

A Unified Optimal Voltage Selection Methodology for Low-power Systems

Foad Dabiri [†]
dabiri@cs.ucla.edu

Roozbeh Jafari [‡]
rjafari@utdallas.edu

Ani Nahapetian [†]
ani@cs.ucla.edu

Majid Sarrafzadeh [†]
majid@cs.ucla.edu

[†] University of California Los Angeles, Los Angeles, CA, 90095 [‡] University of Texas at Dallas, Dallas, TX, 75083

Abstract—Reduction in power consumption has been an important concern in low-power and high-performance systems. This paper addresses the problem of static voltage scaling in such systems which is a well studied technique. In this paper we present an optimal methodology for static voltage scaling. Previous techniques, use path-based timing constraints in the system model which requires exponential runtime even for problem generation. Our main contribution is the unified formulation with linear number of constraints in the optimization problem as opposed to the exponential number. This methodology results in a fully polynomial time solvable problem. Our formulation can be applied to dynamic voltage scaling on single or multiple resources and moreover, it results in a convex optimization problem which can be solved in fully polynomial time. We propose a general formulation for bounded supply voltage assignments as well. Furthermore, we present two heuristics to find and/or map optimal voltages to discrete levels. We evaluated the performance of our techniques on benchmarks from TGFF and MPEG4 video encoder. An average of 43.96% power reduction was gained for unbounded supply voltage assignment along with $\approx 40\%$ average power saving where discrete voltage levels are available.

I. INTRODUCTION

Energy consumption is recognized as one of the most important parameters in designing modern portable electronic and wireless systems in today's very large scale integration (VLSI) circuits. In CMOS digital circuits, power dissipation consists of dynamic and static components. Since dynamic power is proportional to the square of supply voltage V_{dd} and static power is proportional to V_{dd} , lowering V_{dd} is obviously the most effective way to reduce power consumption.

Various voltage schemes have been proposed by researchers for power optimization. A run-time dynamic voltage scaling scheme for low-power real-time systems was proposed in [9]. This scheme employs software feedback control of supply voltage, which is applicable to off-the-shelf processors and provides power reduction by exploiting slack time arising from workload variation. A simulation study on dynamic voltage schemes was performed in [13].

As for static supply voltage assignments, several techniques have been proposed. A dynamic programming technique for solving the multiple supply voltage scheduling problem in both non-pipelined and functionally pipelined data-paths was proposed in [1]. The nature of their proposed method, however, remains a heuristic. An $O(n)$ algorithm is proposed in [3] for assigning supply voltages to serially executing functional units

(FUs) in a digital system such that the overall dynamic energy consumption is minimized for a given timing constraint. Their model, however, cannot be applied to general computational graphs. Another similar scheme is also proposed in [7].

A dual-threshold technique to reduce leakage power by assigning a high-threshold voltage to some transistors in noncritical paths, and using low-threshold transistors in critical path(s) has been addressed in [17]. Despite the fact that their proposed method is optimal, they only consider two voltage levels. Other similar approach was studied in [12].

Dynamic threshold voltage scaling [10] for energy minimization through an extra circuit was studied. Voltage Islands, a scheme that can be used to reduce active and static power consumption for System-on-Chips (SoC), was outlined in [8].

In [2], an optimum methodology for assigning supply and threshold voltages to modules in a CMOS circuit to minimize energy consumption was proposed. A similar technique at the level of application specific modules mapped onto a processing element as opposed to voltage assignment to circuit level modules was suggested in [11]. In their problem formulations, however, the number of constraints are proportional to the number of paths in circuits, which can be exponential, while our proposed technique does not have such a limitation. [19] have used a more efficient formulation but again their approach cannot be easily extended to more generalized systems like those with path based constraints and local deadlines in digital circuits.

II. POWER AND DELAY MODELS

Power dissipation for every processing element in a system based on digital CMOS circuits is in the form of dynamic and static. Dynamic power of each element can be represented as in Equation (1).

$$P_{dynamic} = \sum_{i=1}^N f_{clock} C_i V_{DD}^2 = \Phi_i \cdot V_{DD}^2 \quad (1)$$

C_i is the effective switching capacitance of the gate i , f_{clock} is the clock frequency, V_{DD} is the power supply voltage and Φ_i simply represents $\sum_{i=1}^N f_{clock} C_i$. The sum is taken over all the gates in the module. In this paper we investigate dynamic power dissipation of CMOS based systems. The delay of each

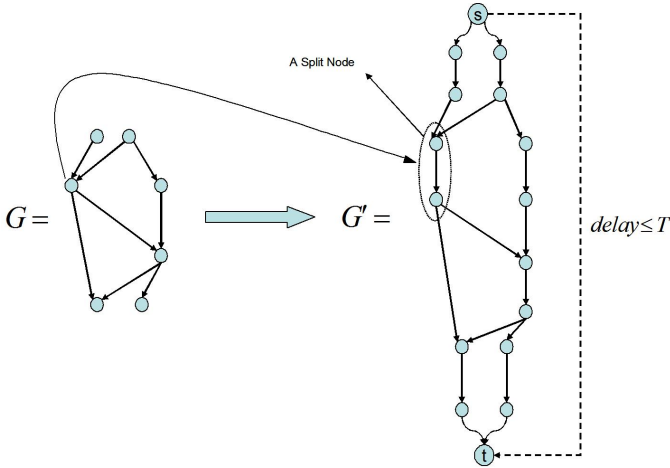


Fig. 1. Graph transformation

unit can be written as:

$$d_i = K_i \times \frac{V_{DD}}{(V_{DD} - V_t)^\alpha} \propto \frac{K_i}{V_{DD}^{\alpha-1}} \quad (2)$$

where K_i and α are technology dependent parameters for the i^{th} unit. Note that discarding V_t in the above equation is not necessarily a reasonable assumption. Specially with the current low voltage technologies, V_{DD} and V_t might be in the same order of magnitude. Fortunately, our proposed method is valid even with considering V_t . In Section 3 we will see that the only required condition on Equation 2 is it's convexity with respect to V_{DD} . Equation 2 satisfies this conditions even with V_t consideration. Throughout this work, we will use the simplified form of Equation 2 for the sake of readability of the paper. From Equations 1 and 2, we can deduce the direct relation between power and delay of a unit:

$$P_i = \phi_i \left(\frac{K_i}{d_i} \right)^{\frac{2}{\alpha-1}} \quad (3)$$

For $1 \leq \alpha \leq 2$ the above function is in convex form with respect to d_i . We assume $\alpha = 2$ just to have simple representation of for problem formulation whereas one can use any feasible value for α . Throughout the rest of the paper, we use Equation 3 in which there is no direct notion of supply voltage. We will assign delay values instead of supply voltage and afterwards using Equation 2 we computer corresponding voltage values.

III. PROBLEM FORMULATION

Intuitively, the voltage scaling problem can be stated as a timing management problem in such a way that given an application with distinct constituting blocks, the maximum tolerable power reduction of individual blocks is desired while the timing constraints of the system is not violated. Although these blocks are often modeled as nodes in a directed acyclic graph (DAG), the problem of voltage scaling on nodes is a special case of a more general voltage scaling on edges. In this section, we illustrate how the the delay budgeting on nodes is transformed to delay budgeting on edges. Therefore,

we focus on delay budgeting on edges throughout the paper. In high level synthesis, systems are usually modeled as a directed acyclic graph $G = (V, E)$. In this model, nodes represent the operational modules and edges stand for the precedence relation between them. We transform the given graph G into G' in such a way that, each node v in G is split into two nodes v_1 and v_2 and an edge connecting v_1 to v_2 . In the transformed graph, the new edges are basically the modules from the original graphs. Figure 1 shows an example of such transformation. In order to have a single input and a single output, nodes s and t have been added to the graph; s is connected to all primary inputs and all primary outputs are connected to t .

Definition The delay of a path $p = \langle s, v_1, v_2, \dots, t \rangle$ from node s to node t is equal to the summation of the delays of each edge along the path. We use the terms delay of the path and the distance between nodes s and t , interchangeably.

The problem is defined as: Given a DAG $G = (V, E)$ and a timing constraint T

$$\text{minimize} \sum_{\forall e_{ij} \in E} P_{ij} \quad (4)$$

such that the delay of every path from s to t is less than or equal to T . P_{ij} is the power consumption of ij^{th} edge in the DAG which is a function of the delay of the edge, d_i . The timing constraint can be stated as: $\sum_{i \in p_i} d_i \leq T$ for every path p_i from s to t . Note that the number of paths in a DAG is exponential in terms of the number of edges in the graph, therefore this formulation is not efficient. Throughout the rest of this section, we will convert it to a formulation with the same objective function that has linear number of constraints.

Theorem 1: In the optimal voltage scaling of a DAG, the distance between any node u and the output t is independent of the choice of the path taken between them and is unique.

Proof: Suppose the claim is not true, i.e. there exists a node

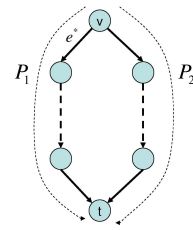


Fig. 2. Figure for Theorem 1

v where its distance to t through path P_1 is less than P_2 (see Figure 2). The intuition behind the proof is that, larger delay assigned to an edge yields in more power saving which is the objective of the optimization problem. If P_1 is shorter than P_2 , there exists an edge (e^*) in P_1 that can be slowed down and still not violate the timing constraint because P_2 is on the critical path from v to t . One immediate candidate for e^* is the first edge in P_1 . Increasing the delay of e^* by $d_{P_2} - d_{P_1}$

will not cause a timing violation and therefore, reduces the total power dissipation.

The following observation is immediately inferred from the above theorem:

Corollary The delay of each path in the optimal solution from the primary input node s to primary output node t is equal to T .

Now that the distance of every node to the destination is independent of the path taken, let t_i be a variable assigned to each node v_i that represents its distance to t . A similar technique was proposed in [4] which has resulted in an efficient integer delay budgeting algorithm. We call t_i the distance variable of node v_i . In other words, t_i is the delay of the system from node v_i to the output. Therefore, the delay and power consumption of each edge (node in the original graph) is represented by:

$$\begin{aligned} d_{ij} &= t_i - t_j \\ P_{ij} &= \phi_{ij} \frac{K_{ij}^2}{d_{ij}^2} = \phi_{ij} \frac{K_{ij}^2}{(t_i - t_j)^2} \\ &\forall e_{ij} \in E \end{aligned}$$

Thus, instead of having a constraint for each path from s to t , we construct the following constraints:

$$h_{ij}(t_i, t_j) = t_i - t_j \geq 0, \forall e_{ij} \in E \quad (5)$$

$$h_{st}(t_s, t_t) = t_s - t_t \leq T \quad (6)$$

Equation 5 enforces that the delay assigned to each edge is positive and Equation 6 guarantees that the distance from s to t is less than or equal to the timing constraint T . Equation 6 can be interpreted as a minimum delay required for the virtual edge between s and t . This edge is also shown in Figure 1. All the timing constraints on paths are reformulated as edge constraints and the optimization problem can be restated as:

$$\text{minimize } f(\vec{t}) = \sum_{\forall e_{ij} \in E} \phi_{ij} \frac{K_{ij}^2}{(t_i - t_j)^2} \quad (7)$$

subject to constraints in Equations 5 and 6. The constraint in 6 enforces the timing constraint on the system. Equations 5, 6 and 7 form a nonlinear optimization problem with a linear number of constraints in terms of the size of the graph. In the next section we solve this problem using the Lagrange Multipliers method.

A. Convexity of the Optimization Problem

Global optimization is the task of finding the absolutely best set of parameters to optimize an objective function. In general, there exist solutions that are locally optimal but not globally optimal. Consequently, global optimization problems are typically quite difficult to solve; in the context of combinatorial problems, they are often NP-hard. In convex optimization problems, a locally optimal solution is also globally optimal. These include LP problems; QP problems

where the objective is positive definite, if minimizing (and negative definite if maximizing). Furthermore NLP problems belong to the same class where the objective is a convex function, if minimizing (and concave if maximizing) and the constraints form a convex set [16]. In this section, we will show that our proposed formulation, yields a convex optimization problem.

Convex optimization problems are far more general than linear programming problems, but they share the desirable properties of LP problems: They can be solved quickly and reliably even in very large size. A convex optimization problem is a problem where all of the constraints are convex functions and the objective is a convex function while minimizing, or a concave function while maximizing. With a convex objective and a convex feasible region, there can be only one optimal solution, which is globally optimal. Several proposed methods – notably Interior Point methods – can either find the globally optimal solution, or prove that there is no feasible solution to the problem.

First we need to show that the feasible solution space is in fact convex. As seen in equations 5 and 6, all constraints are linear which can be viewed as planes, bounding the solution space. It is trivial that non-finite planes yield in a convex solution space. To show the convexity of the objective function (Equation 7), we will show that each term in f is convex and because the sum of convex functions is convex, it suffices to show that $f_{ij} = \phi_{ij} \frac{K_{ij}^2}{(t_i - t_j)^2}$ is indeed convex. Note that $\frac{\partial^2 f_{ij}}{\partial t_i^2} = \phi_{ij} \frac{6K_{ij}^2}{(t_i - t_j)^4} > 0$ and $\frac{\partial^2 f_{ij}}{\partial t_j^2} = \phi_{ij} \frac{6K_{ij}^2}{(t_i - t_j)^4} > 0$. This concludes the convexity of f_{ij} and therefore the original problem is a convex optimization problem.

B. Extension on Bounded Supply Voltages

The methodology described in Section III yields optimum values of supply voltages for each module that minimizes the overall system power dissipation. However, due to technological constraints, the assumption on unbounded supply voltage assignment may not be feasible. In such cases, it is desirable to have a lower/upper bound on the supply voltages (which means for each edge in the transformed DAG model, an upper/lower bound for the supply voltage is imposed). In this section, we refine the formulation of Section III such that the voltages assigned to each component can be chosen within a specific range. Suppose the supply voltage for a module is bounded by:

$$V_{min} \leq V_{DD} \leq V_{max}$$

As seen in Equation 2, V_{min} and V_{max} correspond to a maximum and minimum possible delay for that module, therefore:

$$d_{min} \leq d \leq d_{max}$$

Upper bounds on supply voltages (lower bound on delays respectively) can easily be accommodated. In this case, constraints in Equations 5 can be rewritten as:

$$h_{ij}(t_i, t_j) = t_i - t_j - d_{min_{ij}} \geq 0 \quad (8)$$

Where $d_{min_{ij}}$ is the minimum allowable delay for the ij^{th} edge which corresponds to the largest available voltage for module ij ; $V_{max_{ij}}$. However, lower bounds on supply voltage (upper bound on delays), are not as easy to handle in general. The challenge is that Theorem 1 does not necessarily hold for every graph which has upper bounds on delays assigned to edges and therefore distance variables cannot be defined. Figure 3 shows a simple example in which the upper bound for delay on every edge is 2 and the total timing constraint of the DAG is 7. As seen in the figure, minimum power consumption is achieved when all edges are operating in minimum voltage (maximum delay which is equal to 2) and in that solution, the delay of leftmost and rightmost paths are not equal to each other. Fortunately, we prove that the following lemma holds and therefore we can still define distance variables.

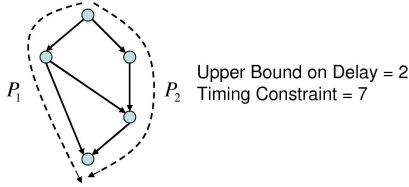


Fig. 3. An example showing how lower upper bounds on edge delay can contradict Theorem 1

Lemma: The optimal voltage scaling of nodes in a DAG will satisfy Theorem 1 on the transformed graph, where voltage scaling is applied to the edges.

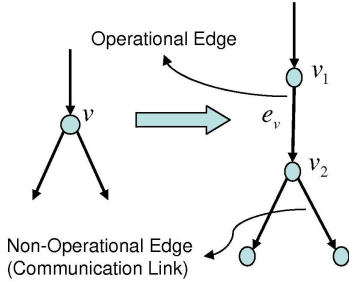


Fig. 4. Node v_2 is adjacent to edges which have no delay bounds

Proof: As seen in the proof of Theorem 1, the branching of two non-equal paths can only happen at the end point of an edge which is a split node. Consider a node v in the original graph G and its split in the transformed graph G' as illustrated in Figure 4. The outgoing edges from v_2 in G' are communication edges, not operational ones. Which means that they just represent dependencies and there is no supply voltage assigned to them. e_v is basically the edge that represents node v from the original problem. The set of outgoing edges from v_2 do not consume power nor have any bound on delays. Therefore, any delay can be assigned to these edges without affecting the total power dissipation.

This fact ensures that the same proof for Theorem 1 still holds. In other words, in an optimal voltage scaling, one can assign "dummy" delays to these non operational edges just

to equalize path delays without increasing the total power consumption. Therefore, distance variables can be defined even with upper/lower bounds on the supply voltages. We add the following timing constraint in presence of lower bounds on supply voltages:

$$h'_{ij}(t_i, t_j) = t_i - t_j \leq d_{max_{ij}} \quad (9)$$

where $d_{max_{ij}}$ is the maximum allowable delay for the ij^{th} edge which is calculated from Equation 2 by substituting $V_{max_{ij}}$ in Equation 2. Equation 6 will remain the same global timing constraint as before. In this section, we outlined that the bounded supply voltages can be handled with the proposed formulation while still maintaining a linear number of constraint.

To solve the above set of equations (Equations 7,5,6), Lagrange dual function and constraints can be written as:

$$\Lambda(\vec{t}, \vec{\lambda}) = P(\vec{t}) - \sum_{\forall e_{ij} \in E} \lambda_{ij} h_{ij}(t_i, t_j) \quad (10)$$

$$\nabla \Lambda(\vec{t}, \vec{\lambda}) = \nabla P(\vec{t}) - \sum_{\forall e_{ij} \in E} \lambda_{ij} \nabla h_{ij}(t_i, t_j) = 0 \quad (11)$$

$$\forall e_{ij} \in E, \lambda_{ij} \leq 0 \quad (12)$$

$$\sum_{\forall e_{ij} \in E} \lambda_{ij} h(t_i, t_j) = 0 \quad (13)$$

This dual optimization problem can be solved efficiently by Interior-point method which simply applies Newtons method to a sequence of equality constraint problems.

IV. DISCRETE VOLTAGE MAPPING

In the previous section, we obtained an optimal voltage assignment to modules in a system that minimizes the energy consumption. These voltages might all have different values and therefore result in large number of power supplies with various voltage levels which may not be available. Current technologies, allow the designer to utilize only a few number of voltages. In this sections, we apply different heuristics to map the optimal voltage assignments from the previous section to a discrete set of values. Such problem is known to be NP-hard [18]. We consider two different scenarios:

A. Fixed Supply Voltage Mapping (FSVM)

Assume a set of k possible voltages is given for a system: $V = \langle v_1, v_2, \dots, v_k \rangle$. The objective is to find a mapping F from the set of optimal voltages obtained in the previous section, \vec{V}_{opt} , to \vec{V} . We propose the following heuristic :For each $v_{i_{opt}}$, we map $v_{i_{opt}}$ to v_j where v_j is the closest value to $v_{i_{opt}}$. Obviously, the timing constraint may be violated. Iteratively, we find a node that lies in the maximum number of critical paths, and increase its voltage to the next level. We will call the first phase of this mapping algorithm NNM, where each voltage is mapped to its nearest neighbor (Nearest Neighbor Mapping). In the section of experimental results, we

will illustrate that even this simple NNM phase, yields a very good power saving while the timing constraint is violated by a small fraction.

B. Variable Supply Voltage Mapping (VSVM)

In this section, we assume that k different voltage levels, v_1, \dots, v_k may be used where k is given but their values are unknown prior to the solution. The objective is to find k voltage levels, along the mapping function F , such that the overall power consumption is minimized. Therefore, we propose the following method; we group the optimal voltages into k groups and map the voltages in each group to the maximum voltage in that group, such that the total square distance of the voltages in a group to the maximum voltage in the that group is minimized, (i.e. $D = \sum_{\forall i} (v_{opt_i} - v_{F(i)})$) is minimized, where F is the mapping function. We apply the following iterative algorithm for this purpose.

procedure VSVM

- 1: $\vec{V}' = \text{sort}(\vec{V}_{opt})$
- 2: $\forall i; F(i) = \max(\vec{V}_{opt})$, this is the initializing step
- 3: **for** $j = 1 : k - 1$ **do**
- 4: $t \leftarrow 1$, t is the index of the new voltage level
- 5: **while** D is decreasing **do**
- 6: $\forall i \leq t, F(i) = \vec{V}'[t]$
- 7: $t \leftarrow t + 1$
- 8: recompute D
- 9: **end while**
- 10: **end for**

After this grouping phase, the power consumption will increase, because each voltage is mapped to a value larger than or equal to optimal value. The new total delay of the DAG, T' , is however decreased. Therefore, we scale all k voltage levels by a factor of $\frac{T'}{T}$. This scaling increases the total delay to T while reducing the total power consumption.

V. EXPERIMENTAL RESULTS

Voltage assignment is applied to designs at various stages of CAD flow. We apply voltage scaling to 15 benchmarks generated by TGFF [14] and one real application: MPEG4 Video Encoder from [6]. TGFF benchmarks are used in [5], [15] and [19] to demonstrate voltage scaling algorithms. In a few interesting studies, gate level circuit benchmarks are used [2] for experimental purpose. Although the results on those benchmarks are quite promising, applying different voltages in gate level with today's technology may not be feasible. Therefore, we used synthesized test benches (tgff 1 - tgff 15) generated by TGFF. Table 1 and 2 summarize the experimental results. The second column shows the characteristic of the benchmarks in terms of number of nodes/number of edges in each DAG. In the first experiment, we applied Lagrange multiplier methods to our proposed formulation to get the optimal voltage scaling. In Table 1, column three shows the power reduction in percentile for the continuous voltage scaling case. The results of NNM (simple voltage mapping as discussed in Section IV) is presented in column 4 with three voltage levels, along with the percentile of timing constraint

violation. Columns 2 and 3 in Table 2 correspond to power reduction results when fixed supply voltage mapping (FSVM) algorithm is applied, i.e. the timing constraint are met. Note that increasing the number of predefined voltage level does not necessarily result in more power reduction and the power saving is totally dependent on what the voltage level values are. In our experiments, we assumed that the predefined voltage levels are equidistant. Power reduction from variable supply voltage mapping (VSVM) algorithm is presented in columns 4 and 5 of Table 2, assuming three and six voltage levels are available.

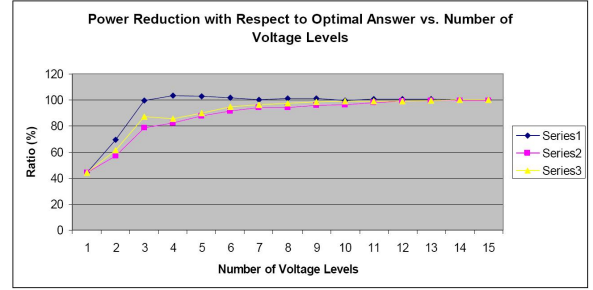


Fig. 5. Effect of discrete voltage level number on power saving for all three algorithms

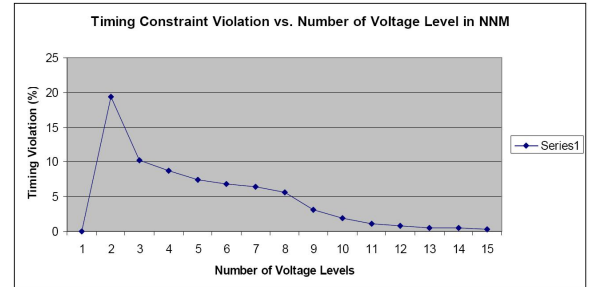


Fig. 6. Improvement of timing constraint violation in NNM method

Benchmark	node/edge	Optimal Power Reduction (%)	NNM only, $k = 3$ Power Reduction (%) / Timing Violation (%)
tgff 1	35/53	52.62	47.32/ 2.63
tgff 2	19/22	34.94	32.37/ 4.44
tgff 3	34/47	41.03	43.15/ 12.21
tgff 4	30/43	30.15	36.89/ 13.14
tgff 5	26/45	18.74	16.63/ 2.63
tgff 6	65/87	48.42	42.83/ 6.25
tgff 7	20/32	17.86	23.52/ 7.73
tgff 8	34/46	42.46	42.96/ 12.31
tgff 9	43/68	50.84	51.14/ 9.73
tgff 10	84/140	59.83	55.63/ 15.42
tgff 11	86/131	61.65	60.84/ 16.07
tgff 12	42/74	55.24	51.26/ 9.09
tgff 13	83/139	58.75	60.74/ 14.34
tgff 14	19/27	30.05	43.33/ 19.11
tgff 15	50/68	56.77	48.05/ 8.42
Average		43.96	43.77/ 10.23
MPEG4		19.2	16.3 / 5.42

Table 1: Power reduction compared to baseline for optimal voltage

scaling and nearest neighbor mapping technique

Increasing the number voltage levels, we observed that the power reduction rapidly approaches to the optimal solution. To justify this claim, we increased the number of voltage levels to 15. Figure 5 illustrates the ratio of power reduction

in discrete scenarios to optimal case for all three cases: Series 1 represents NNM while and series 2 stands for the FSVM approach. VSVM methodology is illustrated with series 3. As seen in Figure 5, increasing the number of voltage levels beyond 3, does not influence the power reduction for the NNM algorithm drastically compared to other two scenarios, instead the timing constraint violation is decreased. Figure 6 illustrated the average timing violation after applying NNM algorithm. The importance of this graph is, although NMM is a simple mapping algorithm, the resulting timing characteristic is reasonably acceptable. An immediate observation from Figure 6 is that the average timing constraint violation in rapidly decreases while applying NNM algorithm with more voltage levels.

VI. CONCLUSION

In this paper, we presented an optimal methodology for static voltage scaling problem on low-power and high-performance systems, and modeled the problem as a convex optimization problem. Our methodology reduces problem size from exponential to linear and can be solved in polynomial time optimally. We also proposed a general formulation for bounded supply voltage assignments and showed the effectiveness of an optimal continuous solution even after simple mapping algorithms. An average of 43.96% power reduction was gained on benchmarks generated by TGFF assuming unlimited number of voltage levels are available. Moreover, for discrete voltage level scenarios, different approaches are proposed. NNM resulted in 43.77% and 44.72% of power saving for three and six voltage levels respectively. We also obtained an average of 34.62% and 40.23% of power reduction in the FSVM method while maintaining the timing constraints. An average of 38.39% and 41.64% power reduction is gained with VSVM method for three and six voltage levels. In this paper, we illustrated the efficiency of using the continuous optimal voltage scaling for discrete case and observed that even limited number of voltage levels (less than 8) can provide us with near optimal power reduction. Also, we tested our methodology on MPEG4 video decoder and gained 14.4% and average of 17.6% in power saving for continuous and discrete voltage levels, respectively. In future, the optimum voltage scaling method may be extended for dynamic voltage scheduling. Furthermore, developing design rules that assist developers with voltage scheduling at the design stage may be investigated. In addition, the effect of voltage level shifters on performance and their related optimization problems may be studied.

Benchmark	FSVM Power Reduction (%) $k = 3$	FSVM Power Reduction (%) $k = 6$	VSVM Power Reduction (%) $k = 3$	VSVM Power Reduction (%) $k = 6$
tgff 1	44.23	51.64	46.72	52.06
tgff 2	29.83	30.12	30.84	32.16
tgff 3	31.64	39.64	38.15	40.65
tgff 4	26.12	28.33	24.63	28.64
tgff 5	14.35	16.19	16.51	17.81
tgff 6	37.83	45.25	45.05	47.07
tgff 7	15.75	17.32	16.00	17.34
tgff 8	30.47	39.84	37.62	41.15
tgff 9	42.01	47.83	48.97	50.13
tgff 10	42.90	52.17	49.83	53.76
tgff 11	47.32	54.53	50.14	55.98
tgff 12	44.88	50.44	50.63	52.74
tgff 13	46.21	51.03	51.52	54.38
tgff 14	25.77	29.54	27.84	28.12
tgff 15	40.11	49.70	41.36	52.60
Average	34.62	40.23	38.39	41.64
MPEG4	14.4	17.9	15.8	18.8

Table 2: Power reduction compared to baseline for fixed and variable supply voltage mapping algorithms

REFERENCES

- [1] J. Chang and M. Pedram. Energy minimization using multiple supply voltages. In *ISLPED '96*, pages 157–162, Piscataway, NJ, USA, 1996. IEEE Press.
- [2] Yuvraj Singh Dhillon, Abdulkadir Utku Diril, Abhijit Chatterjee, and Hsien-Hsin Sean Lee. Algorithm for achieving minimum energy consumption in cmos circuits using multiple supply and threshold voltages at the module level. In *ICCAD '03: Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*, page 693, Washington, DC, USA, 2003. IEEE Computer Society.
- [3] A. U. Diril, Y. S. Dhillon, K. Choi, and A. Chatterjee. An $o(n)$ supply voltage assignment algorithm for low-energy serially connected cmos modules and a heuristic extension to acyclic data flow graphs. In *ISVLSI*, pages 173–179, 2003.
- [4] Soheil Ghiasi, Elaheh Bozorgzadeh, Siddharth Choudhuri, and Majid Sarrafzadeh. A unified theory of timing budget management. In *Proceedings of the 2004 IEEE/ACM ICCAD*, 2004.
- [5] Bitu Gorjiara, Nader Bagherzadeh, and Pai Chou. An efficient voltage scaling algorithm for complex socs with few number of voltage modes. In *Proceedings of the 2004 ISLPED*, pages 381–386, New York, NY, USA, 2004. ACM Press.
- [6] Shaoxiong Hua and Gang Qu. Approaching the maximum energy saving in embedded systems with multiple voltages. In *ICCAD '03: Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*, page 26, Washington, DC, USA, 2003. IEEE Computer Society.
- [7] Mark C. Johnson and Kaushik Roy. Optimal selection of supply voltages and level conversions during data path scheduling under resource constraints. In *ICCD '96: Proceedings of the 1996 International Conference on Computer Design, VLSI in Computers and Processors*, pages 72–77, Washington, DC, USA, 1996. IEEE Computer Society.
- [8] David E. Lackey, Paul S. Zuchowski, Thomas R. Bednar, Douglas W. Stout, Scott W. Gould, and John M. Cohn. Managing power and performance for system-on-chip designs using voltage islands. In *ICCAD '02: Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*, pages 195–202, New York, NY, USA, 2002. ACM Press.
- [9] Seongsoo Lee and Takayasu Sakurai. Run-time voltage hopping for low-power real-time systems. In *DAC '00: Proceedings of the 37th conference on Design automation*, pages 806–809, New York, NY, USA, 2000. ACM Press.
- [10] Y. Moisiadis, I. Bouras, and A. Arapoyanni. Dynamic back bias cmos driver for low-voltage applications. In *Electronics Letters*, volume 36, pages 135–136, 2000.
- [11] Koushik Niyogi and Diana Marculescu. Speed and voltage selection for gals systems based on voltage/frequency islands. In *ASP-DAC '05: Proceedings of the 2005 conference on Asia South Pacific design automation*, 2005.
- [12] P. Pant, R.K. Roy, and A. Chatterjee. Dual-threshold voltage assignment with transistor sizing for low power cmos circuits. In *IEEE Trans. on VLSI Systems*, volume 9, pages 390–394, 2001.
- [13] Trevor Pering, Tom Burd, and Robert Brodersen. The simulation and evaluation of dynamic voltage scaling algorithms. In *ISLPED '98: Proc. of the 1998 international symposium on Low power electronics and design*, pages 76–81, New York, NY, USA, 1998. ACM Press.
- [14] D. Rhodes R. Dick and W. Wolf. Tgff: Task graphs for free. In *CODES '98: Proceedings of the CODES*, pages 97–101, 1998.
- [15] M. Schmitz, B. Al-Hashimi, and P. Eles. Energy-efficient mapping and scheduling for dvs enabled distributed embedded systems. In *DATE '02: Proc. of the conf. on Design, automation and test in Europe*, page 514, Washington, DC, USA, 2002. IEEE Computer Society.
- [16] Lieven Vandenbergh Stephen Boyd. *Convex Optimization*. Cambridge University Press, 2001.
- [17] Liqiong Wei, Zhanping Chen, Kaushik Roy, Mark C. Johnson, Yibin Ye, and Vivek K. De. Design and optimization of dual-threshold circuits for low-voltage low-power applications. *IEEE Trans. Very Large Scale Integr. Syst.*, 7(1):16–24, 1999.
- [18] Han-Saem Yun and Jihong Kim. On energy-optimal voltage scheduling for fixed-priority hard real-time systems. *Trans. on Embedded Computing Sys.*, 2(3):393–430, 2003.
- [19] Yumin Zhang, Xiaobo Hu, and Danny Z. Chen. Task scheduling and voltage selection for energy minimization. In *DAC*, pages 183–188, 2002.