Provably Good Algorithm for Low Power Consumption with Dual Supply Voltages

Chunhong Chen Majid Sarrafzadeh
Department of Electrical and Computer Engineering
Northwestern University, Evanston, IL 60208-3118

Abstract

Dual-voltage approach emerges as an effective and practical technique for power reduction. In this paper we explore the power optimization with dual supply voltages under the given timing constraints. By analyzing the relations among the timing slack, delay and power consumption in a given circuit, we relate the voltage-scaling power optimization to Maximal-Weighted-Independent-Set (MWIS) problem which is polynomial-time solvable on transitive graph. Then we develop a provably good lower-bound algorithm based on MWIS to generate the lower bound of power consumption. Also, we propose a fast approach to predict the optimum supply voltages. The maximum power reduction is obtained by using the modified lower-bound algorithm with optimum voltages. Experimental results show that the resulting lower bound is tight for most circuits and that the estimated optimum supply voltage is exactly, or very close to, the best choice of actual voltages.

1 Introduction

With the increasing demand for personal computing and communication devices, high performance and low power have been the primary goals in the design of VLSI circuits and systems [2]. Since the switching power of CMOS circuits is proportional to the square of supply voltage, one of attractive approaches for power reduction is "*voltage scaling*" which aims to achieve low power consumption by using the reduced supply voltages [4, 5]. In general, the reduced supply voltage results in the loss of circuit performance. However, logic modules on non-critical paths typically have high timing slack. Reducing their supply voltages does not necessarily degrade the timing performance of the circuit.

Usami and *Horowitz* first proposed a dual voltage approach based on the so-called *Cluster Voltage Scaling* (CVS) [6]. The idea is to use *Depth-First Search* from primary outputs for finding gates which can operate at low supply voltage (V_{ddl}) without violating the timing constraints. The disadvantage is that some part of the circuit with high slack may be left to operate unnecessarily at high supply voltage (V_{ddh}), limiting the potential of further power reduction. An extended CVS structure was introduced in [7] where gates of the same voltage may be located in different clusters. However, because of lack of global view, the algorithm easily gets trapped in a local minimum. Also, existing approaches did not account for the switching activity in the circuit. Since different value of two supply voltages can lead to totally different power saving, another open problem with dual-voltage technique is how to select the value of two supply voltages for maximum power reduction. More generally, it is desirable to estimate the lower bound of power consumption for specific circuits with dual voltages.

In this paper, we address the gate-level power optimization with dual supply voltages under the given timing constraints. Our technique makes the following contributions:

- By analyzing the delay and power consumption within a circuit, we relate the power optimization to *Maximal-Weighted-Independent-Set* (MWIS) problem [1].
- We present a provably good algorithm based on MWIS to obtain the lower bound of power consumption.
- We estimate the optimal value of V_{ddh} and V_{ddl} for given circuits to provide a good starting point towards power optimization with dual voltages.

2 Terminology

Consider a combinational circuit represented by a *directed-acyclic graph* $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where each node $v \in \boldsymbol{V}$ corresponds to a logic gate of the given circuit (The terms "*gate*" and "*node*" will be used interchangeably throughout the paper). The existence of a directed edge $(u, v) \in \boldsymbol{E}$ implies that node u is an *immediate fanin* of node v (or, node v is an *immediate fanout* of node u). The set of all immediate fanouts (fanins) of v is denoted by $\boldsymbol{FO}(v)$ ($\boldsymbol{FI}(v)$). If there is a directed path from node v_1 to node v_2 in $\boldsymbol{G}$, v_2 (v_1) is said to be a *transitive fanout* (*fanin*) of v_1 (v_2). The set of all transitive fanouts (fanins) of node v is denoted by $\boldsymbol{TFO}(v)$ ($\boldsymbol{TFI}(v)$). Each node $v \in \boldsymbol{V}$ is associated with a delay $d(v) = k_1 + k_2\, C(v)$, where k_1 is the *intrinsic delay*, k_2 is a constant dependent on the driving ability of the node, and $C(v)$ is the loading capacitance at the output of v. Given the *arrival time* $a(v)$ and *required time* $r(v)$ for node v, its *slack time* is defined as: $s(v) = r(v) - a(v)$. With good accuracy, the node delay, $d(v)$, at supply voltage V_{dd} is proportional to $kV_{dd}/(V_{dd} - V_t)^2$, where V_t is the threshold voltage, and k is a constant [2] (note that the slew effect is not considered here). If the delay of node v at V_{ddh} is denoted by $d_h(v)$, we have

$$d(v) = \frac{V_{dd}}{(V_{dd} - V_t)^2} \cdot \frac{(V_{ddh} - V_t)^2}{V_{ddh}} \cdot d_h(v) \qquad (1)$$

In the well-designed CMOS circuits, the average power consumption is dominated by its switching component which is

$$P_{avg} = 0.5 V_{dd}^2 \cdot f \sum_v C(v) \cdot E(v) \qquad (2)$$

where f is the clock frequency, $C(v)$ and $E(v)$ are the loading capacitance and switching activity of node v, respectively.

0-7803-5832-X /99/$10.00 ©1999 IEEE.

Changing the supply voltage of node v from V_{ddh} to V_{dd} generates the power reduction:

$$\Delta p(v) = 0.5(V_{ddh}^2 - V_{dd}^2) \cdot f \cdot C(v) \cdot E(v) \qquad (3)$$

where $\Delta p(v)$ is called the *power gain* of node v. For any given circuit, our goal is to select some nodes operating at low supply voltage such that the timing constraints are satisfied and the total power gain over these nodes is maximized.

3 Lower Bound of Power Consumption and Optimum Dual Voltages

3.1 Analysis of the node delay and slack

Before discussing the power optimization problem, we give some definitions first.

Definition 3.1 A circuit (or graph $\boldsymbol{G}$) is *safe* if $s(v) \geq 0$ for each node $v \in \boldsymbol{V}$. Otherwise, the circuit is *unsafe*, meaning that it violates the timing constraints.

Definition 3.2 (i) An edge $(u, v) \in \boldsymbol{E}$ in $\boldsymbol{G}$ is called *sensitive* if either $a(v) - a(u) = d(v)$ or $r(v) - r(u) = d(v)$. (ii) A directed path is called *sensitive* if: (a) the path consists of only sensitive edges, and (b) the slack of all nodes on the path is monotonously distributed[1] in the direction of the path. (iii) Two nodes $u, v \in \boldsymbol{V}$ are called *slack-sensitive* if there exists a sensitive path from u to v or from v to u in $\boldsymbol{G}$. Otherwise, they are called *slack-insensitive*.

Definition 3.3 The *sensitive transitive closure graph* $\boldsymbol{G}_s = (\boldsymbol{V}, \boldsymbol{E}_s)$ of $\boldsymbol{G}$ is a directed graph such that there is an edge (v, w) in $\boldsymbol{G}_s$ if and only if there is a directed sensitive path from v to w in $\boldsymbol{G}$.

To deal with the low power problem with dual voltages, let us examine how the delay of a node affects its power gain and slack time. It is assumed that initially a given circuit is safe and that the supply voltage is continuous variable. From (1) and (3), the power gain of node $v \in \boldsymbol{V}$ under unit delay penalty is given by

$$W(v) = \frac{\partial(\Delta p(v))}{\partial(d(v))} = \frac{f \cdot V_{dd}^2 \cdot (V_{dd} - V_t)}{(V_{dd} + V_t)} \cdot \frac{C(v) \cdot E(v)}{d(v)} \qquad (4)$$

Suppose the delay of node v increases by a small positive amount $\varepsilon > 0$, then its slack, $s(v)$, is reduced by exactly ε. Obviously, the slack of any node $x \notin \boldsymbol{TFO}(v) \cup \boldsymbol{TFI}(v)$ can't change. Instead, the slack, $s(u)$, for any node $u \in \boldsymbol{TFO}(v) \cup \boldsymbol{TFI}(v)$ may or may not change, depending on $s(u)$ and the slack sensitivity of u and v. Consider three cases as shown in *Lemma 3.1* below:

Lemma 3.1

Case (a): *if $s(u) \geq s(v)$, then $s(u)$ will decrease by ε as long as u and v are slack-sensitive.*

Case (b): *if $(s(v) - \varepsilon) < s(u) < s(v)$, then $s(u)$ will decrease by $(s(u) - s(v) + \varepsilon)$ as long as u and v are slack-sensitive.*

Case (c): *if either $s(u) \leq (s(v) - \varepsilon)$ or nodes u, v are slack-insensitive, then $s(u)$ remains unchanged.*

Instead of giving the proof of *Lemma 3.1* here, we show an example in Fig. 1, where the delay of node v is 2, while the delay of the rest is assumed to be 1. The arrival, required and slack time for each node are represented by a triple $\{a, r, s\}$ in the figure. From *Definition 3.2*, all edges except (u_3, v) are sensitive. The sensitive paths include $\{u_3, w_2\}$,$\{u_2, v, u_1\}$ and $\{u_2, w_1\}$. Nodes v, u_1 and u_2 are pair-wise slack-sensitive, while v, u_3 are slack-insensitive. We now assume $\varepsilon = 0.8$ and the delay of node v, $d(v)$, increases by ε. Since $s(u_1) = s(v)$, both $s(v)$ and $s(u_1)$ are reduced to be 0.2, as indicated in ***Case (a)***. From ***Case (b)***, $s(u_2)$ decreases to 0.2. Since v, u_3 are slack-insensitive, $s(u_3)$ remains unchanged, as can be expected from ***Case (c)***. In contrast, by using $\varepsilon = 0.2$, we have $s(v) - \varepsilon = 0.8 > s(u_2) = 0.5$. In this case, increasing $d(v)$ by ε does not affect the value of $s(u_2)$, although nodes v, u_2 are slack-sensitive.

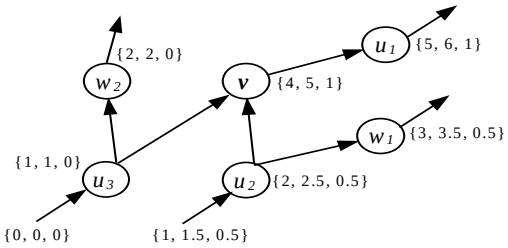


Figure 1. The arrival, required and slack time shown as a triple $\{a, r, s\}$

To maximize the total power gain, it is desirable to keep the minimum number of nodes whose slacks are reduced by the increased delay of node v. In this sense, ***Case (c)*** is the best one. With this in mind, we describe, in the following, how to relate the power optimization to *Maximal-Weighted-Independent-Set* (MWIS)[2] problem [1].

We first select the set of nodes, $\boldsymbol{Q}_m$, with maximum slack (denoted by s_{max}) in $\boldsymbol{G}$, and then construct an *induced subgraph*, $\boldsymbol{G}_m = (\boldsymbol{Q}_m, \boldsymbol{E}_m)$, of $\boldsymbol{G}_s$ (see *Definition 3.3*) on $\boldsymbol{Q}_m$ such that there is an edge $(u, v) \in \boldsymbol{E}_m$ if (u, v) is in $\boldsymbol{G}_s$. We use $\boldsymbol{N}_m(v)$ to denote the set of *neighbors* of node v in $\boldsymbol{G}_m$. There are two important properties of $\boldsymbol{G}_m$ as follows (the proof is omitted).

Lemma 3.2

Property (1): If the delay of any node $v \in \boldsymbol{Q}_m$ increases by $\varepsilon > 0$, then the slack of any node $u \in \boldsymbol{N}_m(v)$ decreases by ε.

Property (2): The delay increase of any node $v \in \boldsymbol{Q}_m$ by ε does not affect the slack of any node $w \notin \boldsymbol{N}_m(v)$ as long as ε is small enough.

Based on the above analysis, if any node $v \in \boldsymbol{Q}_m$ is associated with a *weight*, $W(v)$, which, as given by (4), denotes the power reduction by increasing a unit delay on node v, the maximum power gain can be achieved by selecting nodes in ***MWIS*** of $\boldsymbol{G}_m$ for the delay increase of ε, where $\varepsilon \leq (s(v) - s(w))$, $\forall w \notin \boldsymbol{Q}_m$. Each time the delay of all nodes in ***MWIS*** increases by ε, we update s_{max}, $\boldsymbol{Q}_m$, $\boldsymbol{G}_m$ and $W(v)$, and find MWIS again. This process continues until $\varepsilon > s_{max} \geq 0$.

3.2 Lower-bound algorithm

[1] Assume that a directed path consists of $\{v_1, v_2, \cdots, v_m\}$. The monotonic slack distribution on the path implies that, if $s(v_m) \geq s(v_1)$, then $s(v_{i+1}) \geq s(v_i)$; if $s(v_1) \geq s(v_m)$, then $s(v_i) \geq s(v_{i+1})$, where $i = 1, \cdots, m-1$.

[2] A maximal weighted independent set of a graph is a set of independent nodes (i.e., no two nodes are connected by an edge) such that the sum of their weights is maximum.

From *Lemma 3.2*, selection of ε is key. It can be seen that G_m is independent of ε, and *Lemma 3.2* holds true if $0 < \varepsilon \le s_{max} - s_{max-1}$, where s_{max-1} is the second largest slack of all nodes in G. On the other hand, since the node weight depends on its supply voltage or delay (see (1) and (4)), the node delay and, hence, weight increase dynamically with ε. Therefore, the small value of ε is preferred in general. Theoretically, ε is required to approach zero for the maximum power gain. In the real world, however, smaller ε is not always better. The reason is two-fold. First, too small value of ε can lead to prohibitively expensive computation cost. Second, since **MWIS** depends on the relative node weights in G_m, the small change of weight for specific nodes will not necessarily affect **MWIS**. This is true especially when the weights of nodes in G_m are distributed in a wider range.

Based on this observation, we suggest a reasonable value for ε is $s_{max} - s_{max-1}$. This has been supported by a lot of our experiments on real circuits (see next section). Assuming that the circuit contains K groups of nodes with different slack, the whole **MWIS**-based process can be completed by (K–1) *passes*. Furthermore, choosing $\varepsilon = s_{max} - s_{max-1}$ guarantees that G keeps safe at each pass. This is because $\forall v \in Q_m$, $s(v) = s_{max} \ge \varepsilon \ge 0$, and from *Lemma 3.1*, $\forall u \notin Q_m$, $s(u) \ge s(v) - \varepsilon \ge 0$. Finally, our experience shows $K << n = |V|$ for most circuits, making the computation efficient.

In the above discussion, we assumed the supply voltage is a continuous variable. Instead, when only dual *discrete* voltages are available, the actual power reduction is higher than it is with continuous voltage. From this point of view, our approach provides a lower bound of power consumption under dual voltages. The procedure of the algorithm is outlined below:

Lower-Bound-Algorithm {
Calculate node delay, slack and weight for each node in G under V_{ddh} using (1) and (4);
Identify s_{max} and s_{max-1} in G and let $\varepsilon = s_{max} - s_{max-1}$;
While *($\varepsilon > 0$)* {
Find Q_m and construct G_m ;
*Find **MWIS** of G_m ;*
*Increase node delay by ε for each node in **MWIS**;*
Update the node slack, voltage, weight and ε ;
}
Calculate the final supply voltage, $V_{dd}^*(v)$, for each node v in G;
Obtain the lower bound of power consumption using (3);
}

It should be noted that the MWIS problem is *NP-complete* on general graphs. It is, however, polynomially solvable for transitive graphs [1].

3.3 Prediction of optimal dual voltages

It is interesting to look at the effect of supply voltage on the total power reduction in a circuit. For each gate, the reduced supply voltage results in the increased power saving at the cost of high delay penalty. Under the same timing constraints, using the lower supply voltage means that fewer gates are permitted to work with it. Therefore, an optimum voltage can be expected for the maximum power reduction, depending on the specific circuits. This motivates us to find optimal dual supply voltages.

When the *lower-bound* algorithm terminates, the supply voltage, $V_{dd}^*(v)$, is obtained for each node v in G. In order to meet the given timing constraints, the supply voltage, $V_{dd}(v)$, selected for node v has to be kept more than $V_{dd}^*(v)$. Under dual-voltage environment, either V_{ddh} or V_{ddl} can be used for each node. While it is straightforward to select $V_{ddh} = \max V_{dd}^*(v)$, the optimal value of V_{ddl} can be estimated by finding

$$\left.\begin{aligned} &\max \sum_v (V_{ddh}^2 - V_{ddl}^2) \cdot C(v) \cdot E(v) \cdot k_v \\ &where \quad k_v = \begin{cases} 1 & if \ V_{dd}^*(v) \le V_{ddl} \\ 0 & if \ V_{dd}^*(v) > V_{ddl} \end{cases} \end{aligned}\right\} \quad (5)$$

This can be done by calculating the specific summation in (5) for different values of V_{ddl} ranging from $\max V_{dd}^*(v)$ to $\min V_{dd}^*(v)$ and selecting the maximum sum. This estimate is *optimal* because any deviation from it will either increase power consumption or violate the timing constraints.

3.4 Dual voltage power optimization

When the optimum V_{ddh} and V_{ddl} are available, we modify *lower-bound algorithm* for dual-voltage power optimization as follows. Each time the delay of nodes in **MWIS** increases by ε, we check if some of these nodes are able to work at V_{ddl}. If yes, pick them up and update their weight to be zero. The reason is that, once a node works at V_{ddl}, no further increase of its delay is needed. By doing so, more delay *budget* can be provided for other nodes. Our experiments show that, for most tested circuits, this *modified* algorithm leads to the power consumption which is very close to the lower bound.

4 Experiment and Discussion

Our algorithms were implemented under *SIS* environment [3] and tested on a set of *MCNC*'91 benchmarks. We first used the minimum delay of the circuit as timing constraints. The original power consumption was estimated at the supply voltage of 5V and the clock frequency of 20MHz. On average, the lower bound accounts for about 66% of original power, as will be seen in Fig. 2. To test the optimum voltage estimation, we optimized the same circuits with the *modified* algorithm (described in Section 3.4) using different values of V_{ddl}. The results are summarized in Table 1. Depending on specific circuits, the optimum value of V_{ddl} ranges from 2.2V to 3.5V (here only optimum V_{ddl} was given because V_{ddh} is 5V under the minimum delay constraints). It can be seen that the maximum power saving was achieved at a specific value of V_{ddl} (as shown in shaded box of Table 1) which is exactly, or very close to, the estimated optimum voltage (as shown in the last column of this Table). Although, for a few circuits (e.g., circuit *frg2*), the estimate and actual optimum voltage are much different, the resulting power reduction is nevertheless always very similar.

To look at how the value of ε affects the performance of the lower-bound algorithm, we tested it on different fixed values of ε (note that $\varepsilon = s_{max} - s_{max-1}$ implies it varies dynamically). Fig. 2 shows the resulting lower bound (normalized to original power) and CPU time (on a SUN SPARCstation 5 with 32MB RAM) for the same circuits as above. For most circuits (except one), the lower bound varies within the range of 3% for

different ε , as shown in Fig. 2(a). In contrast, CPU time increases with decreased ε, as shown in Fig. 2(b). Our experiment shows the further reduction of ε results in too expensive computation cost only with almost the same lower bound estimate. We confirm that it is reasonable to choose $\varepsilon = s_{max} - s_{max-1}$ in terms of both accuracy and efficiency of the algorithm.

Finally, we tested our algorithm using different timing constraints to provide the tradeoff between power and delay with dual voltages. Experimental results show that the lower bound of power decreases as the timing constraints are relaxed. However, since the timing penalty increases quickly when the supply voltage of a gate is reduced to the smaller value, the looser timing constraints do not always lead to the lower optimum voltage.

5 Conclusion and Further Work

We have presented a provably good algorithm based on MWIS for low power consumption with dual supply voltages. It has been shown that dual-voltage approach can achieve significant power saving without degrading the circuit timing performance. When using different supply voltages in a circuit, one needs to tackle more issues such as *level converters* at the interface of different voltages. Additional work is required to minimize the number of level converters. Fortunately, our study shows that, by using effective algorithms, the power overhead of level converters can be controlled under 5%, on average, of original power. The detailed discussion is to be given in a separate paper.

Acknowledgment

This work was supported in part by the *National Science Foundation* under Grant MIP-9527389.

References

[1] R.H. Mohring, "Graphs and Orders: the Role of Graphs in the Theory of Ordered Sets and its Application," *Published by D. Reidel Publishing Company, edited by I. Rival, New York and London*, pp.41-101, May 1984.

[2] A.P. Chandrakasan, S. Sheng and R.W. Brodersen, "Low-Power CMOS Digital Design," *Journal of Solid-State Circuits*, vol. 27, no. 4, pp.473-484, April 1992.

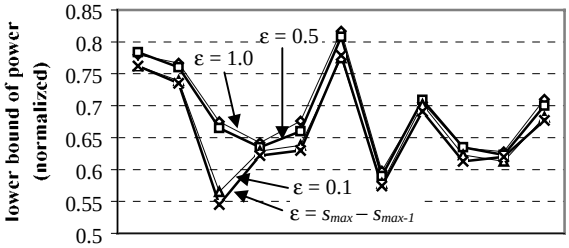
[3] E.M. Sentovish, et al, "SIS: A System for Sequential Circuit Synthesis," *Technical Report UCB/ERL M92/41, Univ. of California, Berkeley*, May 1992.

[4] S. Raje and M. Sarrafzadeh, "Variable Voltage Scheduling," *Proc. of International Symposium on Low Power Design*, pp.9-14, April 1995.

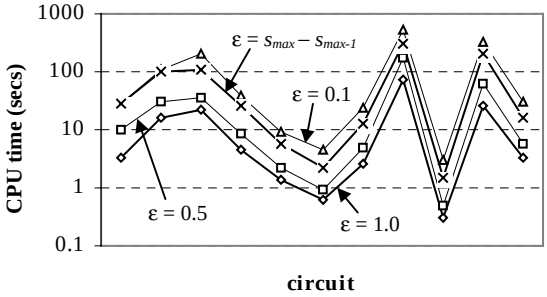
[5] J.M. Chang and M. Pedram, "Energy Minimization Using Multiple Supply Voltages," *IEEE Transactions on VLSI Systems*, vol. 5, no. 4, pp.1-8, December 1997.

[6] K. Usami and M. Horowitz, "Cluster Voltage Scaling Technique for Low Power Design," *Proc. of International Symposium on Low Power Design*, pp.3-8, April 1995.

[7] K. Usami, et al, "Automated Low Power Technique Exploiting Multiple Supply Voltages Applied to a Media Processor," *Proc. of Custom Integrated Circuit Conf*., pp.131-134, 1997.



(a) lower bound of power



(b) CPU time

Figure 2. Performance of lower-bound algorithm

example circuit	maximum power reduction (% of original power) using different V_{ddl}												est. optimum
	2.0V	2.2V	2.4V	2.6V	2.8V	3.0V	3.2V	3.4V	3.6V	3.8V	4.0V	4.2V	V_{ddl}
9symml	13.9	15.2	14.7	14.5	15.5	17.0	16.1	16.0	17.5	15.3	15.5	13.2	3.5V
C1908	22.2	23.0	22.9	25.3	24.8	25.0	25.0	24.7	24.4	22.6	20.9	18.4	2.6V
C880	43.0	43.8	44.5	43.1	41.2	38.7	35.9	33.1	29.7	26.0	22.4	18.4	2.5V
apex7	27.3	28.9	30.7	30.6	30.9	30.0	28.4	26.5	24.4	22.6	20.2	16.6	2.9V
b9	20.5	20.8	22.2	21.2	26.3	26.5	26.4	24.2	23.5	22.0	19.0	16.2	3.1V
c8	6.2	9.8	9.4	12.2	11.5	11.3	14.7	13.6	13.6	11.2	10.8	8.9	3.5V
frg1	35.4	36.7	36.6	35.9	34.4	33.2	30.8	28.1	25.4	22.3	19.4	15.8	2.2V
frg2	22.5	22.0	20.5	28.9	28.4	26.8	23.1	29.2	25.1	24.8	21.1	17.4	2.6V
i1	30.4	30.3	29.9	31.5	29.7	27.3	25.2	24.9	23.1	20.3	17.3	14.1	2.3V
i7	21.8	26.0	18.5	17.5	16.5	15.4	27.3	24.8	22.2	18.1	17.2	14.3	3.1V
term1	18.5	20.9	22.7	27.7	29.7	29.7	28.6	28.2	28.0	24.2	21.7	18.1	2.9V

Table 1. The optimum supply voltage and maximum power reduction using different values of V_{ddl}