Power-Delay Optimizations in Gate Sizing [1]

Sachin S. Sapatnekar
Department of Electrical & Computer Engineering
University of Minnesota
Minneapolis, MN 55455.

Weitong Chuang[2]
Macronix Semiconductor Company
Science-based Industrial Park
Hsinchu, Taiwan 300.

Abstract

The problem of power-delay tradeoffs in transistor sizing is examined using a nonlinear optimization formulation. Both the dynamic and the short-circuit power are considered, and a new modeling technique is used to calculate the short-circuit power. The notion of transition density is used, with an enhancement that considers the effect of gate delays on the transition density. When the short-circuit power is neglected, the minimum power circuit is identical to the minimum area circuit. However, under our more realistic models, our experimental results on several circuits show that the minimum power circuit is not necessarily the same as the minimum area circuit.

[1] This work was supported in part by NSF under contract MIP-9502556.
[2] formerly at AT&T Bell Laboratories, Murray Hill, NJ.

Content Indicators: B.7.2 [Integrated Circuits]: Design Aids: layout.
Keywords: transistor sizing, power estimation, VLSI layout, algorithms, performance.

1 Introduction

With the emergence of portable products and concerns about cooling costs for computers, power dissipation has emerged as a major design consideration, and considerable research effort has been expended in trying to find power-efficient solutions to circuit design problems. One such procedure that is applied at the logic or transistor level is the procedure of gate sizing, which 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 [1–4]. There has been relatively less work on incorporating power considerations into sizing.

The sizing problem can be described as follows. During the optimization of a circuit, it must be ensured that the worst-case delay of each combinational stage is restricted to be below a certain specification. Given a CMOS circuit topology, the delay can be controlled by varying the sizes of transistors in the circuit.[3] Improvements in the timing behavior of a circuit can be achieved by increasing the sizes of some transistors in the circuit from the minimum allowable size, and these improvements are made at the expense

[3]Here, the size of a transistor is measured in terms of its channel width, since area and power considerations dictate that the channel lengths must be kept at the minimum allowable size.

of additional chip area. However, increasing the sizes of transistors in a circuit increases the circuit area and often (but not always, as will be shown later) also leads to increased power dissipation, and hence an optimization problem must be solved to arrive at an optimal set of transistor sizes that gives an acceptable tradeoff.

In this work, we first examine the properties of the power-delay sizing problem as a nonlinear optimization problem under an accurate short-circuit power model. This method also incorporates an accurate method for calculation of transition densities for switching activity measurement.

Various formulations of the sizing problem may be considered, with one of area, delay or power constituting the objective function, and with constraints on the other two. One practical formulation that we use in this paper recognizes that a designer's objective is to achieve the best performance at a given clock period and may be stated as

$$\begin{array}{rc} \text{minimize} & Power(\mathbf{w}) \\ \text{subject to} & Delay(\mathbf{w}) \leq T_{spec} \\ & Area \leq A_{spec} \\ and & \text{Each gate size} \geq \text{Minsize} \end{array} \tag{1}$$

where both $Delay$ and $Power$ are functions of the gate sizes, $\mathbf{w} \in \mathbf{R}^n$ (where n is the number of gate sizes), T_{spec} and A_{spec} are, respectively, the constraints on the circuit delay and area, and Minsize is the minimum gate size allowed by the technology.

Note that the optimization problem specified in (1) must be solved on one combinational subcircuit at a time, and therefore, although the entire circuit may have a million or more gates, the number of variables in the sizing problem will be comfortably small.

Previous approaches that have taken power considerations into account during transistor sizing include [4–8]. The approach in [5] utilizes the idea that the power of a circuit was a weighted function of the transistor sizes; however, since it optimizes one path at a time, the approach may lead to suboptimal solutions. A subsequent approach in [6] uses a heuristic method and incorporates transition densities [9] into the sizing process. The algorithms in [4, 7] present a linear programming approach to exploring the power-delay-area tradeoff for a CMOS circuit; we use more accurate nonlinear Elmore delay models in this work. Another linear programming based approach is presented in [8].

All of the above approaches consider the dynamic power dissipation only, and neglected the role of the short-circuit power. However, this is not always a valid assumption. The idea that the short-circuit power accounts for under 20% of the total power in a "well-designed" circuit is a valid one, but intermediate circuit parameters obtained during the course of an optimization may not correspond to well-designed circuits. These could, for example, lead to incorrectly computed gradients in a gradient-based optimization scheme. The role of short circuit power has been considered in [10,11] using a short-circuit power measurement formula proposed in [12]. However, the inaccuracies in this formula due to its limited assumptions could lead to incorrect optimization results.

We will see that minimizing a power function that considers only the dynamic power, without any constraints on the 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 is more pronounced when large loads are being driven. An example to illustrate this fact are described in Section 3.3. Therefore, a more accurate procedure for estimating the short-circuit power is called for, and one such method is proposed here.

In this work, we utilize Elmore time constants [13] and estimate the circuit delay and power dissipation based on this timing model. We use curve-fitting to determine a short-circuit power dissipation model, and formulate the sizing problem as a nonlinear optimization problem. This formulation does not have the property of convexity, and therefore we use a general nonlinear programming algorithm to arrive at a solution.

In our approach, each gate is characterized by two sizes, W_n and W_p, corresponding to the n and p device sizes in the gate. As a notational comment, it is worth mentioning here that the term "gate sizes" will henceforth be used to refer collectively to the W_n and W_p values for all gates in the circuit. These two variables are the fundamental sizing variables for each gate in the circuit. We note that further performance improvements are attainable by individually adjusting the sizes of each transistor; however, the argument in favor of using uniform sizes in a gate is that the impact on the final layout is low. Moreover, since the area measures used here are consistent with such a layout, the performance of the final circuit after layout is more predictable.

The contributions of this work include appropriate modeling of the power dissipation, including short-circuit dissipation, for accuracy and ease of optimization. We show that power and delay are not necessarily conflicting objectives; one can sometimes take a minimum area circuit and increase the sizes of transistors in the circuit to reduce its power dissipation, as reported in [10]. Also, the minimum power circuit may sometimes have a delay that is larger than the minimum sized circuit delay.

The paper is organized as follows. Section 2 illustrates the gate-level model used in this work, followed by a description of the power and delay models in Sections 3 and 4, respectively. The optimization scheme is described in Section 5, and experimental results are presented in Section 6, followed by concluding remarks in Section 7.

2 Gate-level Modeling of a Circuit

In this approach, we solve the problem at the gate level, where each gate is reduced to an equivalent inverter whose n- and p- transistor sizes are, respectively, W_n and W_p. The optimal values of the gate sizes may be found by solving an optimization problem, and these may be mapped back to the sizes of individual transistors in the gate. One of the advantages of using gate sizes, W_n and W_p, instead of transistor sizes as fundamental variables for optimization is that it is easy to achieve regular layouts with such a procedure. Moreover, as a consequence of this, the area function used to estimate the circuit area is more accurate than those that have been used for transistor sizing in [1] and many other papers. Note that the n-transistors and the p-transistors are sized independently.

A CMOS gate can be represented by an equivalent inverter with n- and p- transistor sizes of W_n and W_p, respectively. Each transistor of size x in the inverter is modeled by a resistance whose value is given by

$$R_{on} = \begin{cases} K_R/x \text{ when the transistor is on} \\ \infty \text{ when the transistor is off} \end{cases} \tag{2}$$

where the value of K_R is different for the n- and p-type transistors. Additionally, each transistor has associated with it the parasitic capacitances, C_s, C_d and C_g, at its source, drain and gate, respectively. The dependence of these on the transistor size x is

$$C_s = K_{sd} \cdot x + K'_{sd} \tag{3}$$

$$C_d = K_{sd} \cdot x + K'_{sd} \tag{4}$$

$$C_g = K_g \cdot x + K'_g \tag{5}$$

Note that the symmetry of the transistor dictates that the proportionality constants for the source and drain parasitic capacitances are equal.

The wiring capacitance driven by a gate with f fanouts are estimated as in [2] using the expression

$$C_{wiring} = \text{Parasitic} \cdot (f + 0.5) \tag{6}$$

where Parasitic is a prespecified constant. Note that this is an educated estimate of the parasitic capacitances. If a more accurate estimate is available, it may be used instead.

3 Computation of the Power Dissipation of a Circuit

The power dissipation of each gate in a circuit is the sum of two components: the *dynamic power* and the *short-circuit power*. The power associated with the leakage current is assumed to be negligible at current supply voltage values and is not considered. In ultra deep submicron CMOS, leakage current will have to included in the optimization formulation; leakage current models are given in [14].

3.1 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_L, operating under a clock frequency f and having a probability p_T of switching is given by

$$P_{dynamic} = C_l \cdot V_{dd}^2 \cdot f \cdot p_T \tag{7}$$

where V_{dd} is the supply voltage.

3.2 Short-circuit Power Dissipation

Unless the waveform that induces switching in a gate is an ideal step waveform, there is a second component of the power dissipation, referred to as the short-circuit power. Since real-life waveforms have nonzero transition times, the short-circuit power could have a significant contribution to the total power.

During the switching of an inverter, when the input voltage value is between V_T and $V_{dd} - V_T$, where $|V_T|$ is the magnitude of the threshold voltage, both the n- and the p-transistors are on and provide a path for current to flow directly between V_{dd} and ground. This current is referred to as the short-circuit current, and the corresponding wasteful power dissipation is referred to as the short-circuit power.

Most transistor sizing methods have considered only the dynamic power dissipation. Recently, a few methods [10, 11] have also considered the contribution short-circuit dissipation, using the following formula for the short-circuit power from [12]:

$$P_{sc} = \frac{\beta}{12} \left(V_{dd} - 2V_T\right)^3 \cdot \tau \cdot f \cdot p_T \tag{8}$$

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.

As we will see later, this formula suffers from some shortcomings, and a different short-circuit power model is used in this work.

3.3 Limitations of the Short-Circuit Power Model

The formula in (8) was derived for an inverter, and any other gate can be handled by considering its equivalent inverter. As seen above, during the transition at the input to an inverter, one transistor is in the process of turning on, while the other is being turned off. Equation (8) is predicated on the following assumptions on an inverter gate:

- the inverter is symmetrical, i.e., $\beta_n = \beta_p = \beta$.

- the inverter has zero load capacitance.

- the transistor that is turning off is in saturation during the transition.

While these assumptions were adequate for the purpose of the work in [12], they can cause errors when the formula (8) is used in the context of transistor sizing. Specifically, during sizing, the first assumption is typically not true, and the second assumption can cause errors in estimating short-circuit power. For example, it has been demonstrated in the example in Figure 4 of [12] that as the load capacitance at the inverter output is increased from zero to 1pF, under the technology parameters used, the short circuit power decreases by a factor of about 4.

A minimum sized circuit corresponds to the minimum power circuit when the short-circuit current is neglected. The reason for this is that the dynamic power function is a linear function (with positive coefficients) in the device sizes, which implies that minimizing the dynamic power without any delay constraints would cause all devices in the circuit to be minimum-sized.

As shown in [10, 11], when the short-circuit power is not negligible, the minimum power circuit is not necessarily minimum-sized. The rationale behind this is illustrated in Figure 7. If gate G drives a large load, then the waveform at its output will probably have a large transition time. This implies that its fanout gates may have a significantly large short-circuit power dissipation since the direct path from V_{dd} to ground through the n- and p-transistors is on for a longer duration (this corresponds to a larger value of τ in (8)). If the gate G were to be made slightly larger, its delay could be reduced, causing a reduction in this short-circuit power, with an accompanying increase in the dynamic power since the capacitances of the transistors in G have been increased by sizing. An appropriate balance must be found at which the two components of power trade off. Note that this balance is critically dependent on the accurate measurement of the short-circuit power, motivating the need for the model that we will now present.

3.4 A Short-Circuit Power Model

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

Consider, for a moment a rising transition at the output of the inverter, corresponding to a falling transition at the input, as shown in Figure 7(a). The short circuit current corresponds to the current through the n-transistor. A schematic of the inverter is shown in Figure 7(b). The n- and p-transistors have been replaced by (nonlinear) admittances Z_N and Z_P, respectively, that increase as w_n and w_p are increased, and the capacitance is shown by a third admittance, Z_C.

If Z_N is increased and Z_P and Z_C are kept constant, then the amount of current that Z_N can carry increases, and the short-circuit current through

Z_N increases. If Z_P is increased while other admittances are kept constant, then a similar effect is seen. If Z_C increases, then a larger fraction of the current flowing in from V_{dd} through Z_P is diverted away from Z_N, and therefore, the short-circuit current decreases. Note that the short-circuit power is simply the short-circuit current multiplied by V_{dd}, and hence the implications on the power show the same trend. Similar observations can be made for the short-circuit power during the output falling transition.

Therefore, as w_n or w_p increase, the short-circuit power increases, and as C increases, the short-circuit power decreases. We used SPICE simulations to generate a large number, m, of data points (we chose $m = 160000$) for different values of w_n, w_p, C and τ and attempted to fit the following function to the data:

$$P_{sc} = \sum_{i=1}^{p} \alpha_{1,i} w_n^{\alpha_{2,i}} w_p^{\alpha_{3,i}} C^{\alpha_{4,i}} \tau^{\alpha_{5,i}} \tag{9}$$

$$\alpha_{1,i}, \alpha_{2,i}, \alpha_{3,i}, \alpha_{4,i}, \alpha_{5,i} \in \mathbf{R} \text{ for } i = 1, \cdots, p \tag{10}$$

This function is more general than a polynomial since the exponents may be arbitrary real numbers, and not necessarily integers.

Given the measured set of data points, $w_{n,j}, w_{p,j}, C_j, \tau_j, P_{sc,j}, j = 1, \cdots, m$, a sequential quadratic programming (SQP) package [15] was used to fit the data using the least squares criterion by solving the following optimization problem to find the values of $\alpha_{1,i} \cdots \alpha_{5,i}$:

$$\text{minimize} \sum_{j=1}^{m} \sum_{i=1}^{p} \left[\frac{\alpha_{1,i} w_{n,j}^{\alpha_{2,i}} w_{p,j}^{\alpha_{3,i}} C_j^{\alpha_{4,i}} \tau_j^{\alpha_{5,i}} - P_{sc,j}}{P_{sc,j}} \right]^2 \tag{11}$$

Note that the function is normalized by dividing by $P_{sc,j}$ so that each data point has the same importance. This optimization must be carried out once for a given set of process parameters. The actual optimization problem takes mere minutes of CPU time, but the generation of m data points using SPICE is highly computationally intensive.

Although the reader should be warned that our results here were dependent on the specific parameters used for simulation, it is worth listing the results of our optimization procedure. For the output falling transition, the short-circuit power parameters was observed to vary as

$$P_{sc,\text{fall}} \propto w_n^{0.69} w_p^{0.88} C^{-0.10} \tau^{1.49} \tag{12}$$

with a 10.2% average least-squares error. For the rising transition, the calculated values of the parameters were

$$P_{sc,\text{rise}} \propto w_n^{0.75} w_p^{0.82} C^{-0.085} \tau^{1.49} \tag{13}$$

with a 9.2% average least-squares error.

A few prominent observations about these results are in order:

- $p = 1$ provided sufficient accuracy, and no additional gains were achievable by using larger values of p.

- For both transitions, the relative effect of w_p is larger than that of w_n. This is logical, since the mobility in p-transistors is lower than in n-transistors, with its corresponding implications on admittance.

- For the output rising transition, the relative contribution of w_n on the short circuit power as compared to w_p is larger than for the output falling transition. This is also logical since for the rising transition, the

p transistor carries both the short-circuit current and the capacitive charging current, while the n transistor only carries the short-circuit current. Therefore, increasing the admittance of the n transistor is likely to have a more pronounced effect on the power.

The short-circuit formula requires knowledge of w_n, w_p, C and τ. The first three of these can be calculated easily, and we will not illustrate how the fourth is calculated. Consider a gate, G, that is driven by another gate, G_1. The transition time of the waveform at the output of gate G_1 may be modeled as twice its Elmore delay, since the Elmore delay is the time required by the signal at its output to reach 50% of its final value. Such an approach has been used in [3] and has been shown to be accurate.

3.5 Computing Transition Densities

The power consumption of a chip is directly related to the extent of its switching activity, i.e., the rate at which its nodes are switching. However, estimating this activity has been very difficult because it depends on the specific signals being applied to the circuit. For large circuits that consist of hundreds of thousands of gates, it is impossible to simulate for all possible inputs. Recently, an estimation technique based on probabilistic simulation approach has drawn much attention [9, 16–18].

In [9], Najm introduced a measure of switching activity that is called *transition density*. The transition density may be defined as the "average switching activity" at a circuit node. Given the signal probability and transition density of each primary input, the algorithm propagates those values through logic modules. However, correlations between internal lines due to reconvergence are ignored during propagation. In [16], symbolic simulation is used to produce a set of Boolean functions which represent conditions for switching at each gate at a specific time instance. The transition probability at a gate is calculated by performing a linear traversal of the Ordered Binary Decision Diagrams. Due to the use of OBDD, however, the procedure suffers from excessive computation time and storage space. In [17], a tagged transition waveform is proposed to approximate the correlation between two lines by the correlation between the steady state values of these lines.

The method in [9] is amenable for use in optimization due to its ease of implementation. However, it may be optimistic, as pointed out by Najm in [18], where the effects of *filtration* in real circuits were discussed. This is essentially due to inertial delays of logic gates which in practice, cause most glitches to be suppressed. Very short pulses are *filtered out* because the module is not fast enough to respond to them. In order to model this filtration effect of the circuit inertial delays, a new delay block called a *filter block* is introduced. With such a filtration mechanism, more accurate and realistic results can be obtained.

As a consequence, we choose the approach as described in [9] and [18] to estimate transition density. In the following, we outline the procedures. For more details on filter mechanism, the reader is referred to [9, 18].

The chief difference between this approach and previous approaches to using transition density estimates during sizing is that this method considers the effect of gate delays on the transition densities.

Let $P(\mathbf{x})$ denote the *equilibrium probability* of a logic signal $x(t)$, i.e., $P(\mathbf{x}) \equiv P(x(t) = 1)$. It can be shown that

$$P(\mathbf{x}) = \lim_{T\to\infty} \int_{-T/2}^{T/2} x(t)dt \tag{14}$$

This gives the fraction of time that the signal $x(t)$ is high. Let $\mathbf{n_x}(T)$ denote the number of transitions of $x(t)$ in $(-T/2, T/2]$. Then the transition density of $x(t)$, $D(x)$, can be is defined to be

$$D(x) = \lim_{T\to\infty} \frac{\mathbf{n_x}(T)}{T} \tag{15}$$

It has been shown that, if $\mathbf{y} = \mathbf{f}(\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_n})$ is a Boolean function and the inputs $\mathbf{x_i}$'s are independent, then the density of output $\mathbf{y}$ is given by:

$$D(\mathbf{y}) = \sum_{i=1}^{n} P(\frac{\partial \mathbf{y}}{\partial \mathbf{x_i}}) D(\mathbf{x_i}) \tag{16}$$

where $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ is the Boolean difference of $\mathbf{y}$ with respect to $\mathbf{x}$ and is defined as

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \equiv \mathbf{y_{x=1}} \oplus \mathbf{y_{x=0}} \tag{17}$$

For simple gates (AND, OR, etc), the Boolean difference can be easily calculated. For more complex Boolean functions, the OBDD package can be used to evaluate the Boolean difference using equation (17). Finally, given the probability and density values at the primary inputs, a single pass over the circuit, using (16), gives the density value at every node.

At this point, the calculated transition density may be overestimated, especially for high-frequency circuits. To overcome such a problem, a conceptual low-pass filter is placed at the output of each logic module. Let the output of a logic module (thus the input of the low-pass filter) be $x(t)$. Let F be the filter block with input $x(t)$ and output $y(t)$. The behavior of F can be defined by a finite-state machine with four states as shown in Figure 7. The state S_0 (corresponding to $x = y = 0$) and S_3 (corresponding to $x = y = 1$) are *stable states*. The other two states, S_1 (corresponding to $x = 0, y = 1$) and S_2 (corresponding to $x = 1, y = 0$), are *unstable states*. The filter will stay in stable states indefinitely if x does not change. If the filter enters states S_1 (S_2), then it can stay there for at most τ_0 (τ_1). After the transition period, the filter will automatically transit to stable state S_0 (S_3); or it could fall back to S_3 (S_0) any time during this period if x switches back to 1 (0) again. With such a formal model, $P(y)$ and $D(y)$ can be computed from $P(x)$ and $D(x)$ using the formulations derived in [18].

4 Computation of the Circuit Delay

The Critical Path Method (CPM), (often incorrectly referred to in the literature as PERT [19]), a standard procedure used in the timing analysis of digital circuits, is used to compute the maximum overall rise and fall delays between the primary inputs and primary outputs of the circuit. A trace-back method is then used to obtain the *critical path*, which consists of the set of gates that lie on the largest delay path from a primary input to a primary output of the combinational network. Two numbers t_h and t_l are assigned to each gate output node in the circuit, corresponding to the total rise and fall delay from the primary inputs, respectively.

The rise and fall delays of each gate are taken to be the Elmore delays of RC networks that correspond to the rise and fall scenarios, respectively. These RC networks are easy to build; for the falling transition, they correspond to the resistance of the n-transistor driving the parasitic capacitances at the gate output. These parasitic capacitances correspond to the sum of the drain capacitances of the n- and the p-transistor in the equivalent inverter corresponding to the current gate, and the sum of the gate terminal capacitances of all gates that the current gate fans out to. For the rising transition, the only change is that the resistance of the p-transistor is used instead of that of the n-transistor.

5 Formulation of the Optimization Problem

In this section, we will temporarily assume that the switching probability at the output of each gate is independent of the gate sizes. This is, however, not a valid assumption, and we will remove this assumption later in this section.

The formal statement of the problem is as given by (1). It has been shown in previous work [11] that under the above assumption, that the power can be modeled as a posynomial function [20] of the gate sizes. (However, we note that the power model used there was that of Equation (8).) It is well known that the circuit delay is a maximum of posynomials [1]. This implies that under a simple transformation, the power may be mapped on to a convex function, and the delay on to a maximum of convex functions, which is also convex [21]. This is a convex optimization problem, with the property that any local minimum is also a global minimum.

Under the formulation presented here, the short-circuit power function is *not* a posynomial since C is a sum of capacitances, each capacitance being directly proportional to a gate size, and the exponent of C is negative. Note that this is unavoidable during modeling since P_{sc} decreases as C increases, and vice versa. This effect can be captured in the expression for P_{sc} either by

(a) using a negative coefficient for the C term, or

(b) using a negative exponent for C.

In fact, in our model, we see that the latter option gave the best fit. In the first case, we would clearly violate the criteria for posynomiality, while in the second, C, which is a weighted sum of gate sizes, is in the denominator, also preventing P_{sc} from being a posynomial function of the gate sizes.

The optimization problem is stated as:

$$\begin{array}{rc} minimize & Power(\mathbf{w}) \\ subject\ \ to & Delay(\mathbf{w}) \leq T_{spec} \\ & Area(\mathbf{w}) \leq A_{spec} \\ and & \text{Each gate size} \geq \text{Minsize} \end{array} \tag{18}$$

where $Delay(\mathbf{w})$ and $Power(\mathbf{w})$ are functions of the vector of gate sizes, $\mathbf{w}$. This is a general nonlinear optimization problem, and we use a sequential quadratic programming package [15] to solve it.

Therefore, we may solve the power-delay-area tradeoff for a *fixed* set of gate switching probabilities as described above. However, as pointed out in Section 3.5, the switching probabilities are dependent on the gate delays, which are, in turn, dependent on the gate sizes, and therefore the above assumption is not quite valid. Consequently, we use the following scheme to solve the gate sizing problem:

error = ∞;

Set all gates to Minsize;

Calculate gate delays;

Compute $\mathbf{p_T}$ = transition probability vector at each gate for current gate delays;

while (error $> \epsilon$) {

  old$\mathbf{p_T}$ = $\mathbf{p_T}$;

  Solve gate sizing (nonlinear optimization) problem assuming $\mathbf{p_T}$ above;

  Calculate gate delays corresponding to gate sizes calculated above;

  Compute $\mathbf{p_T}$ = vector of transition probabilities

    at each gate based on current gate delays;

  error = % change in power under $\mathbf{p_T}_i$ as compared to old$\mathbf{p_T}_i$

}

We first calculate the transition probabilities with all gates set to minimum size. Next, taking these transition probabilities to be fixed, we solve the gate sizing problem, which is a nonlinear programming problem under this assumption. However, the assumption may be invalid, since the gate delays affect the switching probabilities. Therefore we recompute the switching probabilities for the new set of gate delays, and continue the iterations until the switching probabilities converge. In practice, we see that convergence occurs in a very small number of iterations (no more than four iterations for the circuit examples that we tried).

The nature of the variations in the transition probabilities with gate delays are of the type that in many cases, it is not enough to calculate the transition probabilities for the minsized circuit and use them during the optimization. However, it is also true that the variations are small enough that the procedure above gives a good solution in a small number of iterations.

6 Experimental Results

The algorithm described above has been implemented as a C program on an HP735 workstation. Table 1 illustrates the power-delay tradeoff for various benchmark circuits under various delay specifications. The first column lists the circuit name. The delay, D_u, and area, A_u, for the circuit when all devices are minimum-sized are also shown, with the area being measured as the sum of transistor sizes. A timing specification is placed on the circuit, and it is optimized for the minimum power under that constraint. The corresponding power, as a fraction of the power P_u of the unsized circuit, and circuit area found by the algorithm are shown in the next three columns. The first result line for each circuit shows the minimum power circuit and its corresponding delay.

In most cases, it was found that the results were obtained in one or two iterations of the algorithm shown in Section 5. In a few rare cases, three or even four iterations were needed. The explanation for the small number of iterations is that while the transition densities are dependent on the delay, this dependence is quite weak. The value of ϵ used to control convergence was set to 0.001. The bulk of the computation is consumed by the optimization algorithm, and the CPU time required for transition density calculations is virtually negligible.

For each circuit, as the delay specification is made tighter, the minimal power dissipation required to achieve the specification increases nonlinearly and monotonically. Interestingly enough, the circuit area does not necessarily increase monotonically, as seen in circuit `cht`.

Traditional estimates of power for sizing purposes have considered only the dynamic component of power, given by Equation (7). The dynamic power function is a linear function (with positive coefficients) in the device sizes. Therefore, minimizing the dynamic power without any delay constraints implies that all devices in the circuit must be minimum-sized. However, when one considers the role of short-circuit current, this may not remain so. If a gate G drives a large capacitance, it will have a slow rise time. Therefore, the transition time at the input to any fanout gate is significant and the short-circuit power is noticeable. To optimize the power, it may be necessary to size G to reduce the transition time at its output, and therefore, the short-circuit current for any fanout gate of G.

A curious side-effect of this is seen for circuit `cht`, where the delay of the minimum power circuit is actually larger than that of the minimum area circuit. This arises because a gate that is on an off-critical path has a large delay, and it must be sized to reduce the short-circuit power of its fanouts. This sizing leads to a larger capacitance that must be driven by a gate on the critical path, leading to an increased delay.

A similar effect is also seen in circuit `cordic`, where the minimum power circuit has a larger area than the minimum area circuit, but has the same delay. In this case, some gates off the critical path were sized to minimize the power, but this sizing did not affect the critical path.

We caution the reader that the minimum area circuit is not *always* different from the minimum power circuit, and this was seen in several cases in the benchmark suite, particularly in the smaller circuits. However, it was found that for the larger circuits, the minimum area and minimum power points were distinct. This is consistent with the fact that larger circuits often have larger delays, which implies that the transition time at the input to some gates is liable to be relatively large, leading to more short-circuit power dissipation.

7 Conclusion

An algorithm for sizing with power considerations has been presented. The sizing problem for power-delay-area tradeoffs is formulated as a nonlinear optimization problem, and is solved using a sequential quadratic programming algorithm. Results on various benchmark circuits have been shown.

By considering the dynamic power dissipation alone while sizing, as has been commonly done, the minimum power circuit under no delay constraints corresponds to the minimum-sized circuit. It has been shown here that when the short-circuit power dissipation for a minimum area circuit is significant, as in the case where large interconnect capacitances or off-chip capacitances must be driven, the minimum power circuit does not correspond to the minimum sized circuit. In such a case, reduction of the circuit delay and power dissipation are not necessarily contradictory objectives, and a certain amount of delay reduction is available at no additional expenditure in power; in fact, the minimum power circuit is achieved by sizing some gates in the circuit. Additionally, it is also possible for the minimum power circuit to have a larger delay than the minimum area circuit.

References

[1] J. Fishburn and A. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," in *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 326–328, 1985.

[2] J.-M. Shyu, A. L. Sangiovanni-Vincentelli, J. Fishburn, and A. Dunlop, "Optimization-based transistor sizing," *IEEE Journal of Solid-State Circuits*, vol. 23, pp. 400–409, Apr. 1988.

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

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

[5] K. S. Hedlund, "AESOP : A tool for automated transistor sizing," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 114–120, 1987.

[6] C. H. Tan and J. Allen, "Minimization of power in VLSI circuits using transistor sizing, input ordering, and statistical power estimation," in *Proceedings of the 1994 International Workshop on Low Power Design*, pp. 75–80, 1994.

[7] M. R. C. M. Berkelaar, P. H. W. Buurman, and J. A. G. Jess, "Computing the entire active area/power consumption versus delay trade-off curve for gate sizing with a piecewise linear simulator," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 474–480, 1994.

[8] Y. Tamiya, Y. Matsunaga, and M. Fujita, "LP based cell selection with constraints of timing, area and power consumption," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 378–381, 1994.

[9] F. N. Najm, "Transition density, a stochastic measure of activity in digital circuits," in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 644–649, 1991.

[10] M. Borah, R. M. Owens, and M. J. Irwin, "Transistor sizing for low power CMOS circuits," *IEEE Transactions on Computer-Aided Design*, vol. 15, pp. 665–671, June 1996.

[11] S. S. Sapatnekar and W. Chuang, "Power vs. delay in gate sizing: Conflicting objectives?," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 463–466, 1995.

[12] H. J. M. Veendrick, "Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits," *IEEE Journal of Solid-State Circuits*, vol. SC-19, pp. 468–473, Aug. 1984.

[13] J. Rubenstein, P. Penfield, and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Transactions on Computer-Aided Design*, vol. CAD-2, pp. 202–211, July 1983.

[14] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design.* New York, NY: Oxford University Press, 1987.

[15] C. Lawrence, J. L. Zhou, and A. L. Tits, "User's Guide for CFSQP Version 2.4: A C Code for Solving (Large Scale) Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality Constraints," Tech. Rep. TR-94-16r1, Institute for Systems Research, University of Maryland, College Park, Feb. 1996.

[16] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of average switching activity in combinational and sequential circuits," in *Pro-*

ceedings of the ACM/IEEE Design Automation Conference, pp. 253–259, 1992.

[17] C.-Y. Tsui, M. Pedram, and A. M. Despain, “Efficient estimation of dynamic power consumption under a real delay model,” in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 224–228, 1993.

[18] F. N. Najm, “Low-pass filter for computing the transition density in digital circuits,” *IEEE Transactions on Computer-Aided Design*, vol. 13, pp. 1123–1131, Sept. 1994.

[19] T. M. Burks, K. A. Sakallah, and T. N. Mudge, “Critical paths in circuits with level-sensitive latches,” *IEEE Transactions on VLSI Systems*, vol. 3, pp. 273–291, June 1995.

[20] J. Ecker, “Geometric programming: methods, computations and applications,” *SIAM Review*, vol. 22, pp. 338–362, July 1980.

[21] D. G. Luenberger, *Linear and Nonlinear Programming.* Reading, Massachusetts: Addison-Wesley, 2nd ed., 1984.

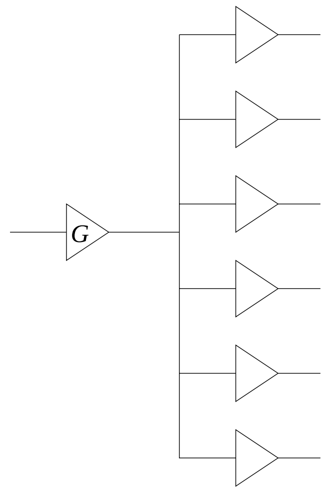

Figure 1: Effects of short-circuit power dissipation.

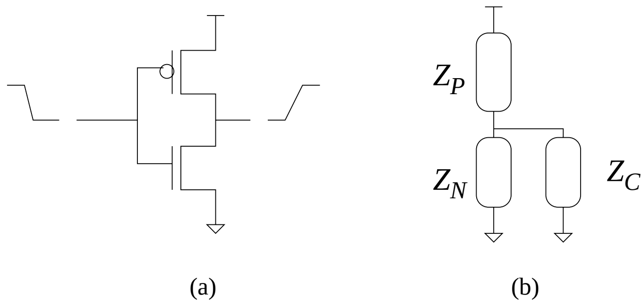

Figure 2: Examining the effect of w_n, w_p and C on short circuit power.

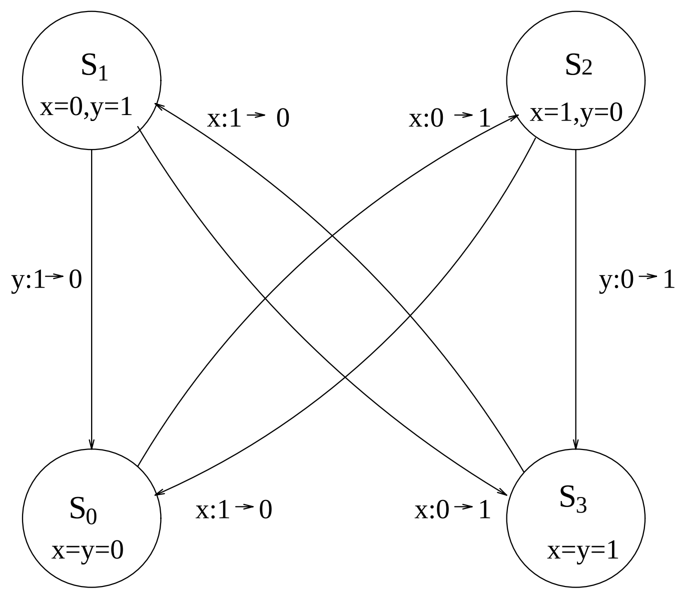

Figure 3: State diagram of a conceptual filter.

Table 1: Minimizing Power under Delay Constraints

Circuit	Timing Spec.(ns)	Area	$\frac{Power}{P_u}$	CPU Time (Iterations)
`cm138a` (90 tran)	$>$ 97.0ns	167.7	0.9929	17s(1)
	90ns	173.7	1.0007	39s(2)
$D_u = 118.0$ns	80ns	191.2	1.0093	89s(1)
$A_u = 162.0$	75ns	202.8	1.0200	314s(4)
`i1` (168 tran)	$>$ 147ns	303.3	0.9997	6.7m(1)
	120ns	309.2	1.0026	4.3m(1)
$D_u = 150.5$ns	100ns	313.8	1.0046	6.4m(2)
$A_u = 302.4$	80ns	335.8	1.0125	15.2m(1)
`cordic` (386 tran)	$>$ 199.0ns	699.6	0.9986	32m(1)
	150ns	732.4	1.0055	121m(4)
$D_u = 199.0$ns	140ns	752.2	1.0070	106m(2)
$A_u = 694.8$	130ns	914.4	1.0237	324m(2)
`comp` (588 tran)	$>$ 326.7ns	1078.3	0.9974	182m(1)
$D_u = 331.3$ns	300ns	1086.3	1.0044	518m(2)
$A_u = 1058.4$	280ns	1138.9	1.0148	1001m(2)
`c8` (698 tran)	$>$ 238.2ns	1260.5	0.9964	126m(1)
$D_u = 238.2$ns	230ns	1269.3	0.9978	445m(1)
$A_u = 1256.4$	220ns	1277.9	0.9992	779m(1)
`cht` (870 tran)	$>$ 288.7ns	1585.9	0.9570	413m(1)
$D_u = 286.2$ns	260ns	1585.2	0.9612	2412m(1)
$A_u = 1566.0$	240ns	1662.7	0.9830	985m(1)