56	2.93 - 2.78 - 2.78 - 2.92 - 2.92	$v_6 v_3 v_3 v_5 v_5$	3.36 - 1.68 - 0.56 - 7.84 - 2.80
$b_0 b_2 b_3 b_5 b_6 b_8$	0.054	2.93 - 3.23 - 3.14 - 3.14 - 2.26 - 2.26	$v_6 v_8 v_7 v_7 v_2 v_2$	0.324 - 0.216 - 0.108 - 0.054 - 0.432 - 0.270
$b_0 b_2 b_3 b_5 b_7 b_8$	0.216	2.93 - 3.23 - 3.14 - 3.14 - 3.30 - 3.30	$v_6 v_8 v_7 v_7 v_9 v_9$	1.296 - 0.864 - 0.432 - 0.216 - 3.024 - 1.080
$b_0 b_2 b_4 b_5 b_6 b_8$	0.006	2.93 - 3.23 - 3.89 - 3.89 - 2.80 - 2.80	$v_6 v_8 v_{10} v_{10} v_4 v_4$	0.036 - 0.024 - 0.042 - 0.006 - 0.048 - 0.030
$b_0 b_2 b_4 b_5 b_7 b_8$	0.024	2.93 - 3.23 - 3.89 - 3.89 - 4.10 - 4.10	$v_6 v_8 v_{10} v_{10} v_{11} v_{11}$	0.144 - 0.096 - 0.168 - 0.024 - 0.336 - 0.120

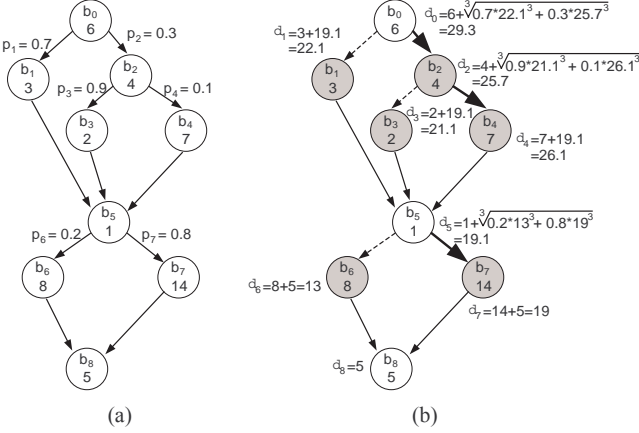


Fig. 1. (a) CFG of example task  $\tau_{\text{simple}}$ ; (b) Calculation of  $\delta$  values

schedule for a given task, which is the most desirable result we can have.

Table I shows the execution profile of the task  $\tau_{\text{simple}}$  obtained assuming that the deadline  $D_{\tau_{\text{simple}}}$  is 10 unit time. For each execution path, the probability that the execution follows the path and the minimum energy speed/voltage schedule are presented. Note that  $v_i$  denotes the voltage corresponding to the  $i$ th higher operating speed. Last column of the table indicates the expected number of cycles to be executed at the corresponding operating point in the path, which is obtained by multiplying each basic block's number of cycles by the path probability.

To see how much energy is consumed at each voltage level, we group all instances of basic blocks that are executed at the same voltage level and construct *voltage distribution graph* where  $x$ -axis represents each voltage used in the obtained schedule and  $y$ -axis corresponds to the total expected number of cycles executed at each voltage. For example, only basic block  $b_1$  and  $b_5$  in both path  $b_0 b_1 b_5 b_6 b_8$  and  $b_0 b_1 b_5 b_7 b_8$  are executed at voltage  $v_3$ , the total expected execution cycles for  $v_3$  is  $0.42 + 0.14 + 1.68 + 0.56 = 2.80$ . Fig. 2 shows the voltage distribution graph for task  $\tau_{\text{simple}}$ .

Suppose that we have  $n$  voltage levels in the voltage distribution graph. Let  $v_i$  be the  $i$ th higher voltage ( $1 \leq i \leq n$ ) in the graph and  $c_i$  be the total expected number of cycles executed at the voltage  $v_i$ . Then the average energy consumption for the voltage distribution graph is given by the following equation.

$$E_{\text{ideal}} = \sum_{1 \leq i \leq n} v_i^2 \cdot c_i \quad (5)$$

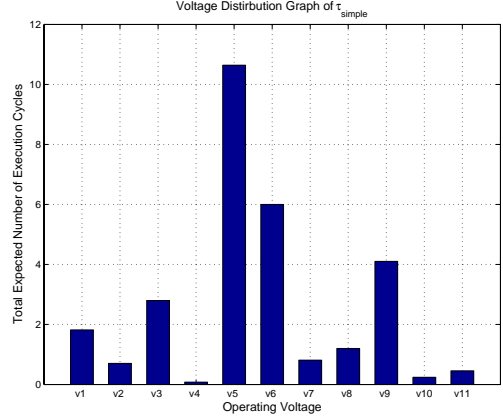


Fig. 2. Voltage distribution graph of  $\tau_{\text{simple}}$  obtained from the execution profile shown in Table I

Note that this equation is essentially identical to Eq. (3) since the energy consumption for each execution path of Eq. (3) can be decomposed into the sum of energy consumptions for individual basic blocks and the all instances of basic blocks that are executed at the same voltage level are grouped together producing the energy consumption for each voltage level as shown in Eq. (5).

Since the voltage schedule is obtained without any limitation on the number of voltage levels and it produces the optimal energy consumption in the sense that the minimum average energy consumption is achieved, we call its voltage distribution graph *ideal voltage distribution graph* and the set of voltage levels used in the schedule *ideal voltage set*.

### E. Minimum energy $k$ -voltage cover problem

Returning to Problem 1, since only  $k$  voltages are permitted, we should carefully choose  $k$  voltages to *replace the ideal voltage set*. It can be considered that each selected voltage substitutes for a subset of the ideal voltage set i.e., we select  $k$  voltages  $\hat{v}_1, \hat{v}_2, \dots, \hat{v}_k$  and corresponding  $k$  subsets  $V_1, V_2, \dots, V_k$  of the ideal voltage set  $V_{\text{ideal}}$  such that  $\forall i, V_i \neq \emptyset$  and  $\bigcup_{V_i} V_i = V_{\text{ideal}}$ . Since the operating speed is proportional to the corresponding voltage level, replacing a voltage  $v$  with a lower voltage implies that the portion of the task which was originally executed at the speed corresponding to the voltage  $v$  is now executed at a new lower speed, which may cause missing the deadline due to the increased execution time. Thus each voltage  $\hat{v}_i$  and the corresponding subset  $V_i$  should be selected

to satisfy following constraint.

$$\text{Voltage-cover Condition : } \forall v \in V_i, v \leq \hat{v}_i \quad (6)$$

We call the set of  $k$  voltages satisfying the above condition  $k$ -voltage cover. After replacing the ideal voltage set with  $k$ -voltage cover, the energy consumption becomes:

$$E_{k\text{-cover}} = \sum_{1 \leq i \leq k} \sum_{\forall v_j \in V_i} \hat{v}_i^2 \cdot c_j \quad (7)$$

Then our purpose is to find a  $k$ -voltage cover that minimizes the Eq. (7), which can be stated as follows.

**Problem 2 (Minimum Energy  $k$ -Voltage Cover Problem)** For a given ideal<sup>2</sup> voltage distribution graph with  $n$  voltage levels and corresponding expected numbers of execution cycles, find a  $k$ -voltage cover that leads to the minimum energy consumption.

Note that this new formulation only requires the information about the tasks' execution profile from which the voltage distribution graph is obtained, and does not assume anything about the specific voltage scheduling technique.

### III. PROPOSED APPROACH

#### A. Finding minimum energy $k$ -voltage cover

For the sake of illustration, assume that the number of available voltage levels is three ( $k=3$ ). Note that the following voltage cover selection technique is easily extended to handle various numbers of voltage levels and we also present the general version of the algorithm at the end of this section. For a given ideal voltage distribution graph with voltage levels  $v_1 < v_2 < \dots < v_n$  and corresponding expected numbers of execution cycles  $c_1, c_2, \dots, c_n$ , we are to select three 'representative' voltages  $\hat{v}_{\text{low}}, \hat{v}_{\text{mid}}$  and  $\hat{v}_{\text{high}}$  ( $\hat{v}_{\text{low}} < \hat{v}_{\text{mid}} < \hat{v}_{\text{high}}$ ) to cover the entire voltage spectrum  $v_1, v_2, \dots, v_n$ . Without loss of generality suppose that  $v_1 < \dots < v_i \leq \hat{v}_{\text{low}} < v_{i+1} < \dots < v_j \leq \hat{v}_{\text{mid}} < v_{j+1} < \dots < v_n \leq \hat{v}_{\text{high}}$ , then we set  $V_{\text{low}} = \{v_1, v_2, \dots, v_i\}$ ,  $V_{\text{mid}} = \{v_{i+1}, v_{i+2}, \dots, v_j\}$  and  $V_{\text{high}} = \{v_{j+1}, v_{j+2}, \dots, v_n\}$  i.e., we use  $\hat{v}_{\text{low}}$  instead of  $v_1, v_2, \dots, v_i$ ,  $\hat{v}_{\text{mid}}$  instead of  $v_{i+1}, v_{i+2}, \dots, v_j$  and so on. Recall that due to the linear dependency of the operating speed on the operating voltage, the selected voltage, say  $\hat{v}_{\text{low}}$ , should not substitute for any other voltages higher than it to avoid possible deadline miss (*voltage-cover condition*). From Eq. (7) the energy consumption for this 3-voltage cover is expressed as

$$E_{3\text{-cover}} = \sum_{0 < m \leq i} \hat{v}_{\text{low}}^2 \cdot c_m + \sum_{i < m \leq j} \hat{v}_{\text{mid}}^2 \cdot c_m + \sum_{j < m \leq n} \hat{v}_{\text{high}}^2 \cdot c_m \quad (8)$$

Our objective is to select  $\hat{v}_{\text{low}}, \hat{v}_{\text{mid}}$  and  $\hat{v}_{\text{high}}$  appropriately so as to make the above energy consumption approximate the ideal energy consumption of Eq. (5) as closely as possible. Note that it is obvious that we should set these voltages tightly making  $\hat{v}_{\text{low}} = v_i, \hat{v}_{\text{mid}} = v_j$  and  $\hat{v}_{\text{high}} = v_n$  for that purpose.

<sup>2</sup>In this definition the term *ideal* does not require that the given voltage distribution graph shows the minimum energy consumption. It is ideal in the sense that unless  $k=n$ , we cannot achieve the minimum energy consumption using a  $k$ -voltage cover.

Let  $E_{\text{sub}}(l, m)$  be the energy consumption for sub-spectrum of execution cycles,  $c_l, c_{l+1}, \dots, c_m$  ( $1 \leq l \leq m \leq n$ ) where we use the voltage  $v_m$  as a 'representative' voltage. More specifically,  $E_{\text{sub}}(l, m)$  is defined as

$$E_{\text{sub}}(l, m) = \sum_{l \leq i \leq m} v_m^2 \cdot c_i \quad (9)$$

Let  $E_{\text{min}}^{k\text{-cover}}(m)$  be the minimum energy consumption for execution cycles  $c_1, c_2, \dots, c_m$  where only  $k$  voltages are permitted. We wish to determine  $E_{\text{min}}^{3\text{-cover}}(n)$ , which can be expressed recursively as

$$E_{\text{min}}^{3\text{-cover}}(n) = \min_{2 \leq i < n} \{E_{\text{min}}^{2\text{-cover}}(i) + E_{\text{sub}}(i+1, n)\} \quad (10)$$

$$\text{where } E_{\text{min}}^{2\text{-cover}}(i) = \min_{1 \leq j < i} \{E_{\text{min}}^{1\text{-cover}}(j) + E_{\text{sub}}(j+1, i)\} \quad (11)$$

$$\text{and } E_{\text{min}}^{1\text{-cover}}(j) = E_{\text{sub}}(1, j) \quad (12)$$

The intuition is based on the fact that if we already know how to select a 2-voltage cover for any sub-spectrum of voltage levels, then we can easily find a 3-voltage cover since we need to select only one additional voltage. Note that Eq. (12) implies that in case only one voltage is to be selected as a 'representative', we should choose the highest  $v_j$  among all available voltages  $v_1, v_2, \dots, v_j$  so as to avoid any timing violation (*voltage-cover condition*).

In general, the minimum energy consumption of  $k$ -voltage cover with  $n$  voltage levels,  $E_{\text{min}}^{k\text{-cover}}(n)$  ( $1 < k \leq n, n > 0$ ), can be expressed as

$$E_{\text{min}}^{k\text{-cover}}(n) = \min_{k-1 \leq i < n} \{E_{\text{min}}^{(k-1)\text{-cover}}(i) + E_{\text{sub}}(i+1, n)\} \quad (13)$$

The following theorem states that based on the above derivation of the minimum energy consumption for  $k$ -voltage cover we can solve the minimum energy  $k$ -voltage cover problem in polynomial time.

*Theorem 1: For a given ideal voltage distribution graph with  $n$  voltages, we can determine its minimum energy  $k$ -voltage cover in  $O(kn^2)$  time.*

*Proof:* We can compute  $E_{\text{sub}}(l, m)$  of Eq. (9) for all possible pairs of  $(l, m)$  ( $1 \leq l \leq m \leq n$ ) in  $O(n^2)$  time. (See lines 1-6 of Algorithm 1.) For fixed  $p$  and  $q$  ( $1 < p \leq q \leq n$ ),  $E_{\text{min}}^{p\text{-cover}}(q)$  is obtained in  $O(n)$  time from Eq. (13), provided that  $E_{\text{min}}^{(p-1)\text{-cover}}(i)$  has already been calculated for  $p-1 \leq i < q$ . Consider we compute all values of  $E_{\text{min}}^{p\text{-cover}}(\cdot)$  at the  $p$ th stage. Then each stage can be completed in  $O(n^2)$  assuming that the previous stage has already been completed. Since  $E_{\text{min}}^{k\text{-cover}}(n)$  is calculated at the  $k$ th stage, we have  $O(kn^2)$  time complexity to obtain the minimum amount of energy consumption with  $k$ -voltage cover. Including the time taken to compute  $E_{\text{sub}}(l, m)$ , finally we have the time complexity of  $O(kn^2)$ . Note that the list of voltages in the  $k$ -voltage cover is obtained by backtracking the calculation process as described in Algorithm 1, which can be done in  $O(k)$  time. ■

Algorithm 1 describes the procedure of finding the minimum energy  $k$ -voltage cover for a given voltage distribution graph. First, we compute  $E_{\text{sub}}(l, m)$  according to Eq. (9) for all pairs of  $(l, m)$  (Lines 1-6). Then each  $E_{\text{min}}^{1\text{-cover}}(i)$  is obtained by simply setting it equal to  $E_{\text{sub}}(1, i)$  (Lines 7-9). For  $p > 2$ , each  $E_{\text{min}}^{p\text{-cover}}(q)$  is calculated using Eq. (13) starting from the 2nd

---

**Algorithm 1** Finding minimum energy  $k$ -voltage cover

---

```
1: for all  $m = 1, 2, \dots, n$  do
2:    $E_{\text{sub}}[m][m] \leftarrow v_m^2 \cdot c_m$ ;
3:   for all  $l = m - 1, m - 2, \dots, 1$  do
4:      $E_{\text{sub}}[l][m] \leftarrow E_{\text{sub}}[l + 1][m] + v_m^2 \cdot c_l$ ;
5:   end for
6: end for
7: for all  $i = 1, 2, \dots, n$  do
8:    $E_{\text{min}}[1][i] = E_{\text{sub}}[1][i]$ ;
9: end for
10: for all  $p = 2, 3, \dots, k$  do
11:   for all  $q = p, p + 1, \dots, n$  do
12:      $E_{\text{min}}[p][q] \leftarrow +\infty$ ;
13:     for all  $i = p - 1, p, \dots, q - 1$  do
14:        $tmp \leftarrow E_{\text{min}}[p - 1][i] + E_{\text{sub}}[i + 1][q]$ ;
15:       if  $tmp < E_{\text{min}}[p][q]$  then
16:          $E_{\text{min}}[p][q] \leftarrow tmp$ ;
17:          $history[p][q] \leftarrow i$ ;
18:       end if
19:     end for
20:   end for
21: end for
22:  $selected \leftarrow n$ ;
23: for all  $i = k, k - 1, \dots, 2$  do
24:   output  $selected$ ;
25:    $selected \leftarrow history[i][selected]$ ;
26: end for
27: output  $selected$ ;
```

---

stage ( $p=2$ ) to the  $k$ th stage ( $p=k$ ) (Lines 10-21). Note that the array ‘*history*’ keeps the record of selected voltages i.e., the  $i$  values of Eq. (13) that minimizes its quantity (Line 17). Finally, the algorithm outputs the selected voltages starting from the highest voltage referencing ‘*history*’ array (Lines 22-27).

### B. Determine $k$ - the number of voltage levels

If there is no hardware overhead to increase the number of operating voltage levels it is obvious that we should use as many voltages as possible. However it does require additional hardware and will introduce area, delay and power penalties on the voltage regulator or DC-DC converter. Thus it becomes important to investigate the tradeoff between the number of voltage levels and the overhead they incur.

To help determine the number of operating voltage levels at the initial design stage, we provide  $k$ - $E_{\text{min}}^{k\text{-cover}}$  graph where the minimum energy consumption  $E_{\text{min}}^{k\text{-cover}}$  is plotted varying the number of voltage levels  $k$ . Then the designer can select the proper number of voltages which leads to the minimum energy consumption within acceptable overheads. The following theorem states that we can construct  $k$ - $E_{\text{min}}^{k\text{-cover}}$  graph within almost the same time complexity as the one taken to determine the minimum energy  $k$ -voltage cover.

*Theorem 2:* For a given ideal voltage distribution graph with  $n$  voltages, we can determine its all minimum energy voltage covers in  $O(n^3)$  time.

*Proof:* Recall that we compute the minimum energy  $k$ -voltage cover at the  $k$ th stage of calculation, provided that all the previous stages have already been completed. Therefore by the time we compute  $E_{\text{min}}^{n\text{-cover}}(n)$ , we already have all other minimum energy voltage covers. Thus the time complexity is  $O(n^3)$ . ■

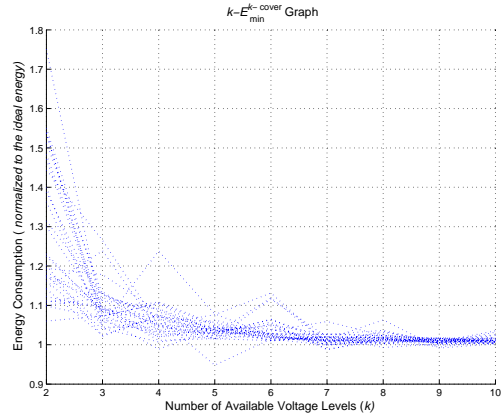


Fig. 3.  $k$ - $E_{\text{min}}^{k\text{-cover}}$  graph for 30 randomly generated tasks

We omit the algorithm of drawing the  $k$ - $E_{\text{min}}^{k\text{-cover}}$  graph since it is a simple extension of Algorithm 1 by adding a step between line 20 and line 21 that plots  $(p, E_{\text{min}}^{p\text{-cover}}(n))$ . Note that in this case the input  $k$  of Algorithm 1 is set to  $n$ .

## IV. EXPERIMENTAL RESULTS

We test the proposed approach on a set of benchmark tasks as well as randomly generated tasks to demonstrate the effectiveness of our technique. The voltage distribution graph is constructed from the optimal execution profile which is obtained using our optimal intra-task voltage scheduling technique. We set the *slack factor* of each task  $\tau$ , defined as  $(D_\tau - W_\tau)/D_\tau$  where  $W_\tau$  is the worst-case execution time of  $\tau$  executed at the highest voltage, to 0.1 giving a tight timing constraint. A random task is generated by splitting the arbitrarily chosen basic blocks repeatedly until the number of branches reaches some fixed number. The probability at each branch of the generated tasks is drawn from a random normal distribution with a standard deviation of 1.0 and a mean 0.5 and the number of execution cycles of each basic block is restricted such that the length of the longest basic blocks does not exceed 1000 times length of the shortest one.

Fig. 3 shows  $k$ - $E_{\text{min}}^{k\text{-cover}}$  graph for 30 randomly generated tasks varying  $k$  from 2 to 10. Note that the energy consumption of each task is normalized to the corresponding ideal energy consumption ( $k=\infty$ ). We can easily see that the energy-efficiency increases as we are permitted to use more voltages. However it is reported that only 10% more energy is consumed when we use 4 voltages ( $k=4$ ) compared to the ideal energy consumption.

Fig. 4 shows the distribution of the selected voltages in the minimum energy 10-voltage cover of 100 randomly generated tasks.  $x$ -axis indicates the each voltage level normalized to the highest one and  $y$ -axis represents the corresponding frequency of selection. Contrary to the expectation that the selected voltages will reveal a uniform distribution, it is observed that the higher voltage levels are more likely to be chosen, which suggests that the simple uniform selection-based voltage set-up process may fail to find the energy-efficient solution.

The effectiveness of our technique is evaluated by comparing the results against those of the other technique proposed by Buss et al. [7]. To verify the applicability of our solution we also obtain the energy consumption using another voltage

TABLE II  
COMPARISONS OF RESULTS PRODUCED BY OUR TECHNIQUE AND BUSS [7]

Benchmark Task [9]	Energy Overhead (%) (compared to ideal energy consumption)											
	Optimal Intra-Task Scheduling [3]						Shin [8]					
	$k=2$		$k=4$		$k=8$		$k=2$		$k=4$		$k=8$	
	Ours	[7]	Ours	[7]	Ours	[7]	Ours	[7]	Ours	[7]	Ours	[7]
amotsa	8.03	5.51	1.04	2.25	0.12	0.12	20.46	11.24	7.56	7.88	4.89	4.89
dawson	7.9	29.39	0.24	29.39	0.14	0.14	7.9	29.52	0.37	29.52	0.25	0.25
gcf	6.07	6.07	1.09	6.07	0.03	6.07	7.45	7.45	8.28	7.45	7.42	7.45
gser	3.61	6.19	2.66	4.35	0.33	2.53	5.81	6.19	5.64	5.81	2.72	4.34
gsimp	0.01	210.38	0.0	210.38	0.0	0.0	210.33	210.38	210.32	210.38	210.32	210.32
hypser	3.49	3.02	1.09	0.24	0.02	0.51	3.49	3.02	2.93	3.02	2.49	2.49
igray	11.68	11.68	1.72	8.61	0.0	8.43	11.68	11.68	4.04	11.1	4.0	11.08
realft	26.3	26.3	7.43	26.3	4.49	7.43	26.3	26.3	9.48	26.3	9.32	9.48
rtnewt	5.3	33.06	0.23	5.3	2.86	0.23	20.68	33.06	5.55	20.68	5.25	5.55
trsec	4.62	9.08	0.23	0.96	0.01	0.21	4.62	9.08	6.93	8.5	6.83	6.92
sncndn	23.15	47.33	10.62	47.33	4.4	16.5	33.62	47.33	26.49	47.33	20.34	32.41
trapzd	0.0	0.07	0.0	0.0	0.0	0.0	0.0	0.07	0.0	0.0	0.0	0.0
Average	<b>8.79</b>	35.01	<b>2.30</b>	31.00	<b>1.13</b>	3.79	31.71	35.66	25.88	34.08	24.67	26.61

scheduling technique by Shin et al. [8]. Table II shows the comparison results on a set of real tasks [9] for  $k=2, 4$  and  $8$ . On average our approach combined with the optimal intra-task voltage scheduling reduces the energy consumption by 19.2% over that of [7], which is only 4.1% more energy than the ideal one. Due to the dependency of resulting energy consumption on a specific DVS technique, the quality of the voltage schedule obtained using [8] is poor compared to the one from the optimal scheduling. However, the proposed technique still has an advantage over [7] with the same scheduling technique showing on average 4.7% reduction of the energy consumption, which suggests that the proper selection of the DVS technique is the essence in the minimization of the system's energy consumption.

## V. CONCLUSIONS

We presented a novel technique for dealing with the voltage set-up problem of finding  $k$  operating voltages to minimize the energy consumption. The major contributions of our work are: (1) We formulate the voltage set-up problem in a new way that the proposed technique is less dependent on the specific domain making it widely applicable to various DVS schemes, (2) we solve the problem of finding  $k$  voltages optimally in the sense that the minimum energy consumption is obtained and (3) we provide a fast exploration scheme to help determine the number of operating voltages at the initial system design stage. It is confirmed from the experiments that the proposed voltage set-up solution reduces energy consumption by 19.2% on average over that of previous technique [7].

## REFERENCES

[1] F. Yao, A. Demers and S. Shenker, "A Scheduling Model for Reduced CPU Energy", *IEEE Symposium on Foundations of Computer Science*, Oct. 1995

[2] W. Kwon and T. Kim, "Optimal Voltage Allocation Techniques for Dynamically Variable Voltage Processors", *Design Automation Conference*, Jun. 2003

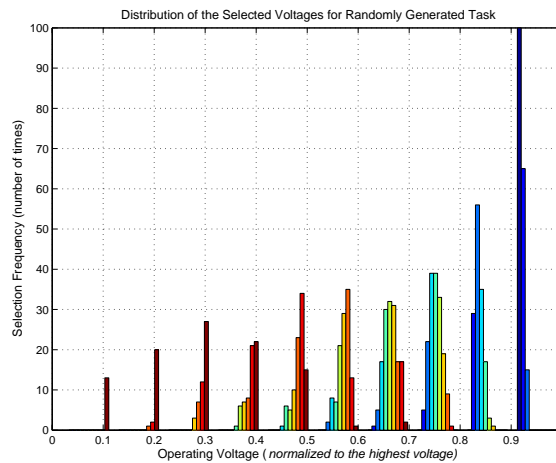


Fig. 4. Distribution of selected voltages in the minimum energy 10-voltage cover for 100 randomly generated tasks

[3] J. Seo, T. Kim and K. Chung, "Profile-based Optimal Intra-task Voltage Scheduling for Hard Real-time Applications", *Design Automation Conference*, Jun. 2004

[4] C. Chen and M. Sarrafzadeh, "Provably Good Algorithm for Low Power Consumption with Dual Supply Voltages", *International Conference on Computer-Aided Design*, Nov. 1999

[5] S. Dhar and D. Maksimović, "Low-Power Digital Filtering Using Multiple Voltage Distribution and Adaptive Voltage Scaling", *International Symposium on Low Power Electronics and Design*, Jul. 2000

[6] S. Hua and G. Qu, "Approaching the Maximum Energy Saving on Embedded Systems with Multiple Voltages", *International Conference on Computer-Aided Design*, Nov. 2003

[7] M. Buss, T. Givargis and N. Dutt, "Exploring Efficient Operating Points for Voltage Scaled Embedded Processor Cores", *IEEE International Real-Time Systems Symposium*, Dec. 2003

[8] D. Shin, J. Kim and S. Lee, "Intra-Task Voltage Scheduling for Low-Energy Hard Real-Time Applications", *IEEE Design & Test of Computers*, 2001

[9] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, 1988.