Combined Transistor Sizing with Buffer Insertion for Timing Optimization

Yanbin Jiang † Sachin S. Sapatnekar ‡ Cyrus Bamji * Juho Kim *

† Department of ECE, Iowa State University, Ames, IA 50011
‡ Department of ECE, University of Minnesota, Minneapolis, MN 55455
* Cadence Design Systems, San Jose, CA 95052.

Abstract

This paper presents strategies to insert buffers in a circuit, combined with gate sizing, to achieve better power-delay and area-delay tradeoffs. The delay model incorporates placement-based information and the effect of input slew rates on gate delays. The results obtained by using the new method are significantly better than the results given by merely using a TILOS-like transistor sizing algorithm alone.

1 Introduction

The transistor sizing problem [1, 2, 3] is often formulated as $Minimize\ \ Area$ subject to $Delay \leq T_{spec}$.

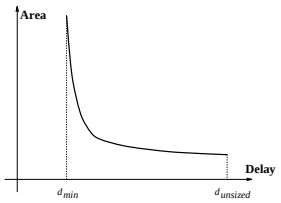

Figure 1: Area-delay curve for sizing

For a given combinational circuit, the nature of the area-delay tradeoff curve for transistor sizing is as shown in Fig. 1. Typically, a small amount of sizing is adequate to reduce the delay corresponding to the unsized circuit, $d_{unsized}$. However, as the specification is tightened, the circuit has to be sized tremendously to achieve further delay reduction. Further, it is impossible to reduce the delay of a circuit indefinitely through sizing, and there is a minimum achievable delay, d_{min}, that cannot be bettered through sizing.

Traditionally, transistor sizing and buffer insertion (the "fanout problem") [4, 5] have been carried out separately and at different stages of the design process However, as sizing changes the capacitances driven by various gates, the locations of high-capacitance nodes are accurately established only during sizing, and any optimizations performed before sizing are necessarily based only on educated guesses. Therefore, it is useful to combine the two optimizations into a single step, and this is the objective of this research.

The organization of this paper is: Section 2 introduces the delay and area modeling; Section 3 talks about the criticality calculation; The program outline is presented in Sections 4 and 5; Complexity is given in Section 6; Experimental results are presented in Section 7, followed by concluding remarks in Section 8.

2 Delay and area modeling

As in previous work (for example, [1, 2]), the circuit area is modeled as the sum of all transistor sizes. We use the width of each transistor to represent the size of the transistor.

At the gate level, each static CMOS gate G_i is modeled by an equivalent inverter. The relation between the gate sizes in the equivalent inverter and transistor widths in the gate can easily be computed for various type of gates. For example, for a k-input NAND gate, $S_{n,i} = w_{n,i}/k$, $S_{p,i} = w_{p,i}$. $S_{n,i}$ ($S_{p,i}$) is the n-transistor (p-transistor) size of the equivalent inverter.

The capacitance loading, C_L, of gate G_i is:

$$C_L = \sum_{j \in fanout_i} C_{gate_j} + C_{intrinsic} + C_{wire} \qquad (1)$$

where $C_{intrinsic}$ corresponds to the source and drain capacitance connected to the output node of G_i. The wire capacitance values are based on the placement.

The Elmore fall step delay, t_{f_i}, of gate G_i can then be obtained from C_L and $S_{n,i}$ as [6, 7]

$$t_{f_i,\text{step}} = R_{i_n} \cdot C_L \qquad (2)$$

where $R_{i_n} = \frac{R_n}{S_{n,i}}$. R_n is a constant. The rise delay is similarly obtained as $t_{r_i,\text{step}} = R_{i_p} \cdot C_L$.

We use the inverter delay model presented in [8]. The effect of the input-to-output coupling capacitance and input slope effects are considered in this model. When the applied input is the ramp

$$V_{in} = \begin{cases} 0 & t \leq 0 \\ \frac{V_{DD}}{\tau} t & 0 \leq t \leq \tau \\ V_{DD} & t \geq \tau \end{cases}, \qquad (3)$$

where τ is the slope of the input ramp, the delay is given by

$$t_{f_i,\text{ramp}} = v_{TN} \cdot \frac{\tau}{2} + \left(1 + 2\frac{C_M}{C_L}\right) t_{f_i,\text{step}} \qquad (4)$$

Here, v_{TN} is $\frac{V_{TN}}{V_{DD}}$ where V_{TN} is the threshold voltage of the n-transistor and V_{DD} is the supply voltage. C_M is the coupling capacitance between the input and the output nodes. C_L is the driving load. Typical values of v_{TN} and $\frac{C_M}{C_L}$, which we use in this work, are 0.2 and 0.1, respectively [8]. A similar expression is used for the rise transition. The value of τ is taken to be twice the Elmore delay of the preceding gate, as in [2].

3 Criticality Calculation

Roughly speaking, the criticality of a path is dependent on the magnitude of the violation of the timing specification, so that paths with large violations are identified as being highly critical, and those with small violations are only mildly critical.

Consider the sensitivity, $\frac{\partial d}{\partial x_i}$ for each gate i, where x_i is the size of the gate, and d is the delay of the most critical path through the gate. We maintain the number

$$\sigma_i = \min(0, \frac{\partial d}{\partial x_i} \cdot \Delta x_i) \qquad (5)$$

for each gate i, where Δx_i is the amount by which the gate size would be increased if it were to be bumped up. Therefore, σ_i estimates the reduction in the gate delay through a possible bumping up operation. Note that gates with a positive sensitivity are assigned a σ_i of zero since the gate size would be left unchanged if the bumping operation were to increase the delay.

We define a measure for the criticality that we call χ, associated with each gate fanout. Fanouts with larger χ values are less critical than those with smaller χ values.

A backward PERT traversal is performed from the primary outputs towards the primary inputs (PI's) to calculate the value of χ for each gate. The χ value at each primary output is set to be the difference between the maximum delay at the primary output and the actual delay to that point. Therefore, increasing the path delay to that primary output by χ will leave the circuit delay unchanged.

If we know the χ value for all the fanouts of a given gate i, its own χ value is calculated as

$$\chi_i = \min_{j \in \text{fanouts}(i)} [\chi_j + \text{slack}_j] + |\sigma_i| \qquad (6)$$

where slack_j represents the slack at fanout j. The slack is defined as the amount by which the delay along this path may be increased before it becomes the longest delay path in the circuit.

4 Buffer insertion

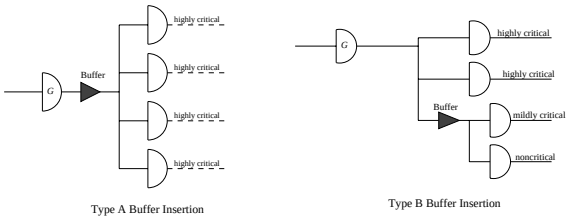

Figure 2: Types of buffer insertion

Type A If a gate whose outputs are all *highly critical* drives a large capacitive fanout, buffer insertion can help in reducing the delays of these paths. By choosing an appropriate size of buffer, the fanout capacitance of Gate G may become smaller, and sum of the delays of the buffer and Gate G may be smaller than the delay of Gate G in the unbuffered circuit.

Type B If a gate has some highly critical outputs and some mildly critical and noncritical outputs, then one may isolate the capacitance of the noncritical outputs from the highly critical path by inserting a buffer. Since the fanout capacitance of gate G becomes smaller, the RC delay of G is reduced, and therefore, the delay along the highly critical paths is reduced.

5 Outline of the algorithm

The transistor sizing problem is well known to be equivalent to a convex programming problem [1, 2] when the topology of the circuit is fixed. However, when the structure of the circuit is allowed to change (by inserting buffers), this is no longer true. Finding the optimal locations for Type B buffers in an unsized circuit, is NP-complete [5]. Therefore, we resort to heuristic methods for solving the problem.

We attempt to improve the delay along the critical path by one of several possible transformations in each iteration. (1) bumping up the size of some transistor along the path. (2) inserting a Type A buffer along the critical path. (3) inserting a Type B buffer to isolate noncritical paths from critical paths.
The general philosophy behind the algorithm is shown below.

```
minimum_delay = minimum-sized circuit delay
Initialization (all gate sizes are set to minimum values)
While (delays at all primary outputs are not ≤ T_spec) {
 Compare path delays with T_spec and find the most critical path
 For all gates on the critical path {
     Estimate figure of merit of bumping up a transistor
     Estimate figure of merit for inserting a Type A buffer
     Estimate figure of merit for inserting a Type B buffer
 }
 If (bumping up a transistor has the best figure of merit)
```

```
    increase the size of a selected transistor
if (inserting a Type A buffer has the best figure of merit)
    insert a Type A buffer
if (inserting a Type B buffer has the best figure of merit)
    insert a Type B buffer
Recompute circuit delays
if (circuit_delay < minimum_delay) minimum_delay = circuit_delay
if (circuit_delay > 1.1 × minimum_delay)
    /* failed to meet specifications */
    exit
}
```

In each step, one of the three transformations is performed for delay reduction. After a certain point, since the circuit delay cannot be reduced indefinitely, the circuit delay will be seen to increase in successive iterations. We terminate the iterations when the delay increase is seen to be significant, i.e., 1.1 x minimum_delay.

5.1 Type B buffer insertion

A type B buffer will always reduce the delay to a highly critical fanout at the expense of an increase area of the inserted minimum-sized buffer. Therefore, a reduction in the delay by an amount ΔD can be effected by an area increase of ΔA. We must now estimate the amount of area, ΔA_T, required by the sizing procedure to achieve the same delay reduction. If $\Delta A < \Delta A_T$, then we insert the Type B buffer.

To estimate the value of ΔA_T, given a specific buffer insertion point, at each such primary output i, we use an extrapolation method to estimate the area increase, Δa_i, required to match the circuit delay reduction. We then calculate the figure of merit for sizing as

$$\Delta A_T = \sum_{i \in po} \Delta a_i \tag{7}$$

We use Lagrangian extrapolation to estimate ΔA_T for ΔD. We found that a fourth order polynomial approximation was adequate.

The steps involved in determining the buffer location can now be summarized as follows:

1. Find the gate i with the maximum fanout capacitance along the most critical path of the circuit.

2. Find the maximum value of χ_j of all fanouts of gate i; let χ_{max} be the maximum value of χ_j. All fanouts j whose χ_j is $\geq c_1 \cdot \chi_{max}$ (where $c_1 < 1$ is an empirically tuned number) are placed in the noncritical set.

3. When a buffer is inserted, the delay of gate i is reduced by an amount ΔD_{dec}, which is the delay reduction along the critical paths. Along a noncritical fanout j, the delay is increased by $\Delta D_{inc} - \Delta D_{dec}$, where ΔD_{inc} is the increased delay due to the insertion of a buffer. Therefore, with the insertion of the buffer, we may say that the delay from j to the primary outputs may be increased by $\chi_j - (\Delta D_{inc} - \Delta D_{dec})$. The larger this amount, the less critical the path would be after buffer insertion. Therefore, we calculate this quantity for each fanout and if its value is small, then we remove the fanout j from the noncritical set.

4. For any fanout j, if $\chi_j - (\Delta D_{inc} - \Delta D_{dec}) < \beta$, the gate is moved from the noncritical set to the critical set.

We perform a type B buffer insertion to isolate the critical set (gates) from the noncritical set (gates).

5.2 Type A buffer insertion

The following procedure is used to estimate the potential delay reduction through Type A buffer insertion at each gate output:

1. Find the minimum (most negative) sensitivity among the gates along the most critical path, denoted as $\frac{\partial D}{\partial x}\big|_{best}$.

2. For each gate on the most critical path, we calculate the values of ΔD_{rise} and ΔD_{fall}, the changes in the rise and fall delays, respectively. Only those gates at which both the rise and fall delays can be reduced are considered as candidates for buffer insertion. For these gates, the sensitivity of the buffer, $\frac{\partial D}{\partial x}\big|_{buffer}$, is determined for the calculated size.

   If $\frac{\partial D}{\partial x}\big|_{buffer} < \frac{\partial D}{\partial x}\big|_{best}$, then this location is designated as a permitted buffer insertion location.

3. Among the permitted buffer insertion points in Step 2, the output of gate k with the best delay reduction is chosen to be the best Type A buffer insertion location.

4. Having performed a Type A buffer insertion, the buffer and its predecessor gate k are now reset to the minimum size to correct for any over-sizing in k in the past. The sizing procedure is permitted to size these gates back up again in subsequent iterations to their optimal sizes, so that the solution is not unduly bound by any incorrect sizing choices that were made before the buffer was added.

6 Complexity

Each iteration requires $O(|V|+|E|)$ time for timing analysis and slack calculation, $O(D_c)$ time for sensitivity calculation, $O(|V| + |E|)$ time to evaluate type

Table 1: Comparison of Sizing vs Sizing+Buffer Insertion

Circuit	$\|G\|$	D_u	A_u	T_{spec}	*Sizing*		*Sizing+Buffer Insertion*		Area Ratio
					Area	CPU time(s)	Area	CPU time(s)	
cc	58	61.4	248	23	900	6.7	706 (A:1; B:6)	4.9	1.27
cm163	43	43.4	160	14	692	3.0	428 (A:1; B:5)	2.6	1.62
f51m	136	82.5	548	50	2627	17.6	1627 (A:1; B:4)	13.3	1.62
i135	269	121.1	1252	36	4307	29.9	2183 (A:0; B:13)	32.8	1.97
c499	202	177.3	816	51	3004	30.2	2571 (A:1; B:15)	62.4	1.17
c1355	546	324.5	2128	100	5001	145.9	4279 (A:1; B:38)	192.3	1.17
c2670	1193	456.0	4152	88	9000	481.3	8586 (A:1; B:94)	595.7	1.05
c5315	2307	831.2	8772	190	15000	987.2	13619 (A:1; B:125)	1013.3	1.10

B buffer insertion, and $O(D_c)$ time to evaluate Type A buffer insertion, and since $D_c < |V|$, the overall complexity of each step is $O(|V| + |E|)$. $|V|$ is the number of vertices in the circuit graph, corresponding to the number of gates in the circuit, and $|E|$ is the number of edges in the circuit graph, where each edge corresponds to an interconnection from one gate to one of its fanouts. D_c is the depth of the circuit (largest number of gates on any path). We emphasize that due to the incremental techniques used, this is a pessimistic estimate of the complexity.

7 Experimental results

The algorithm have been implemented in C on an HP 735 workstation. In Table 1, we present the results on some circuits from the ISCAS85 and LgSynth91 benchmark suites.

For each circuit, the number of gates $|G|$, the unsized delay D_u, and the unsized area A_u are shown. For a given (moderate) timing specification T_{spec}, the area of our approach is compared with the area from our implementation of TILOS, which is a direct implementation from [1]. Next to the area numbers the table are also shown (in brackets) the number of Type A and Type B buffers. The CPU times for both methods are very similar. The area ratio shown in the last column shows the ratio of the area required by sizing alone as compared to the area required by our method. Our algorithm achieved from 5% to 49% area reduction.

The entire area-delay tradeoff for this algorithm for i135 circuit is shown in Fig. 3.

8 Conclusion

In this paper, we have aimed to support the basic idea that buffer insertion can help to improve the area-delay tradeoff curve and have presented heuristic algorithms for the purpose. In this work, the efficacy of reducing the dynamic power dissipation is further improved by considering buffer insertion to achieve the delay goal for the circuit with a smaller area/power cost.

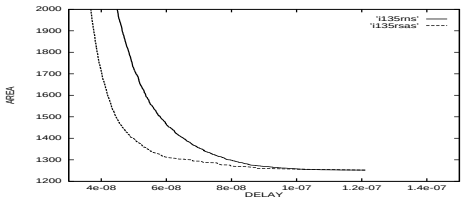

Figure 3: Area-delay tradeoff for circuit i135

References

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

[2] 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. on CAD*, vol. 12, pp. 1621–1634, Nov. 1993.

[3] M. Borah, R. M. Owens, and M. J. Irwin, "Transistor sizing for low power CMOS circuits," *IEEE Trans. on CAD*, vol. 15, pp. 665–671, June 1996.

[4] K. J. Singh and A. Sangiovanni-Vincentelli, "A heuristic algorithm for the fanout problem," in *Proc. of DAC*, pp. 357–360, 1988.

[5] C. L. Berman, J. L. Carter, and K. L. Day, "The fanout problem: From theory to practice," in *Advanced Research in VLSI: Proc. of the 1989 Decennial Caltech Conf.*, pp. 69–99, 1989.

[6] W. C. Elmore, "The transient response of damped linear networks with particular regard to wideband amplifiers," *Jour. of Applied Physics*, vol. 19, Jan. 1948.

[7] J. Rubinstein, P. Penfield, and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. on CAD*, vol. CAD-2, pp. 202–211, July 1983.

[8] K. O. Jeppson, "Modeling the influence of the transistor gain ratio and the input-to-output coupling capacitance on the CMOS inverter delay," *IEEE JSSC*, vol. 29, pp. 646–654, June 1994.