Voltage Set-up Problem on Embedded Systems with Multiple Voltages*

Shaoxiong Hua and Gang Qu
Electrical and Computer Engineering Department and
Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742, USA
{shua, gangqu}@eng.umd.edu

Abstract

Dynamic voltage scaling (DVS), arguably the most effective energy reduction technique, can be enabled by having multiple voltages physically implemented on the chip and allowing the operating system to decide which voltage to use at run-time. Indeed, this is predicted as the future low-power system by International Technology Roadmap for Semiconductors (ITRS). There still exist many important unsolved problems on how to reduce the system's dynamic and/or total power by DVS. One of such problems, which we refer to as the voltage set-up problem, is *how many levels and at which values should voltages be implemented for the system to achieve the maximum energy saving.* It challenges whether DVS technique's full potential in energy saving can be reached on multiple-voltage systems. In this paper, (1) we derive analytical solutions for dual-voltage system. (2) For the general case that does not have analytic solutions, we develop efficient numerical methods that can take the overhead of voltage switch and leakage into account. (3) We demonstrate how to apply the proposed algorithms on system design. (4) Interestingly, the experimental results, on both real life DSP applications and random created applications, suggest that multiple-voltage DVS systems with only a couple levels of voltages, when set up properly, can be very close to DVS technique's full potential in energy saving.

Keywords: System analysis and design, Design automation, Voltage, Energy management.

*Parts of this manuscript have been published in IEEE/ACM International Conference on Computer Aided Design, pp. 26-29, November 2003.

1 Introduction

Energy consumption has become a major design issue for modern embedded systems especially battery-operated portable devices. The aggressive push for low-power design has prompted the International Technology Roadmap for Semiconductors (ITRS) to predict that the future system will feature multiple supply voltages (V_{dd}), and multiple threshold voltages (V_{th}), on the same chip [1]. This enables the dynamic voltage scaling (DVS) technique that varies the supply voltage and clock frequency according to workload at run time to save energy. DVS achieves the highest energy efficiency for time-varying computational loads if voltage can be varied arbitrarily [2, 3]. However, physical constraints of CMOS circuit limit the applicability of having voltage varying continuously and instantaneously. Instead, it is more practical to make multiple discrete voltages simultaneously available for the system. Many commercial high-performance microprocessors, such as Transmeta's Crusoe [4], AMD's Athlon 4 [5], Intel's XScale [6], and some DSP (digital signal processing) cores developed in Bell Labs all support voltage scaling for low power.

Most existing work on multiple voltage DVS systems assumes that the voltage set-up, which includes the number of voltage levels and the voltage value at each level, is given a priori and focuses on developing the voltage scheduling algorithms to minimize system's energy consumption [7, 8, 9, 10, 11]. However, for multiple voltage DVS systems, the energy consumption depends on not only the scheduler but also the voltage set-up. To the best of our knowledge, how to set up the voltages has been discussed in the following contexts: Chen and Sarrafzadeh [12, 13] studied the power minimization problem on dual-voltage system at gate level, where 5.0V was used as the high voltage and different voltages from 2.0V to 4.2V were used as the low voltage. They suggested that the voltages should be chosen carefully based on the slack distribution of the circuits. Qu and Potkonjak [9] gave analytical solutions on how to build energy efficient communication pipelines under latency constraints by voltage scaling and careful packet fragmentation, where each pipeline stage receives one fixed voltage. Dhar and Maksimovic [14] considered the design of finite impulse response filters and applied Lagrangian method to find the 2N+1 voltages for power minimization, where N is the order of the filter.

In this paper, we consider the following voltage set-up problem at the application level: how to determine 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? The differences between our work and the ones mentioned above are: 1) we do voltage scaling at the application level, not the gate level, 2)

we determine the voltage values for any number of voltages, not only for dual-voltage or levels tightly bounded to the applications, and 3) we also find the best number of voltage levels.

We first use an example to show multiple-voltage system's energy efficiency and the importance of the voltage set-up. Suppose that a system periodically executes one application with period equals to 8. The application's possible execution times, at the reference voltage 3.3V, are 6, 4, 3, and 2 that occur with probabilities 0.05, 0.20, 0.45, and 0.30 respectively. The application has a deadline that equals to its period. Table 1 gives the average energy consumption per iteration when this application is executed by systems with different voltage set-ups, where the energy is computed based on the optimal voltage scaling strategies reported in [3] and [7] and is normalized to the average energy consumption per iteration at supply voltage $V_{dd}(ref) = 3.3V$ and threshold voltage $V_{th} = 0.5V$ [1].

Table 1: The average energy consumption per iteration on different systems. (1: the reference fixed voltage system; 2: the best fixed voltage system; 3-6: dual-voltage systems with different voltage set-ups.

Set-up	1	2	3	4	5	6
V_{high}	3.3	2.7	3.3	3.0	3.0	2.7
V_{low}	–	–	1.0	1.0	2.0	1.8
Energy	1	0.67	0.83	0.70	0.43	0.38

[1] We describe how Table 1 is built.

The average energy consumption per iteration for this application can be expressed as $\sum_{i=1}^{4} p_i \cdot E_i$, where p_i is the probability that the application requests an execution time (workload) e_i and E_i is the minimal energy consumption that the system completes such workload based on the optimal voltage scaling strategies as we will explain below [3, 7].

First, at the reference voltage $V_{dd}(ref) = 3.3V$, the average energy consumption per iteration is $E_{V_{high}=3.3V} = P(ref) \cdot \sum_{i=1}^{4} p_i \cdot e_i = 3.05P(ref)$, where $P(ref)$ is the power consumption and the sum gives the average execution time per iteration. Note that we assume the system shuts down to conserve energy when the current iteration is complete. Otherwise, the energy consumption becomes $8P(ref)$ for the always-on system.

Second, the worst case execution time (WCET) 6 is actually less than the deadline 8, we can then utilize this slack to reduce energy by scaling voltage down. Based on Equation (1) on page 8, one can compute that the lowest voltage to complete the workload that requires execution time 6 at the reference voltage at time 8 is roughly $2.7V$. At this supply voltage, the average energy consumption per iteration will be $0.67E_{V_{high}=3.3V}$. Note that system with voltage lower than $2.7V$ will miss the deadline should the WCET occurs. Therefore, this gives us the minimal energy for fixed-voltage systems.

The rest of Table 1 gives energy consumption of dual-voltage systems. On such system, if the low voltage V_{low} provides sufficient fast speed to complete the application by deadline, the system will operate at V_{low} and shut down upon completion. Otherwise, the system will use V_{low} for some time to complete the workload that requires execution time t_i at the reference voltage before scales up to V_{high} so the computation can be completed on the deadline. t_i can be conveniently calculated from the following equation $\frac{V_{high}}{(V_{high}-0.5)^2}\frac{(3.3-0.5)^2}{3.3}(e_i - t_i) + \frac{V_{low}}{(V_{low}-0.5)^2}\frac{(3.3-0.5)^2}{3.3}t_i = 8$, where e_i is the execution time of the application. This allows us to compute the time that the system is on V_{low} and V_{high} and eventually the energy consumption (see, for example, [3] and [7]).

We have two interesting observations from Table 1:

- Multiple-voltage systems in general save more energy over fixed-voltage systems. For example, the voltage set-ups (V_{high}=3.0V, V_{low}=2.0V) and (V_{high}=2.7V, V_{low}=1.8V) save more than 35% and 43% energy respectively over the best fixed-voltage system with the lowest voltage 2.7V without any deadline missing.

- Different voltage set-ups result in significantly different energy reduction as we can see from the last four columns. Moreover, if not set properly, set-ups 3 (V_{high}=3.3V, V_{low}=1.0V) and 4 (V_{high}=3.0V, V_{low}=1.0V) for example, the multiple-voltage system may consume more energy than the best fixed-voltage system, the one with a fixed 2.7V supply voltage in this case.

We formulate and provide practical solutions to the voltage set-up problem that seeks the most energy efficient voltage setting for the design of multiple-voltage DVS systems. This work is a novel extension under the DVS research framework. Our main contributions include: (1) analytical solutions and a linear search algorithm for dual-voltage DVS systems; and (2) an iterative approach and an approximation method for the general multiple-voltage DVS systems. These results can be used to guide system design as we show by simulation. Surprisingly, our results show that the 3- or 4-voltage system can actually be (almost) as energy-efficient as the ideal system that varies voltage arbitrarily. Finally, we mention that although we restrict most of our discussion to dynamic power reduction (we do so for the simplicity to explain our approaches and also because that dynamic power still dominates in embedded system design such as DSP systems), our problem formulation and proposed approaches can integrate both leakage power/energy model and the overhead on voltage scaling.

The remainder of this paper is organized as follows. In the next section, we survey the related work on DVS for low power. We then formulate the voltage set-up problem and present the solutions in Section 3 and Section 4 respectively. Validation of our solutions and experimental results are reported in Section 5. We conclude the paper in Section 6.

2 Related Work

We restrict our survey to the study of multiple-voltage DVS systems. For the discussion on ideal voltage scaling systems and design/implementation issues on DVS systems, one can find excellent surveys in [2, 3, 15, 16, 17].

Early research on multiple-voltage DVS systems focused on voltage scheduling at behavioral level, typically on data flow graphs to exploit the parallelism among all of the operations. Specifically, operations on the critical path are operated at the reference voltage to keep the required throughput, but operations off the critical path will be executed at reduced voltages to save power and energy [12, 18, 19, 20, 21]. Raje and Sarrafzadeh [21] first proposed a multiple voltage scheduling algorithm to assign voltage level to each operation in a data flow graph to minimize power consumption with a given computation time constraint. Dual-voltage (5.0V and 3.0V) and three-voltage (5.0V, 3.0V, and 2.4V) were used for experimental purpose. Chang and Pedram [18] presented a dynamic programming based algorithm extending this to more general cases (such as cyclic graphs, throughput constraints). Four voltages (5.0V, 3.3V, 2.4V, and 1.5V) were used in the simulation for no specific reasons. Johnson and Roy [19] proposed a datapath scheduling algorithm that iteratively reduces the operating voltage until no schedule slack remains. Chen and Sarrafzadeh [12] related the voltage scaling (VS) power minimization problem on dual-voltage system to the maximal weighted independent set problem, which is polynomial solvable on transitive graph. They then developed a provably good algorithm to reduce system's power consumption. In their simulation, 5.0V was used as the high voltage while different voltages from 2.0V to 4.2V were used as the low voltage.

The study of multiple-voltage DVS system at high level focused on how to assign voltage to individual task in order to reduce energy consumption. Ishihara and Yasuura [7] showed that energy is minimized only when at most two voltages are applied to a single task. They formulated the voltage scheduling problem as an integer linear programming problem and relied on solving such problem to obtain the voltage for each task. Quan and Hu [10] studied the problem of determining the optimal voltage schedule for a real-time system with fixed-priority jobs implemented on multiple VS systems. Their approach was based on an integer programming formulation, which can be efficiently solved. Manzak and Chakrabarti [22] proposed periodic and aperiodic task scheduling algorithms for energy minimization on VS systems. Pillai and Shin [23] presented a class of algorithms that modify the operating system's real-time scheduler and task management service to provide significant energy savings while maintaining real-time deadline guarantee. Most recently, Hua *et al.* [24] have proposed some scheduling strategies for a multiple-voltage system in order to reduce the system's energy consumption while providing non-perfect completion ratio guarantee statistically. Some tasks are intentionally dropped according to their on-line scheduling algorithm to conserve energy.

For dependent tasks on multiprocessor systems, many approaches have been developed to reduce energy consumption by VS with multiple voltages. Luo and Jha [25] presented a power-conscious algorithm for jointly scheduling multi-rate periodic task graphs and aperiodic tasks in real-time multiprocessor embedded systems. Their goals were to improve soft aperiodic task's response time and to reduce overall energy consumption. They also proposed static battery-aware scheduling algorithms in battery-powered distributed real-time embedded systems to increase the battery lifespan [26]. Schmitz and Al-Hashimi [27] developed an efficient algorithm for voltage scaling of a distributed embedded system considering variations in the power dissipation among processes and inter process communications. They further investigated VS processing elements' power variations, dependent on the executed tasks, during the synthesis of distributed embedded systems and its impact on the energy savings.

3 The Voltage Set-up Problem

The *voltage set-up problem* seeks for the most energy efficient way to implement a multiple voltage system that executes a given set of applications or a single application with execution time uncertainties. More specific, we want to determine how many different voltage levels and what is the value at each level so we can complete all the applications with the least energy consumption. Before we elaborate our problem formulation, we mention that certain knowledge about the applications being executed on the system (such as their execution time information) is required. Therefore, the proposed problem and solutions target the embedded system design and their impact to general purpose processor will be limited.

Each application has a (or a set of) specific amount of computation requirement [28], or equivalently, a certain amount of CPU time to complete the computation before a deadline constraint. This situation occurs in systems (such as DSP systems) that run a single application characterized by the repetitive processing on periodically arriving input samples and each iteration must be completed during its period. It may also happen in systems that assign a fixed amount of time to each of the applications. Another example is an event-triggered system, in which the application requests arrive with a fixed deadline and the time between any two consecutive requests is not less than the deadline. For such system, there is typically one application at a time and the system executes the computation in the non-preemptive way. However, the application's execution time can vary dramatically due to a number of factors such as data locality and correlation, I/D cache misses, or pipeline stalls etc. However, it is possible to obtain the

application execution time distribution from system's detailed timing information or from simulation on the target hardware [29]. For example, input sample statistics and throughput constraints can be used to model the execution time distribution for many DSP applications such as MPEG decoding. We adopt the assumption in [15] that the real execution time can be known a priori, which is possible particularly in application specific DSP systems. In sum, we assume that the applications are characterized by triples $< e_i, d_i, p_i > (i = 1, 2, \cdots, n)$, where e_i is the execution time, d_i is the deadline, and p_i is the probability that the system executes the application. We mention that e_i's can be the execution times for different applications or the different execution times for the same application. In the latter, these e_i's will have the same deadline.

Unlike the DVS system that uses voltage converter to control the operating voltage at run-time [2], we consider DVS systems with multiple levels of supply voltage and corresponding threshold voltage physically implemented on the chip. We will discuss the advantages of such systems and their associated overhead later in this section. Here we only mention that this is feasible by, for example, using multiple voltage regulators each of which regulates a specific voltage with a given clock frequency. The operating system can control the clock frequency at run time by writing to a register in the system control state the same as in [2]. The voltage set-up problem and its exact solution depend on how we model the multiple-voltage system's voltage, delay, power, and energy consumption. However, the problem formulation we give at the beginning of this section is independent of such models and so are our proposed approaches and most results (as long as the convexity property of the power/energy vs. voltage/speed function holds). For simplicity, we adopt the following models. We will point out whenever an approach and/or result requires some specific feature of these models and explain whether the approach/result is valid for other models. *Suppose that at the reference (highest) supply voltage $V_{dd}(ref)$ and threshold voltage $V_{th}(ref)$, the processor's power dissipation is $P(ref)$ and the execution time is $T(ref)$ for a fixed amount of computation, then at supply voltage V_{dd} and threshold voltage V_{th}, to accumulate the same amount of computation, execution time T, power dissipation P, and energy consumption E are given by* [30]:

$$T = \frac{V_{dd}}{(V_{dd} - V_{th})^2} \frac{(V_{dd}(ref) - V_{th}(ref))^2}{V_{dd}(ref)} T(ref) \quad (1)$$

$$P = \frac{V_{dd}(V_{dd} - V_{th})^2}{V_{dd}(ref)(V_{dd}(ref) - V_{th}(ref))^2} P(ref) \quad (2)$$

$$E = P \cdot T = \frac{V_{dd}^2}{V_{dd}(ref)^2} P(ref) T(ref) \quad (3)$$

We now discuss the advantages of the multiple-voltage DVS systems and their associated overhead. Most DVS systems have either DC-DC converters to provide flexible operating voltage at run time. This inevitably introduces time and energy overhead, most notably the time for voltage to stabilize at the new level. Moreover, such overhead cannot be treated as constant because the *transient response time* and consequently the energy consumption for voltage switch depend on the difference between the source voltage and the desired voltage. Our multiple-voltage DVS system uses multiple on-chip linear voltage regulators, where each regulator is dedicated to generate one specific voltage and clock frequency. This eliminates the use of DC-DC converter or other auxiliary devices (such as buffers, delay line, and/or charge pump) to implement dynamic voltage scaling. Although we sacrifice the ability to generate voltage at any level, we have the time and energy overhead on voltage switch in a much more controllable manner. First, the delay overhead for each voltage switch will be a constant: one clock cycle for writing to the register plus the fixed transient response time (normally in a few cycles [33, 34]). Second, the power dissipation on the voltage regulators will be the sum of one regulator's dynamic power and the static power for other regulators. This is because that the system, when active, uses only one regulator at any time and can shut down the rest for energy conservation. Note that recent advances in linear voltage regulator design have led us to low dropout (LDO) regulators with high efficiency, low power, and low transient response time. The LDO regulator's dynamic power can be well below the mW mark[2] and its quiescent current is in the order of μA's [34]. Third, using multiple LDO regulators may not cause more area overhead than a DC-DC converter. This is because that we do not require the auxiliary devices to produce flexible voltage and, as we will show in the simulation, two or three regulators will be sufficient for energy efficiency for most application specific embedded systems.

Finally, we mention that the energy consumption in the voltage set-up problem formulation should include both dynamic and leakage energy consumption as well as the energy consumed on the voltage regulators. In the above multiple voltage DVS system, both dynamic and static power of the voltage regulators can be explicitly integrated into the voltage set-up problem. More specific, the voltage

[2]We are unable to find any direct reference for on-chip LDO regulator's dynamic power. However, [33] reports an adaptive voltage scaling controller that consists of a voltage regulator, a charge pump, a resettable delay-line, level shifters, and a clock generator among other devices. This controller consumes 2mW at 3V and 20MHz, with most energy dissipation spent on devices other than the regulator.

regulator that supports voltage v_i will have a constant power overhead $P_{regulator_i}$; and if it is used for a period of t, the energy overhead will be $P_{regulator_i} \cdot t$. We have mentioned that the quiescent current of the LDO regulator is in the order of μA's, where implies that its power overhead is in the order of μW's. Furthermore, the latest regulators enable shut down for energy efficiency. The circuit's leakage power dissipation depends on operating voltage and threshold voltage among many other factors. One can refer to, for example [35], for detailed leakage power model. We will not take leakage into the consideration when we solve the voltage set-up problem in the following section due to the complexity of leakage power model. However, for any given leakage power model, one can still apply any of the proposed numerical and approximation approaches either directly or after minor modification, which we will elaborate when these approaches are discussed.

4 Solving the Voltage Set-up Problem

In this section we first introduce three basic lemmas and then present the analytic solution and an exact approach for the dual-voltage DVS system. We also propose an iterative approach and a linear (to the number of voltages) approximation method for solving the problem in the general case. Finally we discuss how to find the best voltage set-up (both the number of voltage levels and the value of each voltage) in order to achieve the maximum energy saving.

We assume that the target multiple-voltage DVS system has m levels of supply voltage ($V_1 < V_2 < \cdots < V_m$) available. Suppose that the i-th application has deadline d_i and requests $e_i \leq d_i$ as execution time under the reference voltage $V_{dd}(ref)$. We define its ideal voltage V_i^0 to be the level at which the system will complete the workload e_i at d_i with minimum energy consumption [7]. From Equation (1), we can compute the value of V_i^0 (for a fixed threshold voltage) or determine the relationship between V_i^0 and its corresponding threshold voltage. When the leakage power/energy, time and energy overhead for voltage switch, and other model variations (for example, the expression $(V_{dd} - V_{th})^2$ in Equation (1) should be updated to $(V_{dd} - V_{th})^{1.2}$ for current technology; temperature should be integrated into the delay model) need to considered, one may not be able to solve Equation (1) to get the exact value of V_i^0. However, we can always find the best V_i^0 by solving equations (1)-(3) numerically.

Without loss of generality, we assume that $V_1^0 < V_2^0 < \cdots < V_n^0$ are the ideal voltages for the n applications characterized by $< e_i, d_i, p_i > (i = 1, 2, \cdots, n)$ and $V_1 < V_2 < \cdots < V_m$ are the m voltage

levels to be set up on the system. Any solution to the voltage set-up problem must satisfy the following lemmas:

Lemma 1: $V_m = V_n^0$.

Proof: If $V_m < V_n^0$, the system will not be able to complete the n-th application by its deadline.

If $V_m > V_n^0$, we consider a new voltage set-up where we replace V_m by $V_m' = V_n^0$. No deadline will be missed because $V_m' \geq V_i^0$. We only need to show that the new voltage set-up reduces energy consumption. It is well-known (see [3] or [7] for example) that on a multiple voltage system, the energy consumption for the i-th application is minimized if we use only two voltages V_j and V_{j+1}, which are immediate neighbors to V_i^0, such that $V_j < V_i^0 < V_{j+1}$. Therefore, the new voltage set-up will only affect the energy consumption for applications with ideal voltages higher than V_{m-1}. For such applications, the original set-up uses voltages V_{m-1} and V_m, while the new voltage set-up uses V_{m-1} and V_m' ($< V_m$). Due to the fact that the power/energy consumption is a convex function of the supply voltage, the energy consumption under the new voltage set-up will be less. ∎

Lemma 2: $V_1 \geq V_1^0$.

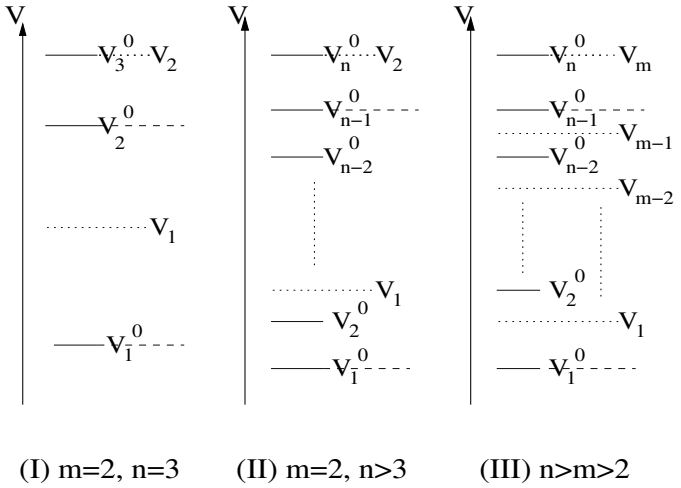
Proof: Similar to the proof of Lemma 1, if $V_1 < V_1^0$, we consider a new voltage set-up where we replace V_1 by $V_1' = V_1^0$. The new voltage set-up will only affect the energy consumption for applications with ideal voltages lower than V_2. For such applications, the energy consumption under the original voltage set-up that uses voltages V_1 and V_2 is more than that under the new voltage set-up, which uses V_1' ($> V_1$) and V_2. Therefore, setting up the lowest voltage V_1 lower than the lowest ideal voltage V_1^0 will not benefit any application. ∎

Lemma 3: There exists at most one $V_i \in (V_{k-1}^0, V_k^0]$ for any integer $k > 1$.

Proof: If there are two or more voltages in $(V_{k-1}^0, V_k^0]$, we replace all of them by two voltages: V_{k-1}^0 and V_k^0, while keeping other voltage levels unchanged. Suppose that this gives us a voltage set-up of $V_1 < \cdots < V_i < V_{k-1}^0 < V_k^0 < V_j \cdots < V_m$. Apparently that this will only impact the execution of application whose ideal voltage is between V_i and V_j. For the applications that have ideal voltages less than V_{k-1}^0, the new voltage set-up will use V_i and/or V_{k-1}^0, while they are previously executed with V_i and some voltage level higher than V_{k-1}^0. Due to the convexity, the new voltage set-up is more energy efficient. This is also true for applications whose ideal voltages are higher than V_k^0. Therefore, the most energy efficient voltage set-up cannot have two or more voltages in any interval $(V_{k-1}^0, V_k^0]$. ∎

We now show how to apply these lemmas to guide the multiple voltage design. Suppose that an application has five possible execution times, corresponding to five ideal voltages, 1.2V, 1.6V, 2.4V, 2.8V, and 3.2V. Lemma 1 says that the highest voltage must be set at 3.2V; Lemma 2 implies that the lowest voltage should not be lower than 1.2V; and Lemma 3 guarantees that any voltage set-up that has two or more voltages fall in the interval of (1.2,1.6], (1.6,2.4],(2.4,2.8], or (2.8,3.2] cannot be optimal. For example, none of the following set-ups will be optimal: $\{1.6, 3.3\}$, $\{1.2, 2.4, 3.0\}$, $\{1.1, 2.4, 3.2\}$, $\{1.8, 2.0, 2.8, 3.2\}$.

These lemmas not only identify non-optimal voltage set-ups, they are also fundamental for our proposed solutions to the voltage set-up problem. In the rest of this section, we first address the problem of how to set up m-voltage systems, where m is given, for application(s) with n distinct possible execution times. Figure 1 gives the details on how we approach the problem. We then discuss how to determine both the number of voltage levels m and the voltage of each level in order to achieve the maximum energy saving.



(I) m=2, n=3 (II) m=2, n>3 (III) n>m>2

Figure 1: Summary of voltage set-up solutions for m-voltage system with n applications. (V_i^0 is the ideal voltage for i-th application, $V_1^0 \leq V_2^0 \leq \cdots \leq V_n^0$; V_j is the j-th supply voltage and $V_1 < V_2 < \cdots < V_m$.)

4.1 Case I: Dual Voltages Three Applications (m=2 and n=3)

We consider a dual-voltage system (m=2) with three applications (n=3). For simplicity, we assume that each application has one fixed execution time. (This does not lose the generality because one can treat an application with k different possible execution times as k applications.) Clearly, this is the simplest

non-trivial case because one can simply use all the ideal voltages if $m \geq n$.

Let $V_1 < V_2$ be the system's two voltages and $V_1^0 \leq V_2^0 \leq V_3^0$ be the ideal voltages for three applications characterized by $< e_1, d_1, p_1 >$, $< e_2, d_2, p_2 >$, and $< e_3, d_3, p_3 >$. From the above lemmas, we know that $V_2 = V_3^0$ and $V_1 \in [V_1^0, V_2^0]$ (because $V_2 \in (V_2^0, V_3^0]$). Under such voltage set-up,

• The third application will be executed at V_2 and completed at its deadline d_3;

• For the second application, the system runs at the lower voltage V_1 for a certain amount of time to save energy before speeds up to V_2 to meet its deadline d_2;

• The first application will be executed at V_1 till its completion.

Therefore, the system's expected energy consumption can be expressed as:

$$E = \frac{P(ref)}{V_{dd}(ref)^2}[p_3 V_2^2 e_3 + p_2(V_2^2(e_2 - t_2) + V_1^2 t_2) + p_1 V_1^2 e_1] \tag{4}$$

where t_2 satisfies

$$\frac{V_2}{(V_2 - V_{th2})^2}\frac{(V_{dd}(ref) - V_{th}(ref))^2}{V_{dd}(ref)}(e_2 - t_2) + \frac{V_1}{(V_1 - V_{th1})^2}\frac{(V_{dd}(ref) - V_{th}(ref))^2}{V_{dd}(ref)}t_2 = d_2 \tag{5}$$

The physical meaning of t_2 is as follows. Suppose that W is the portion of the workload from the second application being executed at voltage V_1. t_2 is the time required to complete the same workload W at the reference voltage.

If V_1 and V_2 are associated with different threshold voltages V_{th1} and V_{th2}, we can prove that analytical solutions do not exist and the problem can only be solved numerically. However, if the threshold voltage remains the same, i.e. $V_{th1} = V_{th2} = V_{th}$, we can apply the first order condition and conclude that energy consumption (4) is minimized only if V_1 is the solution to the following equation:

$$\begin{aligned}(-2V_2 p_2 d_{2p} + 2V_2^2 p_1 e_1)V_1^3 + [(2V_2 V_{th} - V_2^2 + 3V_{th}^2)p_2 d_{2p} - 4V_2 V_{th}^2 p_1 e_1]V_1^2 \\ +[(-4V_{th}^3 + 2V_2 V_{th}^2)p_2 d_{2p} + 2V_{th}^4 p_1 e_1]V_1 + p_2 d_{2p} V_{th}^2 (V_2 - V_{th})^2 = 0\end{aligned} \tag{6}$$

where

$$d_{2p} = \frac{d_2(V_2 - V_{th})^2 V_{dd}(ref)}{(V_{dd}(ref) - V_{th})^2} - V_2 e_2 \tag{7}$$

The cubic equation (6) can be solved analytically and we conclude

Theorem 1. Analytical optimal solution exists for Case I with fixed threshold voltage.

4.2 Case II: Dual Voltages Multiple Applications (m=2 and n>3)

In this case, we know that $V_2 = V_n^0$ and $V_1 \in [V_1^0, V_{n-1}^0]$.

• The n-th application will be executed at V_2 to its completion;

• For applications with ideal voltages larger than V_1, both voltages will be used to meet the deadlines and save energy;

• For applications with ideal voltages less than V_1, only V_1 will be used as it is sufficiently fast to finish these applications earlier than their deadlines.

We seek for V_1 that minimizes the total energy consumption and meets all applications' deadlines. These two conditions can be expressed as:

$$E = \sum_{i=1}^{n} \frac{P(ref)p_i}{V_{dd}(ref)^2}[V_2^2(e_i - t_i) + V_1^2 t_i] \tag{8}$$

$$\frac{V_2}{(V_2 - V_{th2})^2}\frac{(V_{dd}(ref) - V_{th}(ref))^2}{V_{dd}(ref)}(e_i - t_i) + \frac{V_1}{(V_1 - V_{th1})^2}\frac{(V_{dd}(ref) - V_{th}(ref))^2}{V_{dd}(ref)}t_i \leq d_i \tag{9}$$

where t_i is defined the same as t_2 in equation (4).

These conditions are similar to equations (4) and (5) in Case I except that (9) imposes a set of nonlinear inequality constraints. It is well-known in the context of nonlinear programming that this makes the problem difficult to solve [31].

Figure 2 depicts an optimal algorithm with linear complexity, O(n), for the problem in this case. Assuming that $V_1 \in [V_{k-1}^0, V_k^0]$, we can remove the inequality constraints in (9). Specifically, for applications $k, \cdots, n$, deadlines will be met exactly for energy reduction (step 4); for the other applications, their deadlines will be satisfied automatically because V_1 is higher than their ideal voltages (step 5). Now this becomes the same problem as Case I and we can apply Theorem 1 to solve it optimally (step 6). Voltage V_1 that satisfies (8) and (9) must be in one of the above intervals, and we will find it when we visit that interval in step 3.

4.3 Case III: Multiple Voltages Multiple Applications (m>2)

Even when there are more than two voltages available, the system will still use at most two voltages to execute each application [7]. Define $\delta_{ij} = 1$ if voltage V_j is used during the execution of the i-th application and $\delta_{ij} = 0$ otherwise. Similar to t_2 defined in equation (5), define t_{ij} be the required execution

Input: n applications $\{< e_i, d_i, p_i >: i = 1, 2, \cdots, n\}$ with their corresponding ideal voltages $V_1^0 \leq V_2^0 \leq \cdots \leq V_n^0$.
Output: V_1 and V_2 that minimize (8) and satisfy (9).
Algorithm:
1. $V_2 = V_n^0$;
2. for each $k = 2, 3, \cdots, n-1$
3. { assume $V_1 \in [V_{k-1}^0, V_k^0]$;
4. replace " $\leq$ " by " $=$ " for $i = k, k+1, \cdots, n$ in (9);
5. delete the rest of the inequalities in (9);
6. solve the problem as in Case I;
7. let $V_{1,k}$ be the voltage and E_k be the energy;
8. }
9. report the voltage $V_{1,k}$ that has the least E_k as V_1.

Figure 2: Voltage set-up algorithm for the case of m=2, $n \geq 3$.

time of the i-th application under the reference voltage to finish the same portion of computation that is done with V_j. We then can formulate this general voltage set-up problem as a nonlinear programming problem in Figure 3.

$$
\begin{array}{rl}
\textbf{Find} & V_1, V_2, \cdots, V_m \\
\textbf{Minimize} & E = \frac{P(ref)}{V_{dd}(ref)^2} \sum_{i=1}^{n} p_i \sum_{j=1}^{m} V_j^2 \delta_{ij} t_{ij} \qquad (10) \\
\textbf{Subject to} & t_{ij} \geq 0, \\
& V_j > 0, \\
& \sum_{j=1}^{m} \delta_{ij} \leq 2, \ \ \delta_{ij} \text{ is 0 or 1}, \\
& \sum_{j=1}^{m} t_{ij} = e_i, \\
& \sum_{j=1}^{m} \frac{V_j}{(V_j - V_{thj})^2} \frac{(V_{dd}(ref) - V_{th}(ref))^2}{V_{dd}(ref)} t_{ij} \leq d_i.
\end{array}
$$

Figure 3: General voltage set-up problem as a nonlinear programming problem for the case of $m > 2$.

As analytic solutions for this general case do not exist, numerical approaches can be used to exhaustively search for the solution to this nonlinear programming problem. We can further speed up the search process by eliminating all the voltage set-ups that have two or more voltages between two consecutive ideal voltages (Lemma 3). However, this exhaustive search will still be expensive particularly when m is large. We thus propose two heuristics, an iterative approach and an approximation method to efficiently search for the solution based on the convexity of the energy function.

An Iterative Approach:

• Start with the single voltage system with voltage $V_{1,1} = V_n^0$, at which the system has the least energy consumption;

• Apply the algorithm in Figure 2 to solve for $V_{2,1}$ and $V_{2,2}$, the best voltage set-up for dual-voltage system;

• For k-voltage (k≥3) systems repetitively do the following: let $V_{k,k} = V_{k-1,k-1}$, search $V_{k,i}$ between $V_{k-1,i-1}$ and $V_{k-1,i}$ for the most energy efficient set-up such that $V_1^0 \leq V_{k,1} \leq V_{k-1,1} \leq V_{k,2} \leq V_{k-1,2} \leq \cdots \leq V_{k,k-1} \leq V_{k-1,k-1} = V_{k,k} = V_n^0$.

Note that if we know the energy overhead E_k to support k voltages on the system, we can add it to the energy consumption of the best k-voltage system and determine how many voltages we should implement on the system.

An Approximation Method:

• Start with a random m-voltage set-up;

• Fix the (m-1) high voltages and compute the lowest voltage V_1 by a procedure similar to the algorithm in Figure 2;

• Determine V_2 by fixing the obtained V_1 and the other (m-2) high voltages;

• Continue till after we update the value of V_{m-1}, the second highest voltage; (This is one round of updating.)

• If there is energy improvement, go back to the second step with this new obtained voltage set-up;

• Report the optimal voltage set-up.

This method is based on the convexity of the energy function. Although we cannot guarantee how many rounds we need to update the voltage set-ups to reach the optimal values, simulation shows that the voltage set-up converges to the optimal solution (calculated by numerical method) after $2 \sim 3$ rounds.

Finally, we mention that these two techniques and Lemmas 1∼3 can be combined together to solve the problem efficiently.

4.4 Finding the Best Voltage Set-up

Once m, the number of voltages on the system, is fixed, we now know how to find the most energy-efficient voltage set-up from the above discussion. The corresponding average energy consumption

per execution can be conveniently obtained from Equation (10). If we ignore the hardware overhead (e.g., the area and power on the voltage regulators or DC-DC converters) to support multiple levels of voltages, then clearly the more voltages we have, the less energy will be consumed. A simple reason is that m-voltage systems can also be treated as (m+1)-voltage systems where two of the (m+1) voltages have the same value.

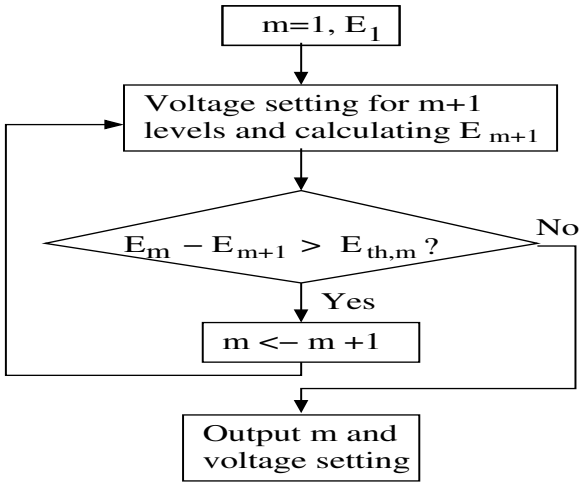


Figure 4: Flow to find the best voltage set-up.

However, supporting multiple voltages on the same system does require additional hardware and will cause area, delay, and also power penalties. It becomes important to investigate the trade-off between more voltage levels and the overhead they introduce. Figure 4 shows a scheme on how to find the best voltage set-up, i.e. the optimal number of voltage levels and the value of each level, to minimize the energy consumption, assuming that there is a threshold energy cost $E_{th,m}$. If the energy saving by including the (m+1)st voltage, $E_m - E_{m+1}$, is higher than this threshold $E_{th,m}$, then (m+1)-voltage system is more energy efficient than any m-voltage systems. Otherwise, it is not worth going to (m+1) voltages and we report the best m-voltage set-up as the overall optimal solution. Although it is relatively easy to measured the hardware cost for using one voltage regulator, it might not be easy to determine the threshold energy cost $E_{th,m}$ due to the design complexity of adding one more regulator. We mention that in general this threshold energy cost increases as one attempts to implement more and more different voltages on the same system. However, as we will show in the next section, simulation results show that normally two or three voltage levels will be sufficiently energy efficient.

5 Simulation Results

There are two goals in our simulation: demonstrating the importance of voltage set-up problem and validating the efficiency and accuracy of our proposed approaches to solve this problem. For all the simulations, we solve the dual-voltage case by the algorithm presented in Figure 2. The 3-voltage and 4-voltage solutions are obtained by the approximation method proposed in the previous section. We also list the energy consumption of the fixed-voltage system and the ideal DVS system for comparison. Note that the energy consumption of the ideal DVS system, where we have the ideal voltage for each possible execution time, is the lower bound of the system energy consumption.

Table 2: Information of the applications and their optimal fixed voltage and dual-voltage setups. The energy is in the unit of the energy dissipation in one CPU unit at the reference voltage.

Application	Deadline	WCET	n	$E_{ref+s/d}$	v_{fixed}	E_{fixed}	dual-voltage	E_{dual}	E_{ideal}
DSC-7-7	18.74	16.51	16	13.20	3.01	10.98	(3.01,2.55)	8.11	8.08
DSC-7-8	16.12	14.75	4	13.16	3.09	11.56	(3.09,2.87)	9.86	9.85
meas	345.39	312.03	8	289.01	3.06	249.27	(3.06,2.86)	224.17	223.95
qmf4	214.20	207.09	4	172.09	3.22	163.76	(3.22,2.68)	129.11	128.37
sum1	105.32	92.35	4	88.40	3.00	73.05	(3.00,2.88)	68.74	68.73
Laplace	1837.35	1757.75	4	1490.96	3.19	1397.20	(3.19,2.77)	1117.10	1112.50
almu	85.29	77.61	4	72.86	3.08	63.50	(3.08,2.92)	58.10	58.09
karp10	670.22	631.25	8	602.25	3.16	551.70	(3.16,3.02)	516.10	515.97
FFT1	902.36	880.40	8	770.24	3.24	742.86	(3.24,2.92)	614.87	614.25
FFT2	670.19	581.93	16	498.40	2.98	406.14	(2.98,2.61)	330.82	329.15
TGFF1	3827.87	3764.70	25	3498.57	3.26	3414.00	(3.26,2.98)	3095.50	3077.10
TGFF2	6505.08	5544.24	15	4648.91	2.94	3689.40	(2.94,2.41)	3046.30	2969.20
TGFF3	7109.84	6868.46	17	6044.11	3.22	5745.30	(3.22,2.77)	4925.60	4830.10
TGFF4	8888.71	7689.07	17	6880.81	2.97	5577.10	(2.97,2.59)	4875.60	4816.90
TGFF5	12691.27	12103.11	24	10503.01	3.19	9797.40	(3.19,2.65)	8357.00	8127.70
TGFF6	14600.00	13724.15	18	11962.96	3.15	10928.00	(3.15,2.60)	9441.70	9211.30
TGFF7	16535.52	14837.42	14	12716.80	3.05	10862.00	(3.05,2.46)	9229.00	8992.50
TGFF8	14933.09	13405.49	11	12626.99	3.05	10792.00	(3.05,2.77)	9995.10	9955.90
TGFF9	20327.33	17088.68	25	14748.26	2.91	11479.00	(2.91,2.45)	9644.90	9481.90
TGFF10	20201.08	18767.40	12	17156.25	3.13	15407.00	(3.13,2.68)	13834.00	13666.00

We first consider using multiple voltage to serve a single application with moderate uncertainty in execution time. Table 2 shows the deadline, worst case execution time (WCET), and the number of

possible execution time (column 'n') for 10 real-life DSP applications and 10 random applications generated by the TGFF (task graph for free) package. *FFT1* and *FFT2* are two different implementations of the Fast Fourier Transform, *Laplace* is the Laplace transform, *qmf4* is a quadrature mirror filter bank, *karp10* is the Karplus-Strong music synthesis algorithm with 10 voices, *meas* is a measurement application, *sum1* is an upside down binary tree representing the sum of products computation, and a more detailed description of these applications can be found in [24]. For each application, we assign a discrete distribution for it execution time that includes multiple (range from 4 to 25 as shown in the fourth column in Table 2) execution times and the probability that they will occur[3].

We assume that the system consumes one unit of energy in one CPU unit at the reference voltage setting ($V_{dd} = 3.3V, V_{th} = 0.5V$). Therefore, a system that is always on will have the energy consumption equals to the application's deadline (the second column in Table 2). If the system is able to shut down on the completion of the application, an average of 18.7% energy saving can be achieved (the column $E_{ref+s/d}$ in Table 2). Knowing that the application's WCET is less than its deadline, we can reduce the voltage from the reference level to the level at which the WCET of the application can be completed exactly at deadline. Such voltage levels and the corresponding energy consumption (assuming system shut-down is allowed) are reported in the columns of v_{fixed} and E_{fixed}. This gives an average of 11.9% energy saving over $E_{ref+s/d}$. The next two columns reports the best dual-voltage system setups and the corresponding energy dissipation, 14.0% less E_{fixed} on average. Finally, the last column E_{ideal} is the lower bound on energy consumption where we assume that there are n levels of voltages, each corresponds to one possible execution time, where n is the number of different execution times in the fourth column.

Figure 5 illustrates the energy consumption, normalized to E_{ideal}, for each application by six different voltage setups: fixed reference voltage $3.3V$ only, $3.3V$ and shut-down, the best fixed voltage v_{fixed} in Table 2, the best dual voltages, the best three voltages, and the best four voltages. We have observed average energy savings in double digits from Table 2 when system shut-down, the best fixed voltage, and the second voltage are introduced respectively. However, the benefit of having the third and fourth voltage is not significant at all. The 3-voltage system saves an average of 0.67% energy over the dual-voltage system and the consumes only 0.18% more than the 4-voltage system. Note that the overhead

[3]We mention that for real life applications, such execution time distribution can be obtained by simulation and/or profiling.

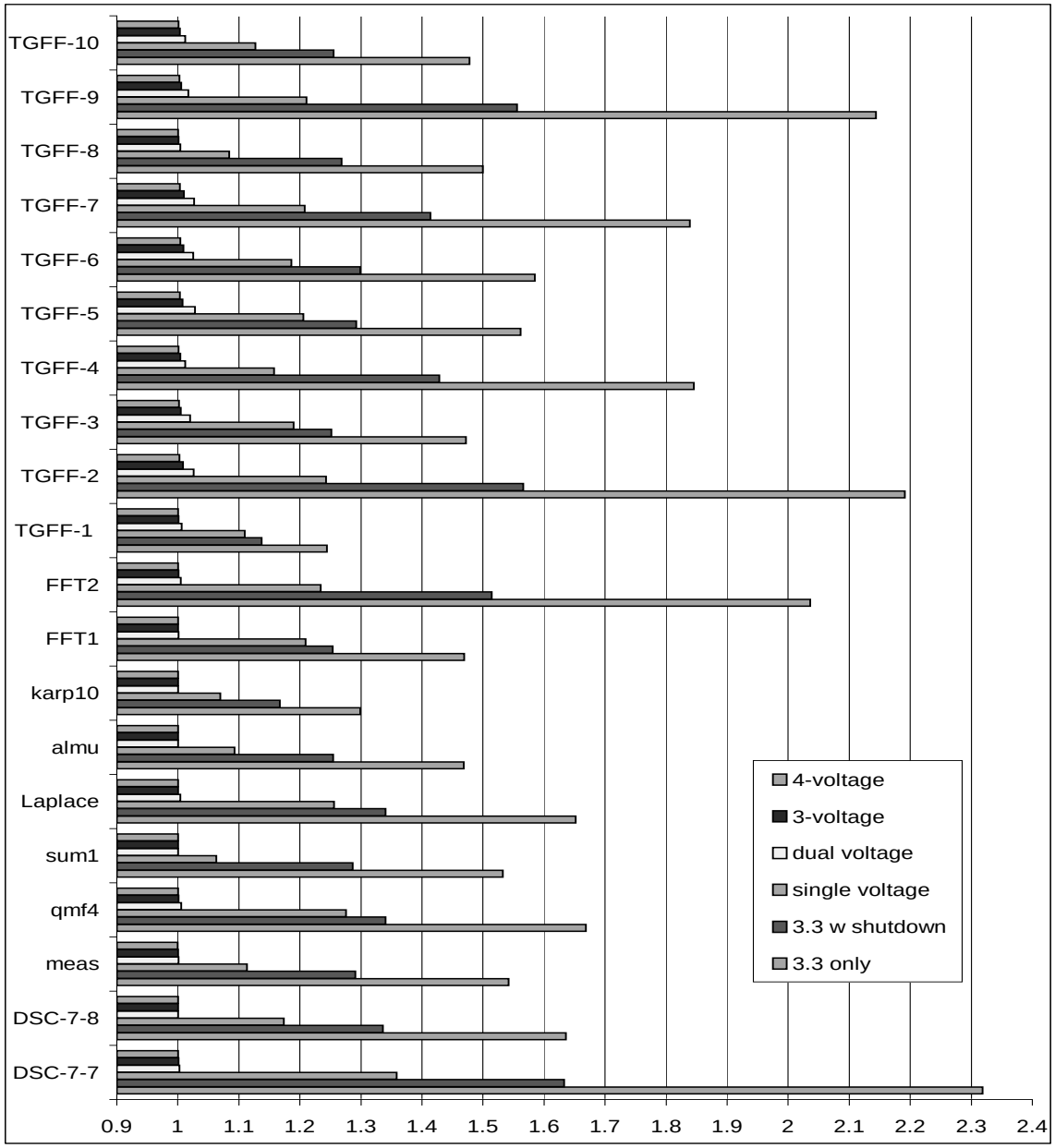


Figure 5: The average energy consumption of six different multiple voltage systems (normalized to E_{ideal}, the energy consumption on the ideal DVS system).

of voltage regulators is not considered in the above simulation. Therefore, we conclude that once the two voltage levels are designed carefully, the dual-voltage system is the most energy efficient. In fact, it energy consumption is within 1% of E_{ideal} on average.

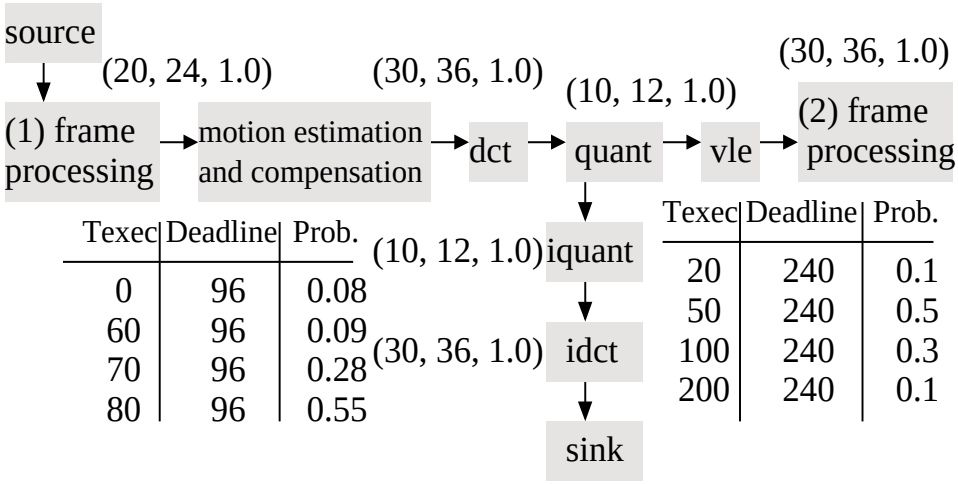


Figure 6: MPEG video encoder execution time distributions and corresponding deadlines in 10^4 cycles (redrawn from [32]).

Now we give a more detailed analysis of two simulations where the system executes multiple applications. The first one is on a set of two randomly generated abstract applications and the second is the MPEG video encoder. We will first report the energy efficiency of the fixed-, dual-, 3-, and 4-voltage systems and then discuss the accuracy of our proposed approaches by comparing our obtained results with those from exhaustive Matlab simulations in searching for the best voltage settings.

Figure 6 depicts the flow of MPEG encoding process as a set of subtasks. Next to each subtask, its <execution time T_{exec}, deadline, probability> triple is reported. The two tables in the figure correspond to the subtasks that do not have a deterministic execution time. The lower left table is for motion estimation and compensation and the other one is for vle (variable length encoding) [32]. The two ad-hoc applications A and B have deadlines of 10 and 8, respectively. Application A occurs 60% of the time, its execution time distribution is $\{(9, 0.03), (4, 0.18), (3, 0.39)\}$. Application B occurs 40% of the time, its execution time distribution is $\{(6, 0.04), (4, 0.10), (3, 0.12), (2, 0.14)\}$.

Table 3 reports the optimal voltage settings for multiple voltage systems and their average energy consumption. For the two ad hoc applications, multiple voltage DVS systems save significant amount of energy over the fixed-voltage system. The saving is more than 53% when we carefully choose the

Table 3: The optimal voltage set-ups and their corresponding average energy consumption per execution. (In the parenthesis of energy columns, "-" is the energy saving over the fixed voltage system, "+" is the "wasted" energy comparing to the ideal voltage system.)

DVS Systems	2-Application		MPEG Encoder	
	Voltages	Energy	Voltages	Energy
fixed-voltage	3.0564	2.9536 (+151.1%)	2.8934	26.7125 (+20.1%)
dual-voltage	3.0564 1.8124	1.3833 (-53.2%) (+17.6%)	2.8934 1.8511	23.1478 (-13.3%) (+4.0%)
3-voltage	3.0564 2.0688 1.5514	1.2337 (-58.2%) (+4.9%)	2.8934 1.8558 1.3031	22.4958 (-15.8%) (+1.1%)
4-voltage	3.0564 2.0768 1.8119 1.5509	1.2071 (-59.1%) (+2.6%)	2.8934 2.6374 1.8554 1.3031	22.3020 (-16.5%) (+0.2%)
ideal	–	1.1763	–	22.2506

second voltage on the dual-voltage system. When we move to 3-voltage and 4-voltage systems, we see the continuous increase in energy reduction, however, at a much slower pace. We have similar observations on the MPEG encoder example. Dual-voltage system has a notable 13% energy saving over the fixed-voltage system, which is much lower than that in the previous example. This is because that majority of the energy is consumed on the deterministic subtasks. However, multiple-voltage systems still successfully reduced the "wasted" energy, comparing to the ideal system, from more than 20% in the fixed voltage system to 4.0%, 1.1%, and 0.2%. In sum, we conclude that multiple voltage DVS systems are effective in energy reduction and can be very close to maximal energy saving by DVS technique with only a couple of different voltage levels if they are set up carefully.

Finally, to validate the correctness of our results, we use Matlab to simulate 100,000 iterations of each application under different voltage settings for dual-, 3-, and 4-voltage systems. For example, for dual-voltage systems, we set the high voltage V_2 to go from V_n^0 (the lowest voltage to complete the WCET, 3.0564V in for the two ad hoc application example and 2.8934V for the MPEG encoder example) to the reference voltage 3.3V, and the low voltage V_1 to go from 1.0V to 3.3V, both with an increment of 0.01V. In all the cases, such exhaustive search finds the same solution, within the precision of voltage increment 0.01V that we set, as we reported in Table 3. The Matlab plots, Figures 7 and 8,

depict the dual-voltage system's energy consumption with different voltage settings. In both figures, we see that the energy consumption is minimized at the same set-up as we obtained theoretically in Table 3 by our proposed algorithm.

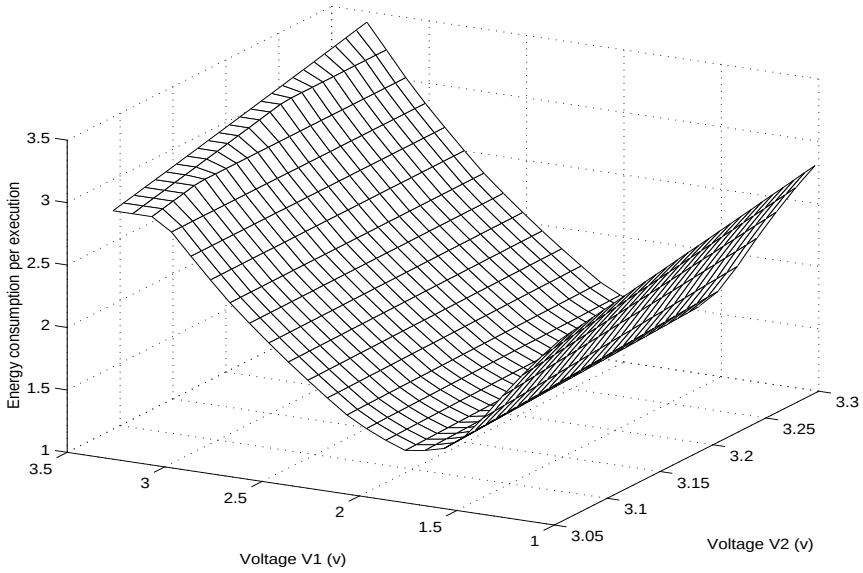


Figure 7: Dual-voltage system's average energy consumption for the two ad hoc applications with different voltage set-ups. The best voltage setting from our algorithm: (3.0564,1.8124).

6 Conclusion

We consider the voltage set-up problem for application specific multiple-voltage DVS system design. The problem seeks to determine the number of voltage levels and the voltage at each level to minimize the average energy consumption for a given set of applications. We give optimal solutions in analytic form for the dual-voltage system and develop two heuristics (an iterative approach and an approximation method) for the general case. The hardware overhead to supply multiple voltages, once obtained, can be conveniently integrated into our techniques to solve the voltage set-up problem. We apply our methods to the designs of an ad hoc application specific system and the MPEG video encoder, as well as DSP and other applications. Simulation results show the correctness and efficiency of our approaches. We also observe that multiple-voltage system, if the voltage levels are set properly, can indeed achieve energy reduction very close to the full potential of DVS.

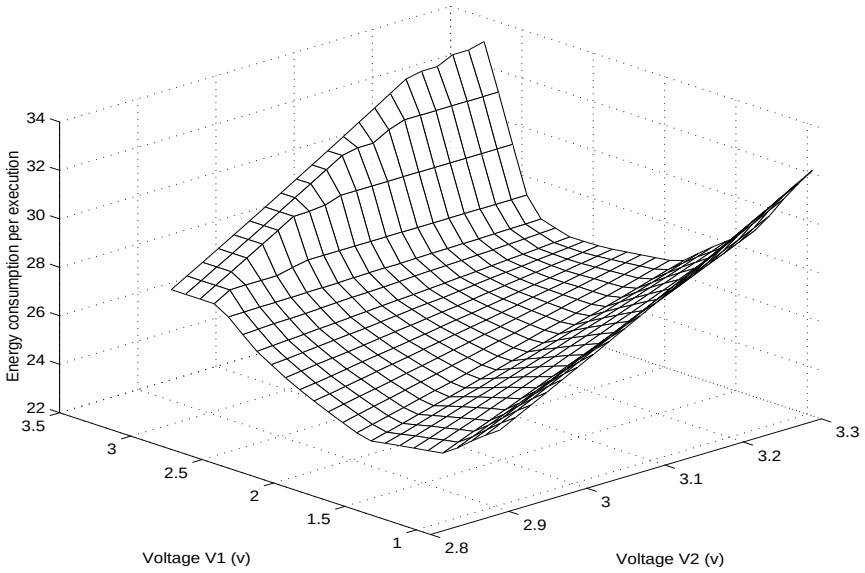


Figure 8: Dual-voltage system's average energy consumption for the MPEG encoder with different voltage set-ups. The best voltage setting from our algorithm: (2.8934,1.8511).

References

[1] "International technology roadmap for semiconductors," *http://public.itrs.net*, 2001.

[2] T. D. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A dynamic voltage scaled microprocessor system," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, Nov. 2000.

[3] G. Qu, "What is the limit of energy saving by dynamic voltage scaling?" *IEEE/ACM International Conference on Computer-Aided Design*, pp. 560–563, 2001.

[4] M. Fleischmann, "LongRun power management - dynamic power management for Crusoe processors," *Whitepaper, Transmeta Corp.*, 2001.

[5] I. Advanced Micro Devices, "AMD Athlon 4 processors, data sheet reference no.24319," 2001.

[6] Intel, "The Intel Xscale microarchitecture," *Technical Summary*, 2000.

[7] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," *International Symposium on Low Power Electronics and Design*, pp. 197–202, 1998.

[8] S. Lee and T. Sakurai, "Run-time voltage hopping for low-power real-time systems," *Design Automation Conference*, pp. 806–809, 2000.

[9] G. Qu and M. Potkonjak, "Techniques for energy-efficient communication pipeline design," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 5, pp. 542–549, October 2002.

[10] G. Quan and X. Hu, "Energy efficient fixed-priority scheduling for real-time systems on variable voltage processors," *Design Automation Conference*, pp. 828–833, 2001.

[11] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for reduced CPU energy," *Proc. USENIX Symposium on Operating Systems Design and Implementation*, pp. 13–23, 1994.

[12] C. Chen and M. Sarrafzadeh, "Provably good algorithm for low power consumption with dual supply voltages," *IEEE/ACM International Conference on Computer Aided Design*, pp. 76–79, 1999.

[13] C. Chen and M. Sarrafzadeh, "Power reduction by simultaneous voltage scaling and gate sizing," *Design Automation Conference*, pp. 333–338, 2000.

[14] S. Dhar and D. Maksimovic, "Low-power digital filtering using multiple voltage distribution and adaptive voltage scaling," *International Symposium on Low Power Electronics and Design*, pp. 207–209, 2000.

[15] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos, "Data driven signal processing: an approach for energy efficient computing," *International Symposium on Low Power Electronics and Design*, pp. 347–352, 1996.

[16] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. B. Srivastava, "Power minimization of variable voltage core-based systems," *Design Automation Conference*, pp. 176–181, 1998.

[17] N. Namgoong, M. Yu, and T. Meng, "A high efficiency variable-voltage CMOS dynamic DC-DC switching regulator," *Proc. IEEE International Solid-State Circuits Conference*, pp. 380–381, 1997.

[18] J.-M. Chang and M. Pedram, "Energy minimization using multiple supply voltages," *International Symposium on Low Power Electronics and Design*, pp. 157–162, 1996.

[19] M. C. Johnson and K. Roy, "Datapath scheduling with multiple supply voltages and level converters," *ACM Trans. on Design Automation of Electronic Systems*, vol. 2, no. 3, 1997.

[20] Y. R. Lin, C. T. Hwang, and A. C. H. Wu, "Scheduling techniques for variable voltage low power designs," *ACM Trans. on Design Automation of Electronic Systems*, vol. 2, no. 2, pp. 81–97, 1997.

[21] S. Raje and M. Sarrafzadeh, "Variable voltage scheduling," *International Symposium on Low Power Electronics and Design*, pp. 9–14, 1995.

[22] A. Manzak and C. Chankrabarti, "Variable voltage task scheduling algorithm for minimization energy," *International Symposium on Low Power Electronics and Design*, pp. 279–282, 2001.

[23] P. Pillai and G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," *Proc. of the 18th ACM Symposium on Operating Systems Principles*, pp. 89–102, 2001.

[24] S. Hua, G. Qu, and S. S. Bhattacharyya, "Energy reduction techniques for multimedia applications with tolerance to deadline misses," *Design Automation Conference*, pp. 131–136, 2003.

[25] J. Luo and N. K. Jha, "Power-conscious joint scheduling of periodic task graphs and aperiodic tasks in distributed real-time embedded systems," *IEEE/ACM International Conference on Computer Aided Design*, pp. 357–364, 2000.

[26] J. Luo and N. K. Jha, "Battery-aware static scheduling for distributed real-time embedded systems," *Design Automation Conference*, pp. 444–449, 2001.

[27] M. T. Schmitz and B. M. Al-Hashimi, "Low power process assignment for distributed embedded systems using dynamic voltage scaling," *Proceedings IEE Hardware-Software Co-Design*, pp. 7/1–7/4, 2000.

[28] D. Mosse, H. Aydin, B. Childers, and R. Melhem, "Compiler-assisted dynamic power-aware scheduling for real-time applications," *Workshop on Compiler and OS for Low Power*, 2000.

[29] T. S. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L.-C. Wu, and J. W.-S. Liu, "Probabilistic performance guarantee for real-time tasks with varying computation times," *Proc. Real-Time Technology and Applications Symposium*, pp. 164–173, 1995.

[30] A. P. Chandrakasan, S. Sheng, and R. W. Broderson, "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, April 1992.

[31] D. P. Bertsekas, "Nonlinear programming," *Athena Scientific*, 1999.

[32] A. Kalavade and P. Moghe, "A tool for performance estimation of networked embedded end-systems," *Design Automation Conference*, pp. 257–262, 1998.

[33] S. Dhar, D. Maksimovic, and B. Kranzen, "Closed-loop Adaptive Voltage Scaling Controller For StandardCell ASICs," *International Symposium on Low Power Electronics and Design*, pp. 103-107, 2002.

[34] C. Simpson, "Linear and Switching Voltage Regulator Fundamentals," National Semiconductor, *www.national.com/appinfo/power/files/f4.pdf*.

[35] N. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J. Hu, M.J. Irwin, M. Kandemir, and V. Narayanan, "Leakage Current: Moore's Law Meets Static Power," *Computer*, Vol. 36, No.12, pp. 68-75, 2003.