Power vs. Delay in Gate Sizing: Conflicting Objectives?

Sachin S. Sapatnekar
Department of ECE
Iowa State University
Ames, IA 50011.

Weitong Chuang*
Macronix Semiconductor Co.
Science-based Industrial Park
Hsinchu, Taiwan 300.

Abstract

The problem of sizing gates for power-delay tradeoffs is of great interest to designers. In this work, the theoretical basis for gate sizing under delay and power considerations is presented, and results on a practical implementation are presented. The dynamic power as well as the short-circuit power are modeled, using notions of delay and transition density, and the optimization problem is formulated using notions of convex programming. Previous approaches have not modeled the short circuit power, and our experimental results show that the incorporation of this leads to counter-intuitive results where the minimum-power circuit is not necessarily the minimum-sized circuit.

1 Introduction

It has long been realized that the procedure of gate sizing is 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 [1–3], to name just a few. Lately, the power dissipation has emerged as another vital consideration in circuit design. This paper approaches the problem of gate sizing for power-delay tradeoffs under a nonlinear programming formulation. The exact solution to the formulated optimization problem is found.

The contributions of this work include appropriate modeling of the power dissipation (including short-circuit dissipation) for accuracy and tractability. We show that power and delay are not necessarily conflicting objectives; one can increase transistor sizes from the minimum size in a circuit and reduce its power dissipation! This is consistent with observations made in [4].

The sizing problem can be described as follows. 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, improvements in the timing behavior of a circuit can be achieved by increasing the sizes of some transistors in the circuit. This incurs expenses in terms of additional chip area, and often (but not always, as will be shown later), increased power dissipation. An optimization problem must be solved for a set of transistor sizes with an acceptable tradeoff.

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 formulation that recognizes that a designer's objective is to achieve the best performance at a given clock period may be stated as

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

where both $Delay$ and $Power$ are functions of the gate sizes, $\mathbf{W}$, T_{spec} and A_{spec} are, respectively, the constraints on the circuit delay and area, and Minsize is dictated by the technology.

*formerly at AT&T Bell Laboratories, Murray Hill, NJ 07974

It should be pointed out here that the optimization problem (1) must be solved on one combinational subcircuit at a time. Although the entire circuit may have millions of 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 [2,5,6]. All of these approaches considered 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. This could lead to incorrect results: for example, neglecting the short-circuit power would imply that a minimum power circuit must have minimum-sized transistors, a statement that we will show to be untrue.

In this work, we utilize Elmore time constants [7] and estimate the circuit delay and power dissipation based on this timing model. We show that both the circuit delay and the power dissipation are posynomial functions[1] [8] of the gate sizes, and show that the sizing problem, with any one of the area, delay and power as the objective and the remaining two as constraint functions, can be solved by solving a small number of convex programs. The relevance of this fact is in that convex programs are unimodal, and any local minimum is a global minimum. These underlying convex programs may be solved exactly using an efficient and rigorous mathematical optimization algorithm as is done here, or by good heuristics like TILOS [1].

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, the term "gate sizes" will henceforth refer to the W_n and W_p values for all gates in the circuit.

The paper is organized as follows. Section 2 outlines the gate level model. The power and delay models are described in Sections 3 and 4, respectively. The formalization of the optimization problem is featured in Section 5. Experimental results are presented in Section 6, and we conclude in Section 7.

2 Gate-level Modeling of a Circuit

We reduce each gate to an equivalent inverter whose n- (p-) transistor size is W_n (W_p). The optimal values of the W_n's and W_p's are found by solving an optimization problem, and these may be mapped back individual transistor sizes.

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

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

[1] A posynomial is a function g of a positive vector $\mathbf{w} \in \mathbf{R}^n$ that has the form $g(\mathbf{W}) = \sum_j \gamma_j \prod_{i=1}^{n} w_i^{\alpha_{ij}}$ where the exponents α_{ij} are real numbers and the coefficients $\gamma_j > 0$. A posynomial can be mapped onto a convex function through the variable transformation $(w_i) = (e^{x_i})$ [8]

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. Each is directly proportional to transistor size, x; due to the symmetry of the transistor, the proportionality constants for the source and drain parasitic capacitances are equal, and different from that for the gate capacitance.

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 leakage current power is negligible and is not considered. Each of these is described below, and it is shown that when the switching count at a gate output is fixed, both components are posynomial functions of the gate sizes, which implies that the total power is also a posynomial.

3.1 Dynamic Power Dissipation

The dynamic power is 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, under a clock frequency f, and with a switching probability of p_T, is given by

$$P_{dynamic} = C_l \cdot V_{dd}^2 \cdot f \cdot p_T \tag{3}$$

where V_{dd} is the supply voltage. From the gate model shown above, it can be seen that the output capacitance driven by a gate, and hence the dynamic power, is a linear (and hence posynomial) function of the W_n and W_p values of the current gate and of all of its fanouts for a fixed set of p_T's.

3.2 Short-circuit Power Dissipation

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 direct path between V_{dd} and ground. The associated power, the short circuit power, is given by [9]:

$$P_{sc} = \frac{\beta}{12}\left(V_{dd} - 2V_T\right)^3 \cdot \tau \cdot f \cdot p_T \tag{4}$$

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.

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, as in [3]. Therefore, we see that the short circuit power dissipation for gate G is dependent on
(a) the size of its own equivalent inverter, which contributes to the factor β in Equation (4), and
(b) that of the equivalent inverters for G_1 and all of its fanout gates, which contribute to the factor τ in Equation (4).

Note that τ is a posynomial function of the gate sizes (as will be shown in Section 4), and when multiplied by β (which is $\propto x$), remains a posynomial function of the gate sizes (albeit a different one from τ) for a fixed set of p_T's.

3.3 Computing Transition Densities

This work utilizes a probabilistic measure of switching activity called *transition density* [10], where the algorithm takes the signal probability and transition density of each primary input and propagates the values through logic modules to estimate the transition density at each node in the circuit. The algorithm was enhanced in [11], where the effect of filtration in real circuits was pointed out: due to the inertial delays of logic gates, short pulses are filtered out as the module is not fast enough to respond to them. To model this filtration effect of the circuit inertial delays, a new delay block called a filter block was introduced. Note that the transition density at the output of a gate using the method of [11] is the function of the gate delays. We now outline the procedure for calculating transition densities; for details the reader is referred to [10,11].

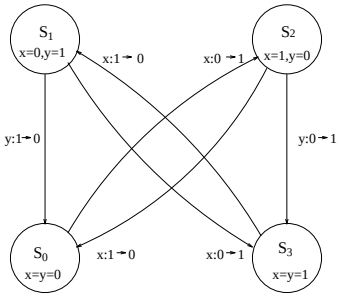

Figure 1: State diagram of a conceptual filter.

Let $P(x)$ denote the *equilibrium probability* of a logic signal $x(t)$, i.e., $P(x) \equiv P(x(t) = 1)$. This gives the fraction of time that the signal $x(t)$ is high. Let $n_x(T)$ denote the number of transitions of $x(t)$ in $(-T/2, T/2]$. Then the transition density of $x(t)$, $D(x)$, is defined to be $D(x) = \lim_{T\to\infty} \frac{n_x(T)}{T}$.

It has been shown in [10] that, if $y = f(x_1, x_2, \cdots, x_n)$ is a Boolean function and the inputs x_i's are independent, then the density of output y is given by:

$$D(y) = \sum_{i=1}^{n} P(\frac{\partial y}{\partial x_i}) D(x_i) \tag{5}$$

where $\frac{\partial y}{\partial x}$ is the Boolean difference of y with respect to x. For simple gates (AND, OR, etc.), the Boolean difference can be easily calculated, and for more complex Boolean functions, the OBDD package can be used. Given the probability and density values at the primary inputs, a single pass over the circuit, using (5), 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 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 3.3. The state S_0 (corresponding to $x = y = 0$) and S_3 ($x = y = 1$) are *stable states*. The other two states, S_1 ($x = 0, y = 1$) and S_2 ($x = 1, y = 0$), 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) at any time during this period if x switches back to 1 (0) again. Under this model, $P(y)$ and $D(y)$ can be computed from $P(x)$ and $D(x)$ using the procedure derived in [11].

4 Computation of the Circuit Delay

The delay calculation procedure is similar to that in [1]. The maximum rise and fall delays between the primary inputs and primary outputs of the circuit are computed using PERT. Two

numbers, t_h and t_l, are assigned to each gate output, and are 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 the corresponding RC networks. These RC networks are easy to build; for the falling (rising) transition, we have the resistance of the n-(p-)transistor driving the gate output capacitances, comprising of drain capacitances of the n- and the p-transistor in the equivalent inverter for the current gate, and the gate terminal capacitances of all fanout gates.

It can be seen from the RC models described in Section 2 that the delay of any gate is a posynomial function of the gate sizes [1]. This has two consequences: firstly, it implies that the transition time, τ, for each gate is a posynomial. This fact has been used in Section 3.2 to show that the short-circuit power is a posynomial function of the gate sizes. Secondly, since the delay of any path is a sum of gate delays, it is also a posynomial, and therefore, the circuit delay, which is the maximum of all path delays, is a maximum of posynomials.

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, an invalid assumption that will later be removed.

The sizing problem is stated in (1). It has been shown earlier that for constant switching probabilities, the power is a posynomial function of the gate sizes, and the circuit delay is a maximum of posynomials. Therefore, the transformation $(w_i) = e^{x_i}$ maps the power to a convex function, and the delay into a maximum of convex functions, which is also convex [12].

Under this mapping, the optimization problem is stated as:

$$\begin{aligned} &minimize && Power(\mathbf{x}) \\ &subject\ \ to && Delay(\mathbf{x}) \leq T_{spec} \\ & && Area(\mathbf{x}) \leq A_{spec} \\ &and && \text{Each gate size} \geq \text{Minsize} \end{aligned} \tag{6}$$

where $Delay(\mathbf{x})$ and $Power(\mathbf{x})$ are convex functions in $\mathbf{x}$. Therefore, all of the constraints above are convex, as is the objective, and we now deal with a convex programming problem. An efficient algorithm for the exact solution of convex optimization problems [3] is used to obtain the solution to (6).

Note that if we present the sizing problem as an power-delay-area tradeoff, with one of these three functions being the objective, and with constraints on the other two, then the corresponding optimization problem can also be mapped onto a convex programming problem. This arises out of the fact that the area can be modeled as a posynomial [1].

Therefore, we may see that solving the power-delay-area tradeoff for a *fixed* set of gate switching probabilities is equivalent to solving a convex programming problem. However, as pointed out in Section 3.3, the switching probabilities are dependent on the gate delays, which are, in turn, dependent on the gate sizes, and therefore the above assumption is invalid. Therefore, we use the following solution scheme:

```
error = ∞;
Set all gates to Minsize;
Calculate gate delays;
Compute p_T = vector of transition probabilities
   at each gate based on current gate delays;
while (error < ε) {
   oldp_T = p_T;
   Solve gate sizing problem (convex program) for p_T;
   Calculate gate delays for new gate sizes;
   Compute p_T = vector of transition probabilities
      at each gate based on current gate delays;
   error = sqrt(Σ [p_Ti − oldp_Ti]^2) / (#gates);
}
```

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 convex programming problem under this assumption. On solving this problem, we get a new set of gate sizes, and therefore a new set of gate delays. We then recompute $\mathbf{p}_T$, and continue the iterations until the switching probabilities converge. We cannot formally prove that convergence is guaranteed, but in practice, we found that convergence occurs in a very small number of iterations (no more than two for all examples that we tried).

Note that due to the dependence of transition probabilities on gate delays, we cannot claim the *true* problem of gate sizing for power-delay-area tradeoffs to be a convex programming problem. However, it is believed that the procedure above gives a good solution in a reasonable amount of time.

Table 1: Minimizing Power under Delay Constraints

Circuit	Timing Spec.(ns)	Area	$\frac{Power}{Cycle}$	CPUTime (Iterations)
cm138a (90 tran)	80ns	202.7	68.9e-12	31.8s(2)
P_u = 65.0e-12	70ns	240.4	89.0e-12	63.5s(2)
D_u = 100.6ns	65ns	365.7	127.8e-12	53.8s(2)
A_u = 162.0	60.3ns †	966.2	295.e-12	128.8s(2)
cordic (386 tran)	100ns	758.8	336.2e-12	953.9s(2)
P_u = 318.5e-12	80ns	1015.5	396.1e-12	388.3s(2)
D_u = 134.1ns	70ns	1693.3	524.1e-12	859.3s(2)
A_u = 694.8	64.4ns †	10826.4	2098.8e-12	256.8s(2)
b9 (472 tran)	minpower	877.8	358.4e-12	684.8s(1)
P_u = 429.9e-12	150ns	881.3	358.4e-12	713.2s(1)
D_u = 188.3ns	100ns	1457.2	563.8e-12	1359.0s(2)
A_u = 849.6	95ns	1860.0	684.1e-12	2282.3s(2)
	90ns	5022.2	1647.2e-12	696.1s(2)
	87.1ns †	9937.1	3669.1e-12	497.6s(2)
comp (588 tran)	minpower	1094.1	350.8e-12	2517.5s(1)
P_u =435.2e-12	120ns	1610.1	513.2e-12	7326.7s(2)
D_u =257.5ns	100ns	2382.1	631.1e-12	3025.5s(2)
A_u =1058.4	90ns	3445.6	793.7e-12	3008.4s(2)
	82.2ns†	16371.3	2427.0e-12	12765.5s(2)
ttt2 (952 tran)	minpower	1791.7	586.3e-12	3171.5s(1)
P_u = 754.0e-12	200ns	1854.7	824.7e-12	3366.8s(2)
D_u = 267.2ns	150ns	2159.5	919.7e-12	1949.5s(2)
A_u = 1713.6	130ns	4078.5	1457.1e-12	1752.4s(2)
	122ns †	16556.9	6176.4e-12	3680.6s(2)

† denotes minimum achievable timing specification

6 Experimental Results

The theory developed above can be applied to extend any existing optimization algorithm for sizing, such as CONTRAST [3] or TILOS [1]. In this work, we have employed Vaidya's convex programming algorithm [13], also used in the CONTRAST algorithm, to find an exact solution to the convex programming problem formulated above. However, the TILOS approach may also be adapted to solve this problem.

The algorithm described above has been implemented as a C program on a DEC Alpha 3000/AXP300. Results on several benchmarks are presented. Table 1 illustrates the power-delay tradeoff for various circuits under various delay specifications. The first column lists the circuit name; the circuits have been chosen from the LgSynth91 benchmarks. The power, P_u, 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 and circuit area found by the algorithm are shown in the next three columns. Note that the power is specified in terms of the power per clock transition, which is denoted by $\frac{Power}{Cycle}$ in the table. In each case, it was found that the results were obtained in under two iterations, i.e. no more than two convex programs were required to be solved. The value of ϵ used to control convergence was set to 0.01. The bulk of the computation is consumed by the transistor sizing 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. Moreover, with a tightening of the delay specification, the marginal increase in the power is found to increase.

Traditional estimates of power for sizing purposes have considered only the dynamic component of power, given by Equation (3). 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.

Table 2: Minimizing Area vs. Minimizing Power

Timing Specification	Min-area Optimization Area/Power @ optimum	Min-power Optimization Area/Power @ optimum
> 257.5 ns	1058.4/435.2e-12	1094.1/350.6e-12
220.1 ns	1067.2/350.6e-12	1094.1/350.6e-12
150.0 ns	1234.4/475.4e-12	1351.3/351.6e-12
120.0 ns	1530.8/546.7e-12	1610.1/513.1e-12
100.0 ns	2231.3/677.7e-12	2382.1/631.1e-12
90.0 ns	3071.7/857.4e-12	3445.6/793.7e-12
85.0ns	4384.1/1159.4e-12	5344.9/1094.3e-12

Lastly, for circuit comp, we present in Table 2 the circuit area and the corresponding power (per cycle), for the case where the objective is to minimize the circuit *area*, and the case where the objective is to minimize the *power*, under various delay specifications. Since the minimum power circuit corresponds to a delay of 220.1 ns, for any larger specification the power minimization will result in a circuit with that delay. This table shows that minimizing power and area are related, but not identical objectives. It also answers the question posed in the title of this paper: delay and power reduction do not always conflict.

The delay corresponding to the minimum-power circuit here is 15% less than that for a minimum-sized circuit. This is not surprising since the goals of reducing the delay and the short-circuit power are consistent. This effect is likely to be more pronounced when large loads are being driven by a gate; in such cases, the improvements in delay and power using nonminimum sizes are likely to be larger. The minimum-power circuit is found to dissipate 19.5% less power than the minimum area circuit.

We caution the reader that the minimum delay 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 delay and minimum power points were distinct. This is consistent with the fact that larger circuits typically 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 as a small number of convex optimization problems. It has been shown here that when the short-circuit power dissipation for a minimum-sized circuit is significant, the minimum power circuit is *not* the minimum sized circuit. It has been shown that power and delay are not necessarily conflicting objectives; it is possible that as a tighter delay specification is applied, a concomitant reduction in power (compared to a minimum-sized circuit) may be achieved up to a point, beyond which further delay reduction implies an increase in the power.

The short-circuit power model in this work may be refined further, and future work relates to the use of more accurate short-circuit power models during optimization.

References

[1] J. Fishburn and A. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," in *Proc. ICCAD*, pp. 326–328, 1985.

[2] M. R. Berkelaar and J. A. Jess, "Gate sizing in MOS digital circuits with linear programming," *Proc. European DAC*, pp. 217–221, 1990.

[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 Trans. CAD*, pp. 1621–1634, Nov. 1993.

[4] M. Borah, M. J. Irwin, and R. M. Owens, "Experiments with low-power and high-performance CMOS layouts," Tech. Rep. CSE-94-027, Department of Computer Science and Engineering, Pennsylvania State University, 1994.

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

[6] Y. Tamiya, Y. Matsunaga, and M. Fujita, "LP based cell selection with constraints of timing, area and power consumption," *Proc. ICCAD*, pp. 378–381, 1994.

[7] J. Rubenstein, P. Penfield, and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. CAD*, pp. 202–211, July 1983.

[8] J. Ecker, "Geometric programming: methods, computations and applications," *SIAM Rev.*, vol. 22, pp. 338–362, July 1980.

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

[10] F. N. Najm, "Transition density, a stochastic measure of activity in digital circuits," *Proc. DAC*, pp. 644–649, 1991.

[11] F. N. Najm, "Low-pass filter for computing the transition density in digital circuits," *IEEE Trans. CAD*, pp. 1123–1131, Sept. 1994.

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

[13] P. M. Vaidya, "A new algorithm for minimizing convex functions over convex sets," *Proc. IEEE Foundations of Computer Science*, pp. 332–337, Oct. 1989.