VLSI DESIGN
2001, Vol. 12, No. 2, pp. 125–138
Reprints available directly from the publisher
Photocopying permitted by license only

Power Optimization of Delay Constrained Circuits

ANSHUMAN NAYAK[a,*], MALAY HALDAR[a,†], PRITH BANERJEE[b,‡], CHUNHONG CHEN[c,¶] and MAJID SARRAFZADEH[c,§]

[a]#L458, [b]#L463, [c]#L469, *Technological Institute, 2145 Sheridan Road, Evanston, IL 60208*

(Received 20 June 2000; In final form 3 August 2000)

We present a framework for combining Voltage Scaling (VS) and Gate Sizing (GS) techniques for power optimizations. We introduce a fast heuristic for choosing gates for sizing and voltage scaling such that the total power is minimized under delay constraints. We also use a more accurate estimate for determining the power dissipation of the circuit by taking into account the short circuit power along with the dynamic power. A better model of the short circuit power is used which takes into account the load capacitance of the gates. Our results show that the combination of VS and GS perform better than the techniques applied in isolation. An average power reduction of 73% is obtained when decisions are taken assuming dynamic power only. In contrast, average power reduction is 77% when decisons include the short circuit power dissipation.

Keywords: Voltage scaling; Gate sizing; Low power; Digital signal processors; Short circuit power

1. INTRODUCTION

Advances in semiconductor technologies have led to chips with millions of transistors. As circuit density and speed increases, power dissipation has become one of the critical parameters in circuit design. The expanding and converging fields of computing and digital comunications are creating new demands for high performance and programmable signal processing engines. To enhance the performance capabilities of today's DSP systems would imply a higher power consumption. Since, the fastest growing area in the computing industry is the provision of high throughput DSP systems in a portable form, the operating time of these systems provided by the battery becomes a major design issue. Hence, a lot of research has been done for power reduction at various design levels of abstraction (such as system, architectural, logic and layout levels) [1], especially for portable DSP applications.

*Corresponding author: Tel.: (847) 467-4610, Fax: (847) 491-4455, e-mail: nayak@ece.nwu.edu
†Tel.: (847) 467-4610, e-mail: malay@ece.nwu.edu
‡Tel.: (847) 491-3641, e-mail: banerjee@ece.nwu.edu
¶Tel.: (847) 491-7378, e-mail: chen@ece.nwu.edu
§Tel.: (847) 491-7378, e-mail: majid@ece.nwu.edu

The average dynamic power consumed by a CMOS circuit is given by [1]

$$P_{avg} = 0.5V_{dd}^2 f \Sigma C(v)E(v) \quad (1)$$

where f is the clock frequency, V_{dd} the supply voltage, $C(v)$ the load capacitance of gate v, and $E(v)$ is the switching activity at the output of gate v. Due to the fact that the charging/discharging of capacitance is the most significant source of power dissipation in CMOS circuits, previous work optimizes the power by considering three factors in a circuit: supply voltage, load capacitance and switching activity. However, most of them deal with one factor at a time. In this work, we are interested in power optimization by reducing both the supply voltage and the load capacitance.

Since the dynamic power consumption is quadratically related to supply voltage, reducing supply voltage (or voltage-scaling) promises to be an effective technique for power saving. The basic problem with *Voltage Scaling* (VS) is the increased circuit delay, since the relation between delay (t_d) and supply voltage (V_{dd}) is given by [1]

$$t_d = \frac{C \times V_{dd}}{K \times (V_{dd} - V_T)^2} \quad (2)$$

where C is the load capacitance, V_T the threshold voltage, and K a constant. If V_{dd} is much greater than V_T, then the delay is almost inversely proportional to supply voltage. For supply voltage near the threshold voltage, however, the V_T term causes the delay to increase rapidly. Another major overhead in using different supply voltages in a circuit is the additional *level converters* required at the interface and layout design. For this reason, it is advisable to restrict oneself to dual-voltage approach where two supply voltages are available for power optimization. Another technique for reducing power at the logic or transistor level is the technique of *Gate Sizing* (GS) which targets power optimization by reducing the load capacitance. Since the intrinsic resistance of the gate is inversely proportional to the size of the gate, GS results in an increase in delay of the gate. Gate sizing is well known to be a useful tool for reducing circuit delays in CMOS integrated circuits. Several methods have been proposed as solutions when the problem is posed as an area-delay tradeoff, such as in the work in [9–11].

From a general point of view, reducing either supply voltage or physical size of a gate, at logic level, leads to a gate delay increase which implies decreased slack time. In this sense, VS and GS can be effective for delay-constrained optimization only if the given circuit has significant timing slack available in some or all of its constituent gates. Because of the discrete nature of supply voltages or gate sizes, VS or GS alone tends to leave more slacks *unutilized*, [20] preventing effective power reduction. Further, slacks used up by one technique could have been used by the other technique to give higher power reduction. This fact motivates us to opt for a combined VS and GS algorithm. We propose a fast heuristic for GS and VS which would identify the maximum number of gates for gate sizing or voltage scaling under the delay constraints so that the total power dissipation of the circuit is minimized.

Previous approaches have also attempted to minimize the total power using simultaneous voltage scaling and gate sizing [12]. But these approaches consider the dynamic power dissipation only, and neglected the role of the short-circuit power. However, this is not a valid assumption as short-circuit power accounts for under 20% of the total power. Minimizing a power function that considers only the dynamic power, without any constraints on delay, would imply that all transistors must necessarily be minimum sized. However, a minimum-sized circuit does not necessarily correspond to a minimum power circuit, the effect being more pronounced when large loads are driven. Further, down sizing a gate might increase the short-circuit power of the fanout gates which could be high enough to offset the decrease in the dynamic power. Most of the traditional models for short-circuit power neglect

the effect of the load capacitance and are incorrect. In this work, we use a more accurate estimate for short-circuit power and minimize the total dynamic and short circuit power using a combined VS and GS technique. We also propose a fast algorithm which would identify more nodes for sizing or for voltage scaling.

Our optimization problem may be described as:

$$\textbf{minimize } Power(W, V) \tag{3}$$

$$\textbf{subject to } Delay(W, V) \leq T_{spec} \tag{4}$$

$$V_i = V_{high} \text{ or } V_{low}, \forall \text{gate } i \tag{5}$$

$$Maxsize(i) \geq w_i \geq Minsize(i) \tag{6}$$

where both *Power* and *Delay* are functions of gate sizes (**W**) and supply voltages (**V**), T_{spec} is the timing constraints, V_{high} and V_{low} are two supply voltages, V_i and w_i are the supply voltage and size of gate i, respectively, and $Minsize(i)$ and $Maxsize(i)$ are given by the gate library. This is a delay-constrained power-minimizing problem. In [16], a method which makes use of *transistor reordering* was described to address a similar problem. Since transistor reordering is simply intended for reducing the average number of transitions at internal nodes of gates for low power, the resulting power reduction is very limited. In this work, we provide new cost models for delay and power with voltage scaling and gate sizing. Algorithms for single VS, single GS and combined VS and GS are proposed to optimize power. Experiments show that the combined VS and GS obtain maximum power improvement.

For our work, we assume that switching activity is a constant for each node and is independent of gate delays. Switching activity is the measure of signal transitions per clock cycle. Switching activity at all nodes inside a circuit not only depends strongly on the topologic structure and input patterns of the circuit, but may also vary with gate delay which introduces *glitching* transitions. Therefore, the zero-delay model provides a lower bound on the activity. Under a general delay model, updating activities iteratively, is computationally prohibitive. Fortunately, VS and GS do not change the circuit topology, and both tend to reach *path-balancing* by reducing the slacks. This helps eliminate glitching to some extent. Intuitively, for the purpose of power reduction, the nodes with high switching activity are good candidates to work at low supply voltage by VS (or work with the small load capacitance by GS).

The remainder of the paper is organizes as follows. Section 2 discusses delay and power modeling with both VS and GS. Section 3 discusses the VS and GS problem in detail. In Section 4, we discuss an algorithm for combined VS and GS for power optimization. Finally, experimental results are described in Section 5.

2. TIMING AND POWER MODELS

Because of the nature of the problem shown in Eqs. (3–6), the general idea behind GS (or VS) is to iteratively select a set of gates to down-size (or reduce their supply voltages), so that the total power reduction is maximized and the timing constraints are met. Thus, a reasonably accurate timing/power model is required to estimate the delay and power consumption of a gate under specific supply voltage and physical size. In this section we discuss the timing model followed by the dynamic and the short-circuit power model used by us.

2.1. Timing Model

In most standard-cell libraries, the gate delay is defined as

$$d_i = \tau_i + c_i \frac{C^i_{load}}{w_i} \tag{7}$$

where τ_i is the intrinsic delay, w_i and C^i_{load} are size and load capacitance of gate i respectively, and c_i is a constant. The load drive capability of gate i

increases with w_i. The internal capacitance of gate i, however, varies almost linearly with w_i. These together keep τ_i almost independent of w_i. C^i_{load} is determined by the size of the fanout gates and wiring capacitances, *i.e.*,

$$C^i_{load} = C_{wire} + c \sum_{j \in FO(i)} w_j \tag{8}$$

where FO(i) is the set of fanouts of gate i, and c is a constant. When ignoring the wiring capacitance, (5) can be written as

$$d_i = \tau_i + k_i \sum_{j \in FO(i)} w_j / w_i \tag{9}$$

where $k_i = c \cdot c_i$. Basically, (7) indicates that a larger gate is required for the delay reduction if it drives more fanouts. Furthermore, it has been shown in [13] that the gate delay at supply voltage V_{dd} is approximately proportional to $kV_{dd}/(V_{dd} - V_t)^2$, where V_t is the threshold voltage, and k is a constant. Assuming d_i in (7) is the delay at V_{high}, the gate delay with size w_i and supply voltage V_i is given by

$$d_i(w_i, V_i) = \left(\tau_i + k_i \sum_{j \in FO(i)} w_j / w_i\right) \cdot \alpha_i \tag{10}$$

$$\text{where } \alpha_i = \frac{V_i}{(V_i - V_t)^2} \cdot \frac{(V_{high} - V_t)^2}{V_{high}} \tag{11}$$

For the purpose of VS, V_i can be either V_{high} or V_{low}. From (8), reducing supply voltage results in increased delay of the gate, while reducing gate size does not always degrade the delay. The reason is that the loading and, hence, the delay of its fanins decreases with the reduced size of this gate.

2.2. Dynamic Power Dissipation

The dynamic power dissipated in a circuit corresponds to the power dissipated in charging and discharging capacitances in the circuit. The magnitude of this power for a gate driving a load capacitance C^i_{load}, and internal capacitance $C^i_{int} = c \cdot W_i$, operating under a clock frequency f and having a probability p_T of switching is given by

$$P^i_{dynamic} = 0.5(C^i_{load} + C^i_{int})V^2_{dd} f p_T \tag{12}$$

where V_{dd} is the supply voltage. It can be seen that reducing the size of gate i leads to the saved power consumption of both gate i itself and its fanin gates.

2.3. Short Circuit Power Dissipation

Most transistor sizing methods have considered only the dynamic power dissipation. Recently, a few methods have also considered short circuit power using the formula

$$P_{sc} = \frac{\beta}{12}(V_{dd} - 2V_T)^3 \cdot \tau \cdot f \cdot p_T \tag{13}$$

where β is the MOS transistor gain factor, and τ is the transition time of the input transition, and f and p_T are as defined earlier.

Equation (13) is inaccurate since it does not model the effect of the load capacitance on the short circuit power. The short circuit power dissipated by an inverter depends on the following parameters:

- the size of the n-transistor, w_n
- the size of the p-transistor, w_p
- the input rise time, τ
- the output load capacitance, C_L.

A more appropriate model for short-circuit power dissipation has been proposed [14] to be:

$$P_{sc} \propto w_n^{0.75} w_p^{0.82} C_{load}^{-0.085} \tau^{1.49} \tag{14}$$

Assuming that $w_p = 2 \cdot w_n$, a modified model would be:

$$P_{sc} \propto w^{1.57} C_{load}^{-0.085} \tau^{1.49} \tag{15}$$

where w is the width of gate i. The input transition time is modeled as:

$$\tau_i \propto R_i \cdot C_i \tag{16}$$

$$R_i = K \cdot 1/w_i \tag{17}$$

$$C_i = K_1 \cdot w_i + K_2 \tag{18}$$

where R_i and C_i are the drain resistance and capacitances of gate i respectively and K, K_1 and K_2 are the constants of proportionality. The constants were evaluated assuming a 0.18 micron technology and a unit-sized gate's input capacitance equal to 0.097 fF and output resistance equal to 23.8 kΩ [15].

3. VOLTAGE SCALING

Reducing the supply voltage, or *voltage scaling* (VS), promises to be an effective low-power technique since the dynamic power consumption is quadratically related to the supply voltage [2–8, 17]. While reducing the supply voltage of a whole circuit suffers from circuit speed loss, a low voltage applied only to non-critical paths of the circuit does not necessarily lead to performance degradation. The major overhead in using different supply voltages at different parts of a circuit is that *level converters* are required to eliminate the static current at their interface [4, 18]. However, the level converters introduce additional power penalty. To avoid too many level converters, it is reasonable to use a dual-voltage approach in which only two supply voltages are available for the optimized circuits.

The typical dual-voltage approach is the *Cluster Voltage Scaling* (CVS) scheme [4]. Its basic idea is to use the depth-first search from the primary outputs to find gates which may operate at a low supply voltage without violating the timing constraints of the circuit. A gate is not allowed to operate at a low voltage until all its transitive fanouts have been selected to do so. This, to a large extent, limits the effectiveness of the algorithm, since a gate with small slack does not imply that the slacks of all its transitive fanins are also small. A linear programming approach was also proposed [18] to address the dual-voltage problem. However, it is based on the *delay balanced configurations* whose generation requires very expensive computation cost. In [6, 19], a *Two-Voltage Power-Optimization* (TVPO) algorithm is proposed to reduce power by translating the power optimization problem into the *Maximal-Weighted-Independent-Set* (MWIS) problem and allowing as many gates as possible working at V_{low}. The number of level converters at the boundary of high-voltage and low-voltage gates is reduced using the "*constrained*" *Fiduccia-Mattheyses* (*F-M*) algorithm [21]. Section 5 talks about the limitations of the MWIS approach which has a high execution time due to slow convergence of the algorithm. We propose a path based heuristic which is faster than the MWIS approach. The number of nodes operating at a lower voltage is limited by the slack of the circuit.

4. GATE SIZING

Gate sizing consists of choosing for each node of a technology mapped network, a gate implementation in the library so that the total power of the circuit is minimized without affecting the overall delay of the network, *i.e.*, under some delay constraints. This is possible as gates in the non-critical path of the network have a lot of slack so that they can be down sized to save on power without violating timing criticality. Figure 1 shows the effect of down sizing gate G on the total power of the circuit. On down sizing gate G, the input capacitance of Gate G decreases. Hence, the load capacitances of the gates which are the fanins of this gate G, *i.e.*, gate G1 decreases. According to Eq. (9), this results in a decrease in the dynamic power of gate G1. As a consequence of down sizing gate G, the transition time of the signal at the output of gate G increases. This effects the gates which are the fanouts of gate G as the time for

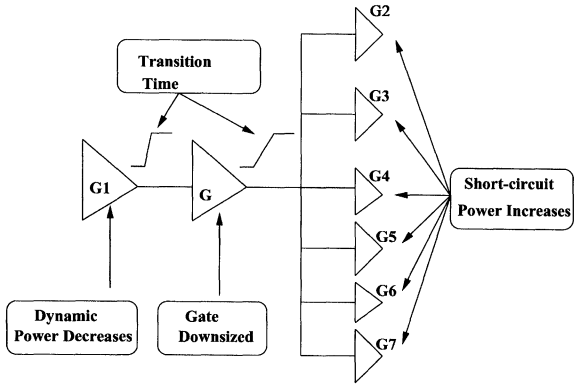

FIGURE 1 Effect of gate sizing on dynamic power and short circuit power.

which both the n and the p gates are ON is increased. This results in an increase in the short-circuit power dissipated by the fanout nodes. Hence, if the number of fanouts are very high, then the total increase in short-circuit power dissipation may offset the decrease in dynamic power dissipation resulting in an increase in the total power, even though we have down sized gate G.

Figure 2 shows the need for optimally choosing the gates for down sizing. If gate G is chosen for down sizing, then the corresponding decrease in slack of this gate, will reduce the slack of its fanout

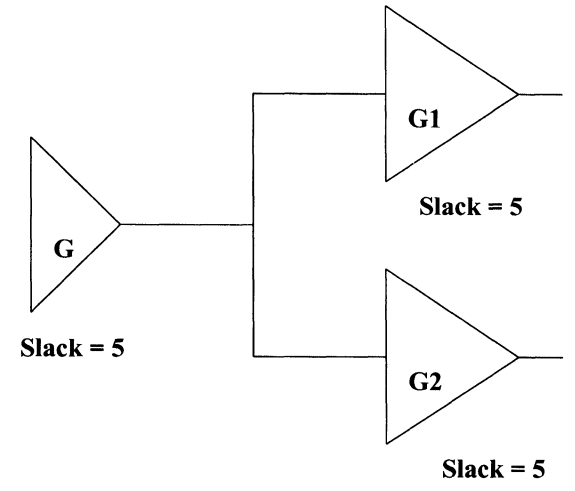

FIGURE 2 Gates which are part of less paths should be down sized.

gates which could have been down sized. On the contrary, if both the fanout gates G1 and G2 were down sized, then we would have got a greater reduction in power. Hence, gates which are part of less paths are better candidates for down sizing before gates which are a part of a large number of paths. Again, since both dynamic and short-circuit power is directly proportional to switching activity, gates with a high switching activity should be down sized earlier. Section 5 describes an algorithm for combined voltage scaling and gate sizing.

5. COMBINATION OF VOLTAGE SCALING AND GATE SIZING

Since both VS and GS decrease the available slack in the circuit, it would be better to apply the two techniques in a simultaneous fashion rather than one after the other. In [12], a technique for power reduction by simultaneous VS and GS using a maximum weighted independent set (MWIS) approach has been proposed. Formulating the power optimization problem as a maximum weighted independent set of the sensitive transitive closure of the graph exposes several opportunities to reduce power. However, the time complexity of the algorithm is quite high. The algorithm attempts to reduce power dissipation by finding a set of nodes for which delay can be traded for power. The selected nodes are usually sized down or operated at a lower V_{dd}. This results in a lower power dissipation and increased delay for the node. To ensure that the increase in the delay of the nodes does not violate any critical path timing constraints, the delay at any step is increased by at most $min\{min_{v \in Q_m}(\Delta d(v)), s_{max} - s_{max-1}\}$. s_{max} is the maximum slack available for any node in the graph and s_{max-1} is the second largest slack available. $min_{v \in Q_m}(\Delta d(v))$ is the minimum change in delay feasible among all the nodes of the graph. Only the nodes with the maximum slacks are considered to increase their delays in each iteration. In a graph $G(V, E)$ where each node has a

different slack, the number of iterations may be $O(V)$, as in each iteration the maximum slack is reduced to the next highest value. As each iteration does a transitive closure computation, the total time complexity may run upto $O(V^4)$. Furthermore, due to the discrete nature of the voltage scaling and gate sizing techniques, the possible delay increase may not equal ε exactly, where $\varepsilon = min\{min_{v \in Q_m}(\Delta d(v)), s_{max} - s_{max-1}\}$. This pushes the number of iteration higher, increasing the complexity even beyond $O(V^4)$.

5.1. A Fast Heuristic

The principal reason behind the success of the MWIS based approach is that the algorithm is able to choose the maximum number of nodes to trade delay for power given the slacks along the paths. For example, consider Figure 3. The MWIS algorithm obtains the optimal solution because it selects the nodes V_1, V_2, V_3, V_4 over the nodes V_5, V_6 or V_7 to introduce delay. Our heuristic is guided by the same principle. The heuristic is based on the *number of paths* that pass through a node from any primary input to any primary output. The intuition is that if the number of paths that pass through a node are large, then introducing a delay at that node uses up the slack of a large number of nodes that lie on the paths that pass through that node. On the other hand introducing delay to a node which has small number of paths passing through it will affect the slacks of a small number of other nodes. Returning to the example of Figure 3, the number of paths that pass through each node are shown in parenthesis. For simplicity, the delay of each node is assumed to be 1. If we take into account the number of paths that pass through each node in selecting which nodes to introduce delays, giving more priority to nodes that have less paths passing through them, then we arrive at the same solution given by the MWIS algorithms. Thus we use the number of paths that pass through each node in deciding which nodes to introduce delays. Further, since power dissipated at a node is directly proportional to the switching activity at the node, nodes with a high switching activity should be gate sized or voltage scaled first. This guides us to the following weight function for each node.

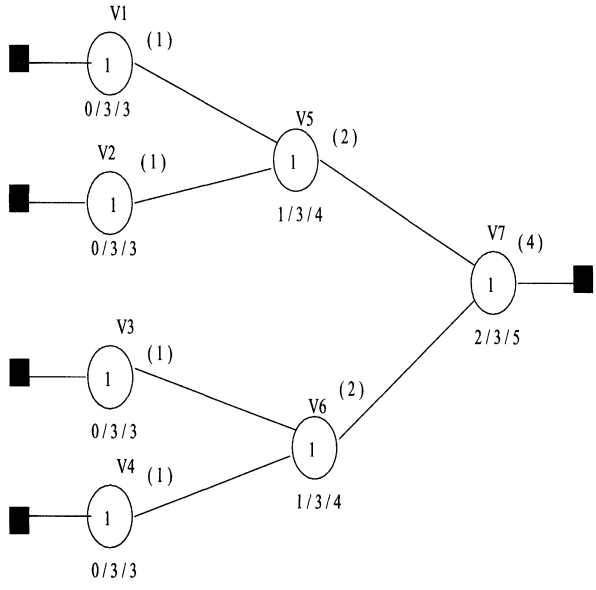

FIGURE 3 An example showing that our path based heuristic gives the optimum result.

$$Weight(i) = \frac{Slack^{\alpha} p_T^{\beta}}{(No.\ of\ Paths)^{\gamma}} \tag{19}$$

where p_T is the switching probability and α, β, γ were assumed to be 1. The weight function assigns a larger weight to gates which have larger slack as these gates can be sized or voltage scaled by a large factor giving us more reduction in power. Also, gates with high switching activity are given a larger weight as power reduction is directly proportional to the switching activity of the gates. Our path based heuristic assigns a lower weight to gates having large number of paths passing through them so that changing slack of an individual gate does not reduce slack of a large number of gates. The parameters α, β, γ were chosen to be 1 so that the effect of slack, switching activity and number of paths on the total power reduction could be studied. These parameters could be changed to obtain better solutions.

The heuristic is described next. Afterwards we describe the algorithm to calculate the number of paths that go through a node. Note that computing the number of paths going through a node is efficient. Moreover, as it is a property of the graph that does not change with the delays of the nodes, we need to calculate it only once as opposed to the MWIS approach where the MWIS had to be calculated after each iteration.

Algorithm 1 proposes our combined VS and GS algorithm. This has the advantage that any slack leftover by one of the techniques will be used over by the other technique. Further, the technique which would bring the maximum power reduction would be used for the particular node. The algorithm finds out the number of paths through each gate and uses this to assign a weight to each node based on the available slack in the node using Eq. (19). Gates which have a larger slack and have less paths passing through them are initially chosen for VS or GS. The change in the total power per unit delay is calculated for these chosen gates. Since the main objective is to achieve a maximum power reduction, gates are chosen for VS or GS depending on which operation decreases the total available slack in the circuit by the least amount. This algorithm terminates when the available slack in the circuit is reduced so that anymore VS or GS operation would violate the timing constraints of the circuit.

ALGORITHM 1 *Voltage Scaling+gate Sizing*

do
  compute **Weight** for each node
  for nodes with the maximum **Weight**
    if $node_i$ can operate at V_{low} so that delay $\leq T_{spec}$
      (ΔPVS/Δdelay) = change in total power per unit delay by VS
      where ΔPVS is the reduction in power consumption due to voltage scaling technique and Δdelay is
      the decrease in the available slack
    if $node_i$ can be resized so that delay $\leq T_{spec}$
      if total power reduction ≥ 0
        (ΔPGS/Δdelay) = change in total power per unit delay by GS
        where ΔPGS is the reduction in power consumption due to gate sizing technique and Δdelay
        is the decrease in the available slack
    if (ΔPVS/Δdelay) $\geq$ (ΔPGS/Δdelay)
      apply VS on $node_i$
      update slacks on affected paths
    else
      apply GS on $node_i$
      update slacks on affected paths
    endfor
while (at least one node is changed)

Algorithm 2 proposes a linear time algorithm to calculate the number of paths which is used to calculate the Weight function to choose the candidate nodes for VS or GS.

Now we prove that the above algorithm indeed gives the number of paths passing through a node. Consider the number of paths *entering* a particular node. Each of these paths must either pass through one of its predecessor or originate at one of its predecessors. Moreover, a path passing through a node has a unique predecessor along the path as the graph is acyclic. Hence the number of paths entering a node is the sum of all paths going through or originating at its predecessors. A similar argument applies for paths leaving a node. Each path leaving a node must pass through or terminate at a successor. The number of entering paths for each node is computed by visiting the nodes in a topologically sorted order and assigning the number of paths as the summation of the number of paths through the predecessor nodes or originating at a predecessor node in case they are primary inputs. The same algorithm can be applied to calculate the number of paths leaving a node by reversing the edges and applying a topological sort starting from the primary outputs. Now the total number of paths going through a node is the number of ways to enter the node times the number of ways to leave the node, *i.e.*, product

of the number of entering paths and paths leaving the node.

ALGORITHM 2 *Calculation of number of paths through a node*

Input Directed Acyclic Graph $G(V, E)$
Output Number of paths passing through each node $v \in V$
for all $v \in V$
  if (v is primary input)
    incoming_paths[v] = 1;
  if (v is primary output)
    outgoing_paths[v] = 1;
Topologically sort vertices of $G(V, E)$.
for each $v \in V$ other than primary i/o in topological sorted order
  incoming_paths[v] = $\Sigma_{u \in pred(v)}$ incoming_paths [u];
Reverse edges and topologically sort vertices of $G(V, E)$
for each $v \in V$ other than primary i/o in topological sorted order
  outgoing_paths[v] = $\Sigma_{u \in pred(v)}$ outgoing_paths [u];
for each $v \in V$ other than primary i/o
  paths_going_through[v] = incoming_paths[v] × outgoing_paths[v];

Since the calculation of the number of paths that pass through each node requires a traversal of the graph in topological sorted order, the time complexity for number of paths calculation is $O(E)$, where E is the number of edges. This computation is required only once in the beginning of the algorithm as the number of paths passing through a node does not change. The time complexity for slack calculation for affected paths in each iteration of the *for* loop in Algorithm 1 is $O(V)$, assuming the nodes are already in topological sorted order. The body of the *for* loop in Algorithm 1 is executed whenever a node is sized or scaled. Hence the maximum number of time the *for* loop body is executed is $O(V)$ as each node is scaled or sized only once. Therefore the time complexity of the algorithm is $O(E+V*V) = O(V^2)$. Note that the time complexity of the combined VS and GS sizing algorithm using the MWIS approach is $O(rV^3)$, where r is the number of iterations executed by the algorithm. Hence, the proposed heuristic is orders of magnitude faster than the MWIS approach.

6. EXPERIMENTAL RESULTS

The experimental setup consists of the combined voltage scaling and gate sizing algorithm implemented in the environment of SIS. Experiments were carried out on a set of MCNC benchmark circuits. Before running our Algorithm 1 for voltage scaling and gate sizing, we performed technology mapping on the given circuit for the *mosis08.genlib* library under minimum delay mode with SIS and used this delay as the timing constraint, both for voltage scaling and gate sizing. The algorithm is implemented on nodes with a higher weight function as defined by Eq. (19). This ensures that maximum number of nodes are chosen for gate sizing. According to Algorithm 1, since only gates that do not violate the timing constraints on any path after down sizing or voltage scaling are accepted, there is no need for a post-processing step to resolve nodes with negative slacks. The power consumption was estimated based on the clock frequency of 100 MHz, threshold voltage of 1 V and supply voltage of $V_{high} =$ 5.0 V and $V_{low} = 3.5$ V. Exact values of change in transition times was calculated using Eq. (16) through Eq. (18). The constants were evaluated assuming a 0.18 micron technology and a unit-sized gate's input capacitance equal to 0.097 fF and output resistance equal to 23.8 kΩ [15].

Table I shows the percentage reduction in total power using only voltage scaling technique. We see a power reduction of about 50% for circuit 9symml when the total power is equal to the dynamic power and about 58% when short-circuit power is also considered during the decision. Table II shows the percentage reduction in total power using only gate sizing technique when all

TABLE I Power reduction using VS technique only

Circuit	#Total gates	#of V_{low} gates	% Reduction in power (dynamic power)	%Reduction in power (dynamic+short-(circuit power)
9symml	157	147	51.00	58.48
C1908	540	481	50.99	58.19
C880	384	297	50.73	58.00
apex7	307	156	50.93	58.30
b9	166	103	50.86	57.54
frg1	124	92	50.78	57.45
frg2	1438	1152	50.56	58.32
i1	89	48	51.00	58.75
i3	252	114	51.00	58.23
i5	505	306	51.00	58.49
i6	701	496	51.00	59.03
i7	828	558	41.12	58.63
rot	777	535	51.00	58.07
term 1	364	320	51.00	58.21

TABLE II Power reduction using GS technique only

Circuit	#of gates	% Reduction in power (dynamic power)	% Reduction in power (dynamic+short-(circuit power)
9symml	157	47.78	54.58
C1908	540	52.44	55.55
C880	384	57.61	60.18
apex7	242	56.98	60.06
b9	166	41.57	45.98
frg2	1438	54.20	48.89
i1	89	62.47	63.68
i3	252	59.99	60.19
i6	701	56.99	61.12
i7	828	51.30	57.10
rot	777	48.95	49.23
term 1	364	46.52	51.99

gates operate on a single supply voltage. Figure 4 shows the percentage reduction in power using gate sizing graphically. We see a power reduction of about 47% for circuit 9symml when the total power is equal to the dynamic power and about 54% when short-circuit power is also considered during the decision. Figure 5 shows that a combined VS and GS approach gives more power reduction than only VS. Table III gives the percentage power reduction using our combined VS and GS technique. A power reduction of as high as 80% is obtained for circuits like i1. The percentage power reduction is very high as the algorithm finds out the maximum number of nodes that are candidates for either VS or GS and do not violate the timing constraints. We can conclude that though VS and GS individually give us high power reduction, we can get much higher reduction by using a combined approach as the slacks which are unutilized by one technique can be used by the other technique. We have not considered the effect on power of the additional level converters that would be introduced due to the dual voltages in the circuit. Figure 1 shows that

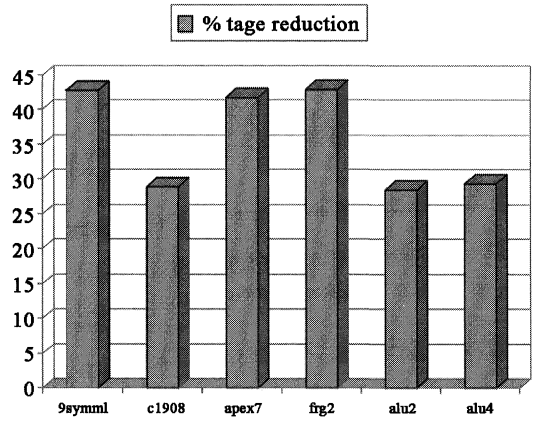

FIGURE 4 Percentage power reduction with gate sizing technique.

down sizing a gate might not always result in total power reduction. Hence, a decision taken with only the dynamic power into consideration would be less accurate. We can see from Figure 6 that an additional power reduction of as high as 6% can be got by taking the short-circuit power in the decision process. The improvement in power reduction depends on the number of implementations of the gates in the library. [12] defines completeness of a gate library for gate sizing. A more complete library would definitely improve the flexibility of the algorithm. The execution time of our algorithm using our fast heuristic for circuit *C1908* is 85.87 seconds. The execution time using

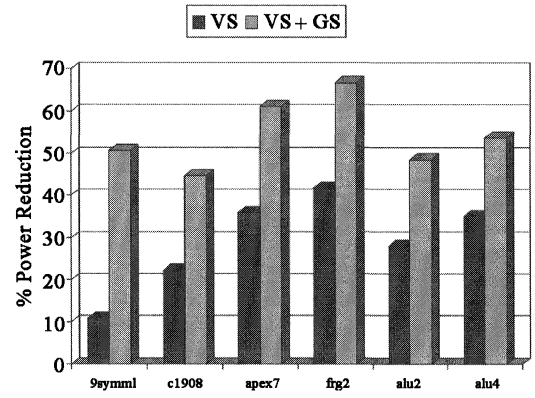

FIGURE 5 Percentage power reduction with VS and with our combined VS and GS algorithm.

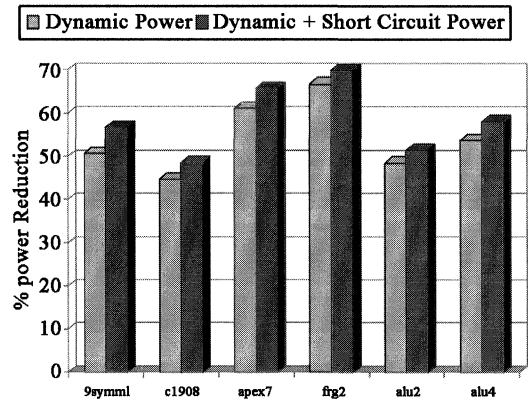

FIGURE 6 Power reduction for combined VS and GS with and without short-circuit power.

TABLE III Power reduction using VS and GS

Circuit	#Total gates	#of V_{low} gates	% Reduction in dynamic power	% Reduction in dynamic+short-circuit power
9symml	157	136	70.27	73.6
C1908	540	410	74.85	77.08
C880	384	264	77.46	80.46
apex7	307	205	76.98	80.95
b9	166	93	69.63	74.06
frg1	124	87	68.04	69.94
frg2	1438	1152	77.45	80.67
i1	89	42	81.3	84.13
i3	252	82	70.69	74.69
i5	505	303	78.54	81.62
i6	701	495	75.2	77.57
i7	828	560	68.33	74.05
rot	777	520	73.23	75.50
term 1	364	288	69.36	70.93
average			73.66	76.80

the MWIS approach [6] is reported as 117.7 seconds for *Library A*, 136.6 seconds using *Library B*, 256.6 seconds using *Library C* and 1485.7 seconds using *Library D*. We are not reporting a complete comparison with the combined VS and GS technique using a MWIS approach as the gate libraries used by them was different than what was available to us. But, from the execution times and the complexity analysis presented in Section 5, it can be concluded that out algorithm is much faster than the MWIS algorithm.

7. CONCLUSION

We have presented an effective framework for integrating voltage scaling and gate sizing techniques for getting maximum power reduction. We have proposed a fast algorithm for choosing the maximum number of gates for voltage scaling and gate sizing. We have used a better model for short-circuit power dissipation and shown that the combined voltage scaling and gate sizing generates an average power saving of 77%, which is greater than the power reduction achieved when the decisions are taken with only dynamic power.

References

[1] Chandrakasan, A. and Brodersen, R. (1995). *Low-Power CMOS Digital Design*, Kluwer Academic Publishers.

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

[3] Chang, J. M. and Pedram, M., Energy minimization using multiple supply voltages, *IEEE Transactions on VLSI Systems*, **5**(4), 1–8, December, 1997.

[4] Usami, K. and Horowitz, M., Cluster voltage scaling technique for low power design, *International Symposium on Low Power Design*, pp. 3–8, April, 1995.

[5] Usami, K. *et al.* (1997). Automated low power technique exploiting multiple supply voltages applied to a media processor, *Custom Integrated Circuit Conference*, pp. 131–134.

[6] Chen, C. and Sarrafzadeh, M., An effective algorithm for gate-level power-delay tradeoff using two voltages, *International Conference on Computer Design*, pp. 222–227, October, 1999.

[7] Raje, S. and Sarrafzadeh, M. (1997). Scheduling with multiple voltages, Integration, *VLSI Journal 23*, pp. 37–59.

[8] Usami, K. *et al.*, Design methodology of ultra low-power MPEG4 codec core exploiting voltage scaling techniques, *ACM/IEEE Design Automation Conference*, pp. 483–488, June, 1998.

[9] Shyu, J. M., Sangiovanni-Vincentelli, A. L., Fishburn, J. and Dunlop, A., Optimization-based transistor sizing, *IEEE Journal of Solid-State Circuits*, **23**, 400–409, Apr., 1998.

[10] Sapatnekar, S. S., Rao, V. B., Vaidya, P. M. and Kang, S. M., An exact solution to the transistor sizing problem for CMOS circuits using convex optimization, *IEEE Transactions on Computer-Aided Design*, **12**, 1621–1634, Nov., 1993.

[11] Berkelaar, M. R. and Jess, J. A. (1990). Gate Sizing in MOS digital circuits with linear programming, *Proceedings of the European Design Automation Conference*, pp. 217–221.

[12] Chen, C. and Sarrafzadeh, M., *Power Reduction by Simultaneous Voltage Scaling and Gate Sizing*, Asia Pacific DAC 2000, pp. 333–338.

[13] Chandrakasan, A., Sheng, S. and Brodersen, R., Low-power CMOS digital design, *Journal of Solid-State Circuits*, **27**(4), 473–484, April, 1992.

[14] Sapatnekar, S. S. and Chuang, W., *Power-Delay Optimizations in Gate Sizing*.

[15] Jason Cong, Zhigang Pan, Lei He, Cheng-Kok Koh and Kei-Yong Khoo, Interconnect Design for Deep Submicron ICs, *International Conference on Computer-Aided-Design*, pp. 478–485, Nov., 1997.

[16] Prasad, S. C. and Roy, K. (1994). Circuit optimization for minimization of power consumption under delay constraint, *Proc. of International Workshop on Low Power Design*, pp. 15–20.

[17] Igarashi, M. *et al.* (1997). A low power design method using multiple supply voltages, *Proc. of International Symposium on Low Power Electronics and Design*, pp. 36–41.

[18] Sundararajan, V. and Parhi, K. K. (1999). Synthesis of Low Power CMOS VLSI circuits using dual supply voltages, *Proc. of ACM/IEEE Design Automation Conference*, pp. 72–75.

[19] Chen, C. and Sarrafzadeh, M. (1999). Provably Good Algorithm for Low Power Consumption with Dual Supply Voltages, *Proc. of International Conference on Computer-Aided-Design*, pp. 76–79.

[20] Chen, C., Yang, X. and Sarrafzadeh, M. (2000). Potential Slack: An Effective Metric of Combinational Circuit Performance, *Proc. of International Conference on Computer-Aided-Design*.

[21] Fiduccia, C. M. and Mattheyses, R. M. (1982). A linear time heuristic for improving network partitions, *Proc. of ACM/IEEE Design Automation Conference*, pp. 175–181.

Authors' Biographies

Anshuman Nayak received his Bachelor's degree in Electronics and Electrical Communication Engg. from the Indian Institute of Technology in 1998 and his Masters in Electrical and Computer Engg. from Northwestern University. He is currently

pursuing is Ph.D. at Northwestern University. His research interests include system level design tools, logic synthesis, embedded systems and reconfigurable computing.

Malay Haldar received his Bachelor's degree in Computer Science and Engg. from the Indian Institute of Technology in 1998 and his Masters in Electrical and Computer Engg. from Northwestern University. He is currently a doctoral student at Northwestern University. His research interests include system level design tools, embedded systems and reconfigurable computing.

Prithviraj Banerjee received his B.Tech. degree in Electronics and Electrical Engineering from the Indian Institute of Technology, Karagpur, India, in August 1981, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Illinois at Urbana-Champaign in December 1982 and December 1984 respectively. Dr. Banerjee is currently the Walter P. Murphy Professor and Chairman of the Department of Electrical and Computer Engineering, and Director of the Center for Parallel and Distributed Computing. ar Northwestern University in Evanston, Illinois. Prior to that he was the Director of the Computational Science and Engineering program, and Professor of Electrical and Computer Engineering and the Coordinated Science Laboratory at the University of Illinois at Urbana-Champaign. Dr. Banerjee's research interests are in Parallel Algorithms for VLSI Design Automation, Distributed Memory Parallel Compilers, and Compilers for Adaptive Computing, and is the author of over 270 papers in these areas. Dr. Banerjee has received numerous awards and honors during his carrer. He became a Fellow of the ACM in 2000. He was the recipient of the 1996 Frederick Emmons Terman Award of ASEE's Electrical Engineering Division sponsored by Hewlett-Packard. He was elected to the Fellow grade of IEEE in 1995. He received the University Scholar award from the University of Illinois for in 1993, the Senior Xerox Research Award in 1992, the IEEE Senior Membership in 1990, the National Science Foundation's Presidential Young Investigators' Award in 1987, the IBM Young Faculty Development Award in 1986, and the President of India Gold Medal from the Indian Institute of Technology, Kharagpur, in 1981.

Chunhong Chen received the Ph.D. degree in electrical engineering from the Fudan University, Shanghai, China, in 1997. He is currently a postdoctoral fellow at Northwestern University, Evanston, IL. From 1997 to 1998, he was with the Hong Kong University of Science and Technology as a Research Associate. His current research focus is on logic-level and high-level synthesis for low power.

Majid Sarrafzadeh received his B.S., M.S. and Ph.D. in 1982, 1984, and 1987 respectively from the University of Illinois at Urbana-Champaign in Electrical and Computer Engineering. He joined Northwestern University as an Assistant Professor in 1987. Since 1997 he has been a Professor of Electrical Engineerng and Computer Science at Northwestern University. His research interests lie in the area of VLSI CAD, design and analysis of algorithms and VLSI architecture. Dr. Sarrafzadeh is a Fellow of IEEE for his contribution to "Theory and Practice of VLSI Design". He received an NSF Engineering Initiation award, two distinguished paper awards in ICCAD, and the best paper award for physical design in DAC for his work in the area of High-Speed VLSI Clock Design. He has served on the technical program committee of numerous conferences in the area of VLSI Design and CAD, including ICCAD, EDAC and ISCAS. He has served as committee chairs of a number of these conferences, including International Conference on CAD and International Symposium on Physical Design. He will be the general chair of the 1998 International Symposium on Physical Design. Professor Sarrafzadeh has published approximately 150 papers, is a co-editor of the book "Algorithmic Aspects of VLSI Layout" (1994 by World Scientific), co-author of the book" An Introduction to VLSI Physical Design" (1996 by McGraw Hill), and the author of an invited chapter in Encyclopedia of Electrical and Electronics Engineering in the area of VLSI Circuit Layout. This is planned for publication in

1997 by John Wiley & Sons, Inc. Dr. Sarrafzadeh is on the editorial board of the VLSI Design Journal, co-editor-in-chief of the International Journal of High-Speed Electronics, and an Associated Editor of IEEE Transactions on Computer-Aided Design. Dr. Sarrafzadeh has collaborated with many industries in the past ten years including IBM and Motorola.