Path-based and Crosstalk-aware Gate Sizing with Simulated Annealing

Zhiyang Ong, Henry Mecke, Hanif Fatemi, Shahin Nazarian, and Massoud Pedram
Department of Electrical Engineering,
University of Southern California,
Los Angeles, CA 90089, USA

***Abstract*— A simulated annnealing algorithm for gate sizing to minimize area and delay while considering crosstalk is proposed. It randomly selects a gate at each temperature for sizing to minimize delay, crosstalk noise, and area. Simulation results for area and critical path delay minimization indicate that the circuit delay with due considerations given to crosstalk is improved by more than four-fold. Also, when the circuit is optimized for area and total gate delay, the critical path delay is found to improve by 75.19% at a marginal trade-off for area (increase of 9.52%). Finally, conclusions to this proposed approach are drawn from experimental results, and a list of open research directions is provided.**

I. Introduction

A. Background and Motivation

As the number of transistors on integrated circuits and their clock frequencies increase exponentially according to Moore's Law, the interconnects are increasingly becoming smaller, and are placed closer together. Consequently, their resistance and coupling capacitance increases, and the gate delay decreases linearly as interconnect delay increases exponentially; these delays can be estimated with the product of resistance and capacitance in the RC delay model, and from models for transistor scaling effects [1], [2]. Also, to further complicate things, there exist fluctuations in voltage levels for power and ground networks due simultaneous switching circuits [3]. This results in the following key design challenges facing the semiconductor industry, such as the reduction of electromagnetic interference, protection of signal integrity, improvement of heat dissipation, and minimization of static and leakage power dissipation [4]–[9].

There are two types of crosstalk, capacitive and inductive crosstalk, that result in the addition of noise to the interconnects, and an electromagnetic interference on them that depend on signal changes. Capacitive crosstalk results from the capacitive coupling between neighboring interconnects when one, some, or all of them switch(es) voltage value(s). For example, the change in the voltage value of an aggressive interconnect, may affect the switching delay or noise level of its adjacent interconnect (also called the victim), depending on whether the adjacent interconnect is not switching, or if they are switching in the same or opposite directions. Inductive crosstalk arises from the inductive coupling of signals on an interconnect to another interconnect, since the switch in voltage levels/current leads to a change in magnetic field that induces proportional voltage changes in interconnects nearby. [1]. They may increase circuit delay and/or glitches on electrical paths that result in timing failure.

When analyzing crosstalk interference between adjacent interconnects, the attributes of wires as aggressor and victim are interchangeable, and they depend on the wire that is being is observed for the effects of electromagnetic interference, which lead to crosstalk effects such as clock jitter and clock skew [3]. This may affect the timing performance on the critical path of the circuit, and cause the circuit to fail to meet timing requirements [10].

Functional correlation analytical techniques to identify critical paths affected by crosstalk delay are used to determine the coupling between neighboring long wire paths. This enables crosstalk delay, rising from various switching directions and switching times, affecting these wires to be estimated. Subsequently, they can be used to determine the changes in critical path delay due to gate-sizing and buffer optimization in various physical design automation steps to reduce crosstalk delay effect for static timing optimization [10].

Gate sizing, which has a considerable impact on delay, power dissipation, and circuit area [11], entails modifying the sizes of gate in the circuit to optimize some objective functions subject to certain constraints and compliance to design rules [12]. Larger gates can drive bigger loads due to lower resistance. However, they consume more area, and increases the load on the previous stage [8]. There exist various techniques for gate sizing, such as gate sizing (GS) [12], statistical gate sizing [13], and geometric programming [11], to deal with these conflicting effects while optimizing delay subject to power and/or area constraints, and vice-versa for power and/or area optimization with delay constraints. The delay model is dependent on the load at the output and the fan-out of gates. Thus, the propagation times and output transition times of the fan-in gates, and their transitive fan-out gates need to be computed again when the gates are sized. Consequently, the sensitivities and slack time of the circuit paths that these gates lie on will be affected. If the slack time is negative, the delay constraints are not met, and signals traveling along this path in the circuit needs to be sped up with further gate sizing or buffer optimization. Hence, delay optimization should employ global optimization techniques, rather than local greedy methods [12].

Since the change in the delays of the proceeding gates on a

path has a significant drop-off when a gate is sized to improve its slack time, we only need to determine the change in slack time of neighboring paths up to two levels of fan-in and fan-out. In addition, further computation time can be saved by only recomputing the change in slack time of gates in the immediate vicinity of the resized gate [12]. Slack time is defined as the difference between the required time the signal must arrive to meet timing constraints and the arrival time that is the time taken for the signal to propagate to the point n in the circuit from the primary inputs; see equation (1) [12].

$$S(n) = RT(n) - AT(n) \tag{1}$$

B. Scope and contributions of our work

The aim of this paper is to propose a simulated annealing algorithm to minimize critical path delay, critical path delay with crosstalk, and circuit delay and area by using gate sizing after routing. The use of electrical effort as a delay metric, based on the linear delay model, may not adequately model the parasitic effects [8], bootstrapping, or different input arrival times [1] in ultra-deep sub-micron circuits. However, it is chosen as our delay model due to its ease of implementation. Similarly, a simple crosstalk noise model is chosen for the ease of implementation; it is based on the ratio of the aggressor's driver size to the sum of the gate sizes of the victim's driver and receiver. In addition, manufacturability and the minimization of clock skew, noise, and power are ignored to keep the problem as a single-objective or dual-objective optimization problem.

The paper is organized as follows. In section 2, our proposed simulated annealing algorithm to size gates for delay minimization, delay minimization with crosstalk noise accounted for, and minimization of circuit delay and area is described. Following that is Section 3, which discusses experimental results from implementing the simulated annealing algorithm for crosstalk-aware gate sizing. Section 4 outlines future work on combining gate sizing with other techniques for optimization with multiple objective such as delay, area, and power. Finally, we draw conclusions for this work in Section 5.

II. Proposed Approach

A. Crosstalk-aware Gate Sizing using Linear Programming

We endeavored to formulate the crosstalk-aware gate sizing problem as a linear program, so that we can enter the objective function and constraints of the linear program as an input file to a linear programming software that will determine an optimal feasible solution. However, we faced problems attempting to linearize the metrics for crosstalk and delay estimation, and were unable to proceed with solving this with linear programming.

B. Simulated Annealing Algorithm for Gate Sizing

As aforementioned, a global optimization technique for power and area minimization is needed. However, there does not exist any general purpose optimization technique that is better than all others, and the selection of appropriate optimization techniques and objective functions are problem specific [14]. This is attributed to the No Free Lunch (NFL) theorem, which states that all possible optimization techniques have the same performance on average upon evaluation of all possible objective functions [15]. Hence, we have decided to use simulated annealing to help us achieve our objectives.

The proposed algorithm is described below [16]–[18]:

Given:
S_0 is the initial solution;
T_0 is the initial temperature;
α is the cooling rate;
β is the constant to increase/decrease the amount of time its takes to cool at each temperature
M is the the amount of time its takes to cool at each temperature
$maxtime$ is the total time allowed for the annealing process to cool
t is the amount of unit time elapsed

Algorithm simulated_annealing $(S_0, T_0, \alpha, \beta, M, maxtime)$
$T = T_0$;
$S = S_0$;
$Time = 0$;
do{
  metropolis(S,T,M);
  $Time = Time + M$;
  $T = cooling\_schedule(T)$;
  $M = \beta \times M$;
}while(Time $<$ Maxtime)
return S = best output solution that is found;

Algorithm metropolis (S, T, M)
% S is the current solution
% T is the current temperature
% M is the amount of cooling time spend at this temperature
do{
  temp_solution = neighbor(S);
  $\Delta E = cost(temp\_solution) - cost(S)$;
  if(($\Delta E < 0$) || ($random < e^{\frac{-\Delta E}{T}}$)) {
    % Accept the attempted move as the new solution
    S = temp_solution;
  }
  M = M – 1;
}while(M $\geq$ 0)

C. Definition of a Move

A gate in the circuit is randomly selected for gate sizing. A random number is uniformly selected from the sequence of {-64, -32, -16, -8, -4, 0, 4, 8, 16, 32, 64}, and the selected gate is sized to this number.

D. Cooling Schedule

Several cooling functions currently used in simulated annealing were considered. They include Boltzmann-Weighted

Ensemble Averages [16], [19], [20] (Eq. (2)), logarithmic temperature schedule [19], [20] (Eq. (3)), fast annealing [19], [20] (Eq. (4)), exponential cooling schedule [20] (Eq. (5)), and linear schedule [20] (Eq. (6)).

$$T(k) = \frac{T_0}{\ln\ k} \tag{2}$$

$$T(k) = T_0 \cdot \frac{\ln\ k_0}{\ln\ k} \tag{3}$$

$$T(k) = \frac{T_0}{k} \tag{4}$$

$$T(k) = T_0 \cdot \alpha^{\mathrm{t}} \tag{5}$$

$$T(k) = T_0 - \alpha \cdot t \tag{6}$$

E. Objective Functions

Three objective functions were considered sequentially in the simulation runs of the simulated annealing algorithm for crosstalk-aware gate sizing. They are the metrics for delay of a gate, which is its electrical effort, circuit delay $D_{\mathrm{v(total)}}$ that combines the delay of a gate and the crosstalk noise (Eq. (7)), and dual-objectives of area and circuit delay E_1 (Eq. (8)) and E_2 (Eq. (9)); crosstalk noise depends on the size of the aggressor's driver, and the sizes of the victim's driver and receiver.

$$D_{\mathrm{v(total)}} = D_{\mathrm{v}} + X_{<\mathrm{e_a,e_v}>} \tag{7}$$

$$E_1 = 2 \cdot \sum_{i=1}^{n} gs_i + \sum_{i=1}^{m} D_{\mathrm{v(total)}} \tag{8}$$

where gs_i is the size of the gate, and m is the number of gates on the critical path.

$$E_2 = 2 \cdot \sum_{i=1}^{n} (gs_i + D_{\mathrm{v(total)}}) \tag{9}$$

where gs_i is the size of the gate.

F. Crosstalk Noise Metric

The metric for crosstalk $X_{<\mathrm{ea,ev}>}$ used is determined by the size of the aggressor driver S_{a1}, and the sizes of the victim's driver S_{v1} and receiver S_{v2}; see equation (10). $X_{<\mathrm{ea,ev}>}$ is the slowdown of the path on e_v, due to the crosstalk delay induced on it by e_a; K represents the proportionality constant that determines that determines the impact of crosstalk on weakly-driven gates [1].

$$\text{Crosstalk noise } X_{<\mathrm{ea,ev}>} = K \cdot \frac{S_{\mathrm{a1}}}{S_{\mathrm{v1}} \cdot S_{\mathrm{v2}}} \tag{10}$$

Only crosstalk coupling between two edges, an aggressor e_{a} and a victim e_{v}, were considered; this is represented by $< e_{\mathrm{v}}, e_{\mathrm{a}} >$. This simplifies the analyses of crosstalk effects between multiple wires, where the crosstalk interference between several interacting interconnects and their mutual effects result in nonlinear behavior. In instances where more than one interconnect has an impact on the crosstalk noise of a victim interconnect, they are modeled as separate pairs of interconnects, where the victim interconnect is a member of each of these pairs. Since an interconnect is represented by 2 gates, a driver gate n_1 and receiver gate n_2, the crosstalk effect of e_{a} on e_{v} is given by $< (n_{\mathrm{v1}}, n_{\mathrm{v2}}), (n_{\mathrm{a1}}, n_{\mathrm{a2}}) >$.

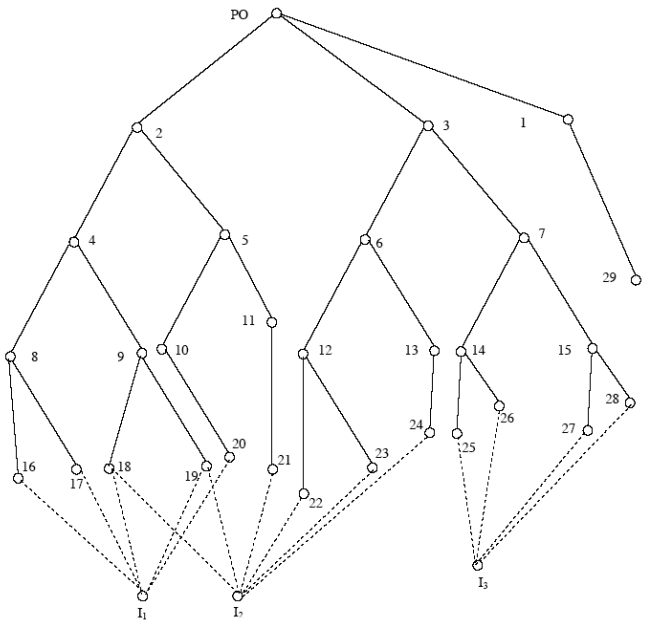

Fig. 1. Benchmark circuit for the simulations.

G. Delay Model

The metric used to determine delay of a gate i, D_{i}, is the electrical effort that depends on the gate fan-out; see equation (11) [1]. The parasitic delay and logical effort in the propagation metric for the linear delay model were ignored for since we are only considering the delay of a gate and ignoring the effects delay propagation on a path.

$$\text{Delay of a gate } i, D_{\mathrm{i}} = C_{\mathrm{out}}/C_{\mathrm{in}} \tag{11}$$

where C_{out} is the capacitive load driven the gate output, ignoring output parasitic capacitances, and C_{in} is the gate's input capacitance.

III. Simulation Results and Discussion

A. Benchmark Used

The benchmark circuit used is a tree with reconvergent fan-out; each node in the tree represents a logic cell (simple gate) and the each edge represents an interconnect between 2 gates. Nodes I_1, I_2, and I_3 denote the primary input drivers of the electronics circuit, and PO is the primary output gate of the circuit; see Fig. 1.

B. Results and Discussion

In our simulation runs, α and β were chosen to be unity; M was also chosen to be 1. The value of $maxtime$ chosen was 1000. The fast annealing cooling schedule was chosen in our simulations due to its faster convergence.

TABLE I

GATE SIZES DETERMINED BY SIMULATED ANNEALING

Gate Number	Gate Size (no crosstalk)	Gate Size (crosstalk)	Gate Size (Critical Path/Total Delay)
PO	200	200	200/200
1	200	200	4/8
2	88	88	8/20
3	76	76	8/20
4	32	32	4/4
5	32	36	4/8
6	28	28	4/4
7	24	24	4/4
8	12	12	4/4
9	12	12	4/4
10	12	20	4/4
11	12	12	4/4
12	12	16	4/4
13	12	8	4/4
14	12	12	4/4
15	12	8	4/4
16	4	4	4/4
17	4	4	4/4
18	4	4	4/4
19	4	4	4/4
20	4	4	4/4
21	4	4	4/4
22	4	4	4/4
23	4	4	4/4
24	4	4	4/4
25	4	4	4/4
26	4	4	4/4
27	4	4	4/4
28	4	4	4/4
29	196	196	4/4
I_1	4	4	4/4
I_2	4	4	4/4
I_3	4	4	4/4

Table I indicates the sizes of the gates in the circuit when the circuit delay is minimized while ignoring crosstalk, when the circuit delay is minimized while considering crosstalk, when the circuit delay on the critical path is minimized with crosstalk consideration along with area, and when the total gate delay in the circuit is minimized with crosstalk consideration along with area.

The delay on the critical path of the circuit at the start of the simulated annealing is found to be 59 when crosstalk is ignored, and its critical path delay is 85.5 when crosstalk is considered; this initial solution is obtained by setting all gate sizes to minimum. The critical path delay of the circuit, without considering crosstalk in the simulation run of the simulated annealing algorithm, was found to be 16.6894, and its total crosstalk cost is 11.0618; the total crosstalk cost is found to be the sum of all crosstalk noises in the gates. When crosstalk slowdown is considered in determining the critical

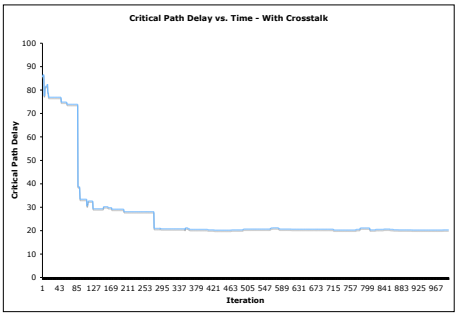

Fig. 2. Benchmark circuit for the simulations.

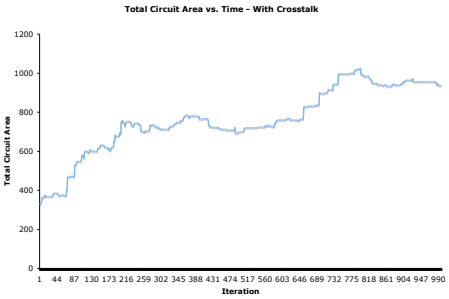

Fig. 3. Benchmark circuit for the simulations.

path delay of the circuit, it was found to be 22.304. When crosstalk is considered in the simulation run of the simulated annealing algorithm, its crosstalk cost was found to be 6.1972, and its circuit delay along the critical path is 19.9338.

Next, when the circuit is optimized for the dual objectives of critical path delay and total gate area, which is found by the sum of all gate widths, its crosstalk cost is 35.025 and its circuit delay is 56.5. Its area was found to be 336. When the area of the circuit is minimized along with the total gate delay of the circuit, which is determined by the sum of all gate delays, its crosstalk cost is 24.26, and its circuit delay is 35.25; its circuit area is 368.

Subsequently, when Eq. (8) is considered again, preference is lower crosstalk delay when determining if the move should be accepted. That is, when a gate is sized, and the overall cost is found to be the same, the crosstalk delay of the gates will be used as the tie-breaker. The crosstalk cost is found to be 1.91932, and the circuit delay is 20.275; its circuit area is calculated at 1244. Finally, Eq. (8) is considered yet again with preference given to smaller area. The crosstalk cost is found to be 5.12222, and the circuit delay is 19.8958; its circuit area is calculated at 528.

From the results indicated above, crosstalk is found to have a significant impact on the delay of the critical path in the

circuit; when crosstalk is considered in the first simulation run, the critical path delay increased by 33.64%. Also, there exists a trade-off between area and circuit delay on the critical path of the circuit, likewise with area and total gate delay in the circuit; see Figures 2 and 3. When the gates in the circuit are sized to achieve minimum area and total gate delay, it was found to have much better delay performance (improvement of 75.19%) for a marginal trade-off in area (increase of 9.52%) compared to sizing the gates to minimize area and the critical path delay on the circuit. We believe that this is because optimizing the circuit for area and critical path delay leads to more local optimizations compared to minimizing area and total gate delay. Lastly, after optimizing for area and critical path delay when crosstalk is considered, the delay is found to improve by more than four-fold.

IV. Future Work

Future avenues of exploration include the following. The dependencies and interactions between each step in the VLSI design flow, and the impact of changes (such as gate sizing) made in a step on others, should be investigated to improve the efficiency of existing design methodology. For example, the buffering of a net may prevent a gate form being optimally sized, and the sizing/resizing of a gate can affect subsequent buffer optimization. Consequently, modifications made in each step can be used to facilitate the global optimization of the circuit's performance, as opposed to local optimizations for a particular design requirement [7], [8].

Buffers, or repeaters, can be inserted into segments of a wire or sized to regenerate signals [1]. They are used to improve the drive strength of gates by restoring signal levels from their attenuated values without modifying the logical organization of the integrated circuit. Hence, interconnects with inserted buffers can be laden with larger loads. They can also provide isolation between the critical path and their heavy loads, such that these loads are perceived as lightweight buffers. However, it suffers from the conflicting effects that are similar to gate sizing; namely, the insertion or sizing of buffers can improve drive strength, reduce signal attenuation, and provide isolation, but also increase the area of the integrated circuit that may result in lower yield [8]. Some buffer optimization techniques that are sequentially or concurrently performed with gate sizing include the generalized De Morgan transformation [21], interleaved buffer insertion and transistor sizing algorithm [22], [23], technology-independent delay optimization using logical effort [24], simultaneous gate sizing and path based buffer insertion [25], and buffer resizing for aggressor and victim in coupled clock networks to minimize clock jitter [3].

Simultaneous buffer optimization and gate sizing can be used to achieve the multiple objectives of minimization in area, delay, crosstalk noise, and power while incrementally iterating between the various stages of the VLSI physical design flow. This interleaving of steps such as floorplanning, placement, and routing allows gradual refinement of the layout [26]. Techniques to be investigated may include simultaneous gate sizing and path based buffer insertion for area minimization [25], and a game theoretic integrated gate sizing and buffer insertion that models progressive priced auction to determine Nash equilibrium for power minimization [27]. However, caution must be taken to avoid simple feedback loops between logical and physical design such that the iterative process is highly time-consuming and convergence in optimal solutions is not guaranteed [28].

In addition, the following techniques can be concurrently or sequentially used with gate sizing and buffer optimization. They are power consumption minimization using variable input delay CMOS logic [29], timing performance improvement using logic replication [30], and improvement of robustness against soft errors using a combination of error masking, and error detection and recovery [31]. Also, global logical transformations to minimize delay can be considered along with gate sizing and buffer optimization techniques that do not modify the netlist and do so locally, respectively. Such techniques include cell replication and implicant-based circuit transformations [28]. This combination may mitigate the effect of crosstalk delay and maintain signal integrity without consuming too much power in an optimal area. Consequently, an over-dependence on buffer insertion and gate sizing can be avoided, and reduce area overheads that exacerbate the Rent limit problem due to larger gates and inserted buffers [32].

Lastly, the models and metrics for delay and crosstalk estimation can be further improved. For crosstalk models, the 2-π crosstalk model or the more accurate coupled noise pulse amplitude model [33] can be used; crosstalk metrics such as coupling capacitance and crosstalk sensitivity [34] can also be considered. Delay models that may considered include the Elmore delay model [35] to estimate wire and gate delay for timing analysis [36]–[38], and the inclusion of wire length as a probability distribution in Elmore modeling [38].

V. Conclusion

To deal with the problem of timing closure in ultra-deep sub-micron design of complex integrated circuits [21], a simulated annealing algorithm has been proposed to perform gate sizing to reduce area and delay, while considering crosstalk effects. This global optimization technique averts local optimizations that result in sub-optimal solutions by considering area minimization with total gate delay optimization. The authors noted the trade-off between area and circuit delay, and suggest that these trade-offs that lie on the Pareto optima should be considered in the design to meet chip size constraints and performance requirements.

Acknowledgment

The authors would like to thank the University of Southern California's Department of Electrical Engineering for their support.

References

[1] N. H. E. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 3rd ed. Boston, MA: Pearson Educational, Inc., 2005.

[2] Semiconductor Industry Association, "International Technology Roadmap for Semiconductors, 2005 Edition: Interconnect," Semiconductor Industry Association, Tech. Rep., 2005.

[3] M.-F. Hsiao, M. Marek-Sadowska, and S.-J. Chen, "Minimizing coupling jitter by buffer resizing for coupled clock networks," in *Proceedings of the 2003 IEEE International Symposium on Circuits and Systems (ISCAS '03)*, vol. 5. Bangkok, Thailand: IEEE Press, 25–28 May 2003, pp. V–509 – V–512.

[4] Semiconductor Industry Association, "International Technology Roadmap for Semiconductors, 2005 Edition: Executive Summary," Semiconductor Industry Association, Tech. Rep., 2005.

[5] Semiconductor Industry Association, "International Technology Roadmap for Semiconductors, 2005 Edition: Design," Semiconductor Industry Association, Tech. Rep., 2005.

[6] Medea+, "Medea+ Design Automation Roadmap," Medea+, Paris, France, Tech. Rep., July 2005.

[7] J. Cong and M. Sarrafzadeh, "Incremental physical design," in *Proceedings of the 2000 International Symposium on Physical Design (ISPD '00)*. San Diego, CA: ACM Press, 9–12 April 2000, pp. 84–92.

[8] O. Coudert, J. Cong, S. Malik, and M. Sarrafzadeh, "Incremental cad," in *Proceedings of the 2000 IEEE/ACM International Conference on Computer Aided Design (ICCAD '00)*. San Jose, CA: IEEE Press, 5–9 November 2000, pp. 236–243.

[9] A. Farrahi, D. Hathaway, M. Wang, and M. Sarrafzadeh, "Quality of eda cad tools: Definitions, metrics and directions," in *Proceedings of the 2000 International Symposium on Physical Design (ISPD '00)*. San Jose, CA: IEEE Press, 20–22 March 2000, pp. 395–405.

[10] M. Marek-Sadowska, "Performance driven layout. Available at http://www.ucop.edu/research/micro/," University of California, Santa Barbara, Santa Barbara, CA, 2000–2001, viewed 11 April, 2006. [Online]. Available: http://www.ucop.edu/research/micro/

[11] O. Coudert, "Gate sizing for constrained delay/power/area optimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, no. 4, pp. 465–472, December 1997, special issue on low power electronics and design.

[12] ——, "Gate sizing: A general purpose optimization approach," in *Proceedings of the 1996 European Design and Test Conference (ED&TC 96)*. Paris, France: IEEE Press, 11–14 March 1996, pp. 214–218.

[13] D. Sinha, N. V. Shenoy, and H. Zhou, "Statistical gate sizing for timing yield optimization," in *Proceedings of the 2005 IEEE/ACM International Conference on Computer-Aided Design (ICCAD–05)*. San Jose, CA: IEEE Press, 6–10 November 2005, pp. 1037–1041.

[14] D. Whitley, S. Rana, and R. Heckendorn, "Representation issues in neighborhood search and evolutionary algorithms," in *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications*, D. Quagliarella, J. Périaux, C. Poloni, and G. Winter, Eds. Chichester, West Sussex, England: John Wiley & Sons, 1998, ch. 3, pp. 39–57.

[15] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, April 1997.

[16] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 13 May 1983.

[17] A. Corana, M. Marchesi, C. Martini, and S. Ridella, "Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm," *Journal of Econometrics*, vol. 60, no. 1–2, pp. 65–69, 1994.

[18] W. L. Goffe, G. D. Ferrier, and J. Rogers, "Global optimization of statistical functions with simulated annealing," *ACM Transactions on Mathematical Software*, vol. 13, no. 3, pp. 262–280, September 1987.

[19] L. Ingber, "Global optimization of statistical functions with simulated annealing," *Journal of Control and Cybernetics*, vol. 25, no. 1, pp. 33–54, September 1996.

[20] Y. Nourani and B. Andresen, "A comparison of simulated annealing cooling strategies," *Journal of Physics A: Mathematical and General*, vol. 31, no. 41, pp. 8373–8385, 1998.

[21] S. Chakraborty and R. Murgai, "Layout-driven timing optimization by generalized de morgan transform," in *Proceedings of the 7^{th} Asia and South Pacific Design Automation Conference and the 15^{th} International Conference on VLSI Design, 2002 (ASP-DAC 2002)*. Bangalore, India: ACM Press, 7–11 January 2002, pp. 647–654.

[22] Y. Jiang, S. S. Sapatnekar, C. Bamji, and J. Kim, "Interleaving buffer insertion and transistor sizing into a single optimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 4, pp. 625–633, December 1998.

[23] ——, "Combined transistor sizing with buffer insertion for timing optimization," in *Proceedings of the IEEE 1998 Custom Integrated Circuits Conference (CICC '98)*. Santa Clara, CA: IEEE Press, 11–14 May 1998, pp. 605–608.

[24] P.-M. Seidel, "Technology-independent delay optimization of complex CMOS circuitries," in *Proceedings of the 45th Midwest Symposium on Circuits and Systems, 2002 (MWSCAS-2002)*. Tulsa, Oklahoma: ACM Press, 4–7 August 2002, pp. II–314 — II–317.

[25] C. N. Sze, C. J. Alpert, J. Hu, and W. Shi, "Path based buffer insertion," in *Proceedings of the 42^{nd} Annual ACM/IEEE on Design Automation Conference (42^{nd} DAC)*. Anaheim, CA: ACM Press, 5–9 January 2005, pp. 509–514.

[26] R. Carragher, R. Murgai, S. Chakraborty, M. R. Prasad, A. Srivastava, and N. Vemuri, "Layout-driven logic optimization," in *Proceedings of the 10^{th} International Workshop on Logic-based Program Synthesis and Transformation (LOPSTR 2000)*. London, UK: Springer–Verlag, 24–28 July 2000, pp. 270–276.

[27] A. K. Murugavel and N. Ranganathan, "Gate sizing and buffer insertion using economic models for power optimization," in *Proceedings of the 17^{th} International Conference on VLSI Design, 2004 (VLSID '04)*. Mumbai, India: IEEE Press, 5–9 January 2004, pp. 195–200.

[28] H. Hartje, I. Neumann, D. Stoffel, and W. Kunz, "Cycle time optimization by timing driven placement with simultaneous netlist transformations," in *Proceedings of the 2001 IEEE International Symposium on Circuits and Systems (ISCAS '01)*, vol. 5. Sydney, Australia: IEEE Press, 6–9 May 2001, pp. 359–362.

[29] T. Raja, V. D. Agrawal, and M. L. Bushnell, "Variable input delay CMOS logic for low power design," in *Proceedings of the 18^{th} International Conference on VLSI Design, 2005 (VLSI Design 2006)*. Kolkata, India: IEEE Press, 3–7 January 2005, pp. 598–605.

[30] G. Beraudo, "A path based algorithm for timing driven logic replication in FPGA," Master's thesis, University of Illinois at Chicago, Chicago, Illinois, 2002.

[31] S. Krishnamohan and N. R. Mahapatra, "Combining error masking and error detection plus recovery to combat soft errors in static CMOS circuits," in *Proceedings of the 2005 International Conference on Dependable Systems and Networks (DSN '05)*. Yokohama, Japan: IEEE Press, 28 June–1 July 2005, pp. 40–49.

[32] P. H. Madden, "Supersized VLSI: A recipe for disaster," in *Proceedings of the Electronic Design Process 2005 Workshop (EDPS '05)*. Monterey, CA: IEEE Press, 7–8 April 2005, organized by the Electronic Design Process Subcommittee of IEEE Computer Society's Design Automation Technical Committee.

[33] B. F. Carlo Guardiani, Christiano Forzan, "Modeling the effect of wire resistance in deep submicron coupled interconnects for accurate crosstalk based net sorting," in *Proceedings of the 1998 International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS '98)*. Lyngby, Denmark: Technical University of Denmark, 7–9 October 1998.

[34] J. Kim, C. Bamji, Y. Jiang, and S. Sapatnekar, "Concurrent transistor sizing and buffer insertion by considering cost-delay tradeoffs," in *Proceedings of the 1997 International Symposium on Physical Design (ISPD '97)*. Napa, CA: IEEE Press, 14–16 April 1997, pp. 130–135.

[35] W. C. Elmore, "The transient response of damped linear networks with particular regard to wideband amplifiers," *Journal of Applied Physics*, vol. 19, pp. 55–63, January 1948.

[36] L. van Ginneken, "Buffer placement in distributed rc–tree networks for minimal elmore delay," *Proceedings of the 1990 IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 865–868, May 1990.

[37] C. J. Alpert, A. Devgan, and S. T. Quay, "Buffer insertion with accurate gate and interconnect delay computation," in *DAC '99: Proceedings of the 36th ACM/IEEE conference on Design automation*. New York, NY, USA: ACM Press, 1999, pp. 479–484.

[38] V. Khandelwal, A. Davoodi, A. Nanavati, and A. Srivastava, "A probabilistic approach to buffer insertion," in *Proceedings of the 2003 IEEE/ACM International Conference on Computer Aided Design*. San Jose, CA: IEEE Press, 9–13 November 2003, pp. 560–567.