

Approaching the Maximum Energy Saving 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) is arguably the most effective energy reduction technique. The multiple-voltage DVS systems, which can operate only at pre-determined discrete voltages, are practical and have been well studied. However, one important unsolved problem is how many levels and at which values should voltages be implemented on a multiple-voltage DVS system to achieve the maximum energy saving. We refer this as the **voltage set-up problem**. 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. (3) We demonstrate how to apply the proposed algorithms on system design. (4) Interestingly, the experimental results suggest that the multiple-voltage DVS system, when the voltages are set up properly, can reach DVS technique's full potential in energy saving. Specifically, on the design of an ad hoc application-specific system and the design of the MPEG video encoder, we find that the best single-voltage systems consume 150% and 20% more energy than the tight theoretical lower bounds, respectively. However, our approach gives dual-, 3-, and 4-voltage DVS system settings that are only 17.6%, 4.9%, and 2.6% for the ad hoc system, and 4.0%, 1.1%, and 0.2% for the MPEG video encoder, over the same lower bounds.*

1. Introduction

Energy consumption has become a major design issue for modern embedded systems especially battery-operated portable devices. Although leakage power becomes more significant in such systems, dynamic power still dominates in designs with the current technology such as most DSP systems. Dynamic voltage scaling (DVS) technique varies the clock frequency and supply voltage according to workload at run-time to reduce dynamic power and save energy. It achieves the highest energy efficiency for time-varying computational loads if voltage can be varied arbitrarily [1, 12]. However, physical constraints of CMOS circuit limit the applicability of having voltage varying continuously. Instead, it is more practical to make multiple discrete voltages simultaneously available for the system. Transmeta's Crusoe, AMD's Mobile Athlon and Intel's XScale are all examples of advanced high-performance microprocessors that support DVS for low power.

Most existing work on multiple voltage DVS systems assume that voltage values are given and focus on how to utilize this option to minimize system's energy consumption [3, 7, 8, 11, 14]. For example, Raje and Sarrafzadeh [14] 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 3-voltage (5.0V, 3.0V, and 2.4V) were used for experimental pur-

pose. Chang and Pedram [3] 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. Hua *et al.* [7] proposed some scheduling strategies for a multiple-voltage system in order to reduce the system's energy consumption while providing non-perfect completion ratio guarantees statistically. The experimental results were reported when the four voltages were set at 3.3V, 2.6V, 1.9V, and 1.2V. To the best of our knowledge, how to select the multiple voltage levels has been discussed only in the following context: Chen and Sarrafzadeh [5] 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 [11] 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 [6] 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 application level voltage set-up problem: *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 application level instead of gate level, and 2) we solve the problem for any number of voltages instead of only dual voltages or levels tightly bounded to applications.

We formulate and provide practical solutions to the voltage set-up problem that seeks the most energy efficient voltage settings in 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, (2) a linear approximation algorithm for the general multiple-voltage DVS systems. When we use these results to guide multiple-voltage DVS system design, interestingly, (3) we find that the 3- or 4-voltage systems are (almost) as energy efficient as the ideal DVS system that can vary voltage arbitrarily.

The remainder of this paper is organized as follows. Section 2 formulates the voltage set-up problem. Section 3 explains our approaches. Validation of our methods and experimental results are presented in Section 4. We conclude the paper in Section 5.

2. Problem Formulation

We consider the design of an embedded system to perform a set of applications (or a single application with uncertainties in execution time). The system has multiple voltages to support DVS.

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

ICCAD'03, November 11-13, 2003, San Jose, California, USA.

Copyright 2003 ACM 1-58113-762-1/03/0011 ...\$5.00.

2.1 Application Model

Each application has a (or a set of) specific amount of computation requirement [13], 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.

Note that an 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. However, it is possible to obtain the application execution time distribution from the system's detailed timing information or from simulation on the target hardware [15]. We adopt the assumption in [4] that the real execution time can be known as a priori, which is possible particularly in application-specific DSP systems. We also assume that the i -th application is characterized by triples $\langle e_i, d_i, p_i \rangle$, 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 time for different applications or different execution time for the same application.

2.2 Multiple-Voltage DVS System Model

We assume that the target multiple-voltage DVS system has m levels of supply voltage ($V_1 < V_2 < \dots < V_m$), which are physically implemented on the chip. The system can also apply the "system shut-down mechanism" for energy efficiency. We further assume that the system can instantaneously switch its operating voltage from one level to another with the power dissipation $P \propto CV_{dd}^2 f$ and gate delay $d \propto V_{dd} / (V_{dd} - V_{th})^2$ at supply voltage V_{dd} and threshold voltage V_{th} [2]. There exists hardware overhead, such as the power dissipation on the voltage regulators, to support multiple voltages. However, this is a constant overhead independent of how we set up the m voltage levels.

2.3 The Voltage Set-up Problem

For a given set of applications characterized by triples of $\{(e_i, d_i, p_i) : i=1, 2, \dots, n\}$, determine (i) each voltage level for an m -voltage DVS system and (ii) determine m , the number of voltage level, to minimize the system's energy consumption without missing any deadlines.

The first part of the problem considers the case when the system has a fixed number of voltages and seeks for the most energy efficient voltage setting. The second part takes the overhead for supporting multiple voltages into consideration and seeks for the optimal number of voltages (and their values) to implement on the multiple-voltage DVS system for energy efficiency.

3. Solving the Voltage Set-up Problem

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 can compute its *ideal voltage* V_i^0 at which the system will complete the workload at d_i with minimum energy consumption [8,12].

Without loss of generality, we assume that $V_1^0 \leq V_2^0 \leq \dots \leq V_n^0$ are the ideal voltages for the n applications and $V_1 < V_2 < \dots < V_m$ 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: $V_m = V_n^0$ and $V_1 \geq V_1^0$.

Lemma 2: There exists at most one $V_i \in (V_{k-1}^0, V_k^0]$ for any $k > 1$.

These two lemmas not only identify non-optimal voltage settings, they are also fundamental for our proposed solutions to the voltage set-up problem. In the rest of this section, we first present the analytic solution and an exact approach for the dual-voltage DVS system. We then give an iterative approach and a linear approximation algorithm for the general case with m voltages, where m is given, and the applications have n distinct possible execution time. Finally we discuss how to determine m , the number of voltage levels, in order to achieve the maximum energy saving.

3.1 Case I: $m=2$ and $n=3$

We first consider a dual-voltage system ($m=2$) with three applications ($n=3$). For simplicity, we assume that each application has one fixed execution time. 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) , (e_3, d_3, p_3) . If the threshold voltage remains the same, we have:

Theorem 1: Analytical optimal solution exists for Case I with fixed threshold voltage. Specifically, the system's energy consumption is minimized when $V_2 = V_3^0$ and V_1 is the solution to the following cubic equation:

$$\begin{aligned} & (-2V_2 p_2 d_{2p} + 2V_2^2 p_1 e_1) V_1^3 + [(2V_2 V_{th} - V_2^2 + 3V_{th}^2) p_2 d_{2p} - 4V_2 V_{th}^2 p_1 e_1] V_1^2 \\ & + [(-4V_{th}^3 + 2V_2 V_{th}^2) p_2 d_{2p} + 2V_{th}^4 p_1 e_1] V_1 + p_2 d_{2p} V_{th}^2 (V_2 - V_{th})^2 = 0 \end{aligned} \quad (1)$$

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

3.2 Case II: $m=2$ and $n > 3$

In this case, from Lemma 1 and 2 we know that $V_2 = V_n^0$ and

Input: n applications $\{(e_i, d_i, p_i) : i=1, 2, \dots, n\}$ with their corresponding ideal voltages $V_1^0 \leq V_2^0 \leq \dots \leq V_n^0$.

Output: V_1 and V_2 that minimize the energy consumption.

Algorithm:

1. $V_2 = V_n^0$;
2. for each $k = 2, 3, \dots, n-1$
3. { assume $V_1 \in [V_{k-1}^0, V_k^0]$;
4. solve the cubic equation (1) by replacing $p_1 e_1$ with $\sum_{i=1}^{k-1} p_i e_i$ and $p_2 d_{2p}$ with $\sum_{i=k}^{n-1} p_i d_{ip}$,
where $d_{ip} = \frac{d_i (V_2 - V_{th})^2 V_{dd}(ref)}{(V_{dd}(ref) - V_{th})^2} - V_2 e_i$;
5. let $V_{1,k}$ be the optimal voltage in that interval and E_k be the energy;
6. }
7. report the voltage $V_{1,k}$ that has the least E_k as V_1 .

Figure 1. Voltage set-up algorithm for Case II $m=2, n \geq 3$.

$V_i \in [V_i^0, V_{n-1}^0]$. We seek for V_1 that minimizes the total energy consumption and meets all applications' deadlines. Figure 1 depicts an $O(n)$ algorithm that solves this problem optimally. The optimal V_1 that minimizes the energy consumption resides in one of the interval $[V_{k-1}^0, V_k^0]$. Assuming that $V_i \in [V_{k-1}^0, V_k^0]$, the problem becomes to the same as Case I so we can solve it optimally (step 4) by Theorem 1. The loop from step 3 to step 6 finds the best V_1 .

3.3 Case III: $m > 2$

When there are more than two voltages available, the system still uses at most two voltages to execute each application for energy efficiency [8,12]. Analytic solutions for this general case do not exist, we thus propose an iterative approach and an approximation algorithm to efficiently search the solution based on the convexity of the energy function.

Lemma 3. Let $\{V_{k,1}, V_{k,2}, \dots, V_{k,k}\}$ and $\{V_{k+1,1}, V_{k+1,2}, \dots, V_{k+1,k+1}\}$ be the optimal k - and $(k+1)$ -voltage set-ups respectively, then we have $V_{k+1,1} \leq V_{k,1} \leq V_{k+1,2} \leq V_{k,2} \leq \dots \leq V_{k,k} = V_{k+1,k+1} = V_n^0$.

An Iterative Approach:

- Start with the single voltage system with voltage $V_{1,1} = V_n^0$;
- Apply the algorithm in Figure 1 to solve for $V_{2,1}$ and $V_{2,2}$, the best voltage set-up for dual-voltage system;
- Repetitively apply Lemma 3 for k -voltage ($k > 2$) system as follows: 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 setting.

An Approximation Algorithm:

- Start with a random m -voltage set-up;
- Fix the $(m-1)$ higher voltages and compute the lowest voltage V_1 by a procedure similar to the algorithm in Figure 1;
- Determine V_2 by fixing the obtained V_1 and the other $(m-2)$ higher 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.

3.4 Determining the Number of Voltage Levels

If we ignore the hardware overhead (e.g., the area and power on the voltage regulators or DC-DC converters) for supporting the multiple voltages, then we should use as many voltages as necessary to reduce the energy consumption [12]. However, supporting

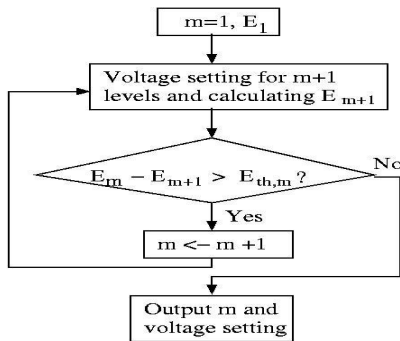


Figure 2. Determining the number of voltage levels.

multiple voltages on the same chip does require additional hardware and will cause area, delay, and also power penalties. Therefore it becomes important to investigate the tradeoff between more voltage levels and the overhead they introduce. Figure 2 shows a scheme on how to determine the optimal number of voltage levels. The threshold energy cost $E_{th,m}$ measures the additional hardware cost in having $(m+1)$ voltages instead of m voltages. We assume that this threshold cost can be obtained empirically.

4. Simulation Results

There are two goals in our simulation: demonstrating the importance of voltage set-up problem and validating our proposed approaches. We formulate the voltage set-up problem in two examples based on a set of randomly generated applications (Table 1) and the MPEG video encoder (Figure 3). The problems are then solved both analytically and numerically by using our approaches. Finally we compare the energy consumption under different voltage set-ups obtained by using exhaustive simulation in Matlab in order to test the correctness of results and the effectiveness of our proposed methods. Note in this section the energy is in the unit of dissipation in one CPU unit at the reference voltage 3.3V.

Table 1. Information on the two ad hoc applications.

Application	Deadline d_i	Execution Time: e_i	Probability p_i	Ideal Voltage (V)
A	10	9	0.03	3.0564
		4	0.18	1.8124
		3	0.39	1.5516
B	8	6	0.04	2.6888
		4	0.10	2.0669
		3	0.12	1.7479
		2	0.14	1.4176

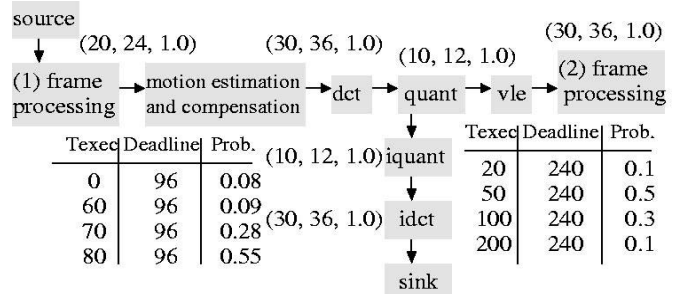


Figure 3. MPEG video encoder execution time distributions and deadlines in 10^4 cycles (redrawn from [10]).

For each example, we apply the proposed approaches to find the best voltage set-ups for dual-voltage, 3-voltage, and 4-voltage DVS systems as reported in Table 2. We also list the energy consumption of the ideal DVS system, where we have one voltage for each possible execution time, and the best fixed-voltage system in the table for comparison.

For the first example with 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 second voltage on the dual-voltage system. With the addition of the third and fourth voltages, we see the continuous increase in energy reduction. On the other hand, comparing to the lower bound in the ideal system, the best fixed-voltage setting consumes more than 150% additional energy. But this "wasted" energy drops to 17.6%, 4.9%, and 2.6% for the dual-, 3-, and 4-

voltage system respectively. It indicates the effectiveness of multiple-voltage DVS system's energy saving.

Table 2. The optimal voltage set-ups and their corresponding average energy consumption per execution. (In the parenthesis of energy columns, “-” is the energy saving over fixed-voltage system, “+” is the “wasted” energy comparing to the ideal DVS 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	1.3833 (-53.2%)	2.8934	23.1478 (-13.3%)
	1.8124	(+17.6%)	1.8551	(+4.0%)
3-voltage	3.0564	1.2337	2.8934	22.4958
	2.0688	(-58.2%)	1.8558	(-15.8%)
	1.5514	(+4.9%)	1.3031	(+1.1%)
4-voltage	3.0564	1.2071	2.8934	22.3020
	2.0768	(-59.1%)	2.6374	(-16.5%)
	1.8119	(+2.6%)	1.8554	(+0.2%)
	1.5509		1.3031	
ideal	—	1.1763	—	22.2506

We have similar observations from the MPEG encoder example except that the energy saving (over the fixed-voltage system) is much lower, albeit a notable 13%. This is because that majority of the energy is consumed on the deterministic subtasks. However, multiple-voltage systems again successfully reduced the “wasted” energy from more than 20% (for fixed voltage) to 4.0%, 1.1%, and 0.2%.

To validate the correctness of our results, we use Matlab to simulate 100,000 iterations of each application under different voltage set-ups for dual-, 3-, and 4-voltage systems. In all the cases, this exhaustive search finds the same solution, within the precision of voltage increment 0.01V we set, as we reported in Table 2 by our methods. Figures 4 illustrates this for the dual-voltage system for the MPEG encoder. We set the high voltage V_2 to go from V_n^0 (2.8934V) to 3.3V, and the low voltage V_1 to go from 1.0V to 3.3V, both with an increment of 0.01V. In the figure, we see that the energy consumption is minimized at the same set-up as we obtained theoretically.

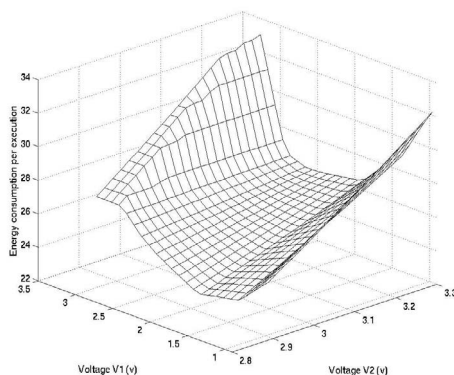


Figure 4. Average energy consumption for the MPEG encoder with different dual-voltage set-ups.

5. Conclusion

We consider the voltage set-up problem for application-specific multiple-voltage DVS systems. The problem seeks to determine the number of voltage levels and the voltage at each level to minimize the energy consumption for a set of applications. We give analytic optimal solution for the dual-voltage system and develop fast iterative and approximation approaches 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. Simulation results show the correctness and efficiency of our approaches. We also observe that multiple-voltage systems can reduce energy consumption almost as much as the ideal DVS systems, which implies that the full potential of DVS can be reached by the practical multiple-voltage DVS systems. We are currently examining ways to extend our approaches to more complicated applications from real life.

6. References

- [1] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen. A dynamic voltage scaled microprocessor system. *IEEE J. Solid-State Circuits*, 35(11):1571-1580, Nov. 2000.
- [2] A. Chandrakasan, S. Sheng, and R.W. Brodersen, Low-power CMOS digital design. *IEEE J. Solid-State Circuits*, 27(4):473-484, Apr. 1992.
- [3] J.-M.Chang and M. Pedram. Energy minimization using multiple supply voltages. *ISLPED*, pp. 157-162, 1996.
- [4] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos. Data driven signal processing: an approach for energy efficient computing. *ISLPED*, pp. 347-352, 1996.
- [5] C. Chen and M. Sarrafzadeh. Provably good algorithm for low power consumption with dual supply voltages. *ICCAD*, pp. 76-79, 1999.
- [6] S. Dhar and D. Maksimovic. Low-power digital filtering using multiple voltage distribution and adaptive voltage scaling. *ISLPED*, pp. 207-209, 2000.
- [7] S. Hua, G. Qu, and S. S. Bhattacharyya. Energy reduction techniques for multimedia applications with tolerance to deadline misses. *DAC*, pp. 131-136, 2003.
- [8] T. Ishihara and H. Yasuura. Voltage scheduling problem for dynamically variable voltage processors. *ISLPED*, pp. 197-202, 1998.
- [9] M. C. Johnson and K. Roy. Datapath scheduling with multiple supply voltages and level converters. *ACM Trans. on Design Automation of Electronics Systems*, 2(3):227-248, 1997.
- [10] A. Kalavade and P. Moghe. A tool for performance estimation of networked embedded end-systems. *DAC*, pp. 257-262, 1998.
- [11] G. Qu and M. Potkonjak. Techniques for energy minimization of communication pipelines. *ICCAD*, pp.597-600, 1998.
- [12] G. Qu. What is the limit of energy saving by dynamic voltage scaling? *ICCAD*, pp. 560-563, 2001.
- [13] D. Mosse, H. Aydin, B. Childers, and R. Melhem. Compiler-assisted dynamic power-aware scheduling for real-time applications. *COLP*, 2000.
- [14] S. Raje and M. Sarrafzadeh. Variable voltage scheduling. *ISLPED*, pp. 9-14, 1995.
- [15] 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. *RTAS*, pp. 164-173, 1995.