VLSI Design
1995, Vol. 3, No. 1, pp. 43–51
Reprints available directly from the publisher
Photocopying permitted by license only

Performance and Area Optimization of VLSI Systems Using Genetic Algorithms

XIAO-DONG WANG and TOM CHEN
Department of Electrical Engineering, Colorado State University, Ft. Collins, CO 80523

(Received July 26, 1993, Revised March 21, 1994

A new performance and area optimization algorithm for complex VLSI systems is presented. It is widely believed within the VLSI CAD community that the relationship between delay and silicon area of a VLSI chip is convex. This conclusion is based on a simplified linear RC model to predict gate delays. In the proposed optimization algorithm, a nonlinear, non-RC based transistor delay model was used which resulted in a non-convex relationship between the delay and the silicon area of a VLSI chip. Genetic algorithms are better suited for discrete, non-convex, non-linear optimization problems than traditional calculus-based algorithms. By using the genetic algorithms in the performance and area optimization, we are able to find the optimal values for both delay and silicon area for the ISCAS benchmark circuits.

Key Words: *Area and Performance optimization; Transistor Sizing; Genetic algorithms*

1. INTRODUCTION

The techniques for performance and area optimization of VLSI systems can be divided into two categories. One is to change the circuit structure by re-synthesizing or re-timing the target system. The other is to change the transistor sizes of the circuit so that the driving and load conditions in the circuit is optimal. The latter approach does not involve any topological changes and is often referred to as "*transistor sizing*".

Transistor sizing is to find an optimum set of transistor sizes in a circuit so that the circuit performance and/or circuit area are optimized. The size of a transistor includes two components: the transistor channel length, L, and the channel width, W. Because the transistor channel length is often fixed to its minimum value, varying transistor sizes to change the circuit performance is often accomplished by varying the transistor widths. The objective function used most often has been $f(A, T) = A * T$, where A is the total active area and T is the longest delay of the circuit.

Most of the existing transistor sizing algorithms [1, 4, 8, 2] uses linear RC delay models in the timing analysis. Fishburn and Dunlop [2] concluded that the relationship between delay and size of a VLSI system is convex. More recently, Sapatnekar [3] and Dunlop [9] formalized the assumption of convexity and further improved the performance of their transistor sizing algorithms. However, these algorithms based on RC delay models suffer from a number of drawbacks. Firstly, the RC-based delay models lead to inaccurate estimations of circuit delays. 20%–30% deviation from the SPICE simulation in predicting delay by RC-based delay model is expected [13]. Secondly, these algorithms are calculus-based algorithms which have difficulties in optimizing a discrete search space. Even though Shyu [1] made improvements in dealing with the non-differentiality by expanding the definition of "gradient" to the extent of both differentiable and non-differentiable points, the algorithm in [1] were reported for having convergence problems on complex circuits and having difficulties in automatically finding an optimum parameter set. And finally, the convex relationship between the circuit size and the circuit delay may not hold if a more accurate, non-RC delay model is used as our experiments have shown [20]. Delay calculation proposed by Hofman and Kim [12] is accomplished by a lookup table. The table-lookup method for delay calculation does not have the

inaccuracy problems of RC models, but it often suffers from inflexibility in adapting to different technologies.

We used an analytic delay model [10] which is similar to the analytic delay model described by Weste and Eshraghian [11]. The main difference is that our delay model takes the input slew rate into account resulting in a more complex but more accurate delay model. Our analytic delay model has the delay prediction accuracy between 0.5% and 2% over a wide range of input slew rates, transistor sizes and output load capacitances compared to SPICE simulations.

The research presented in this paper is motivated by the fact that the relationship between the circuit delay and the circuit size may be non-convex if a more accurate delay model is used rather than the simple RC model. We present a VLSI performance and area optimization algorithm by finding a set of optimum transistor sizes for a given VLSI CMOS circuit. To effectively search for an optimal solution, a chain-like geno-structure, referred to as a *chromosome*, is used to emulate a circuit path structure. The information of the size of a gate on the given path is represented by the encoding of each gene on the chromosome. Search for global optimum is facilitated using the genetic algorithms [7]. Because the optimization does not take interconnects into consideration, it is mainly used for pre-layout optimization.

The experiments on the ISCAS benchmark circuits show that a substantial amount of improvement can be achieved in circuit delay, and very often, in both circuit delay and circuit size.

The remaining part of this paper is organized in 4 sections. In Section 2, a new critical path selection strategy used in our timing optimization is introduced. The organization of the timing and area optimization algorithm based on the genetic algorithms are described in Section 3. Experimental results on the ISCAS benchmark circuits are discussed in Section 4. Finally, in Section 5, concluding remarks and the future work are presented.

2. CRITICAL PATH SELECTION

Extracting critical paths from a given circuit is the first step before timing optimization can be performed. In timing and area optimizations, the choice for optimum set of transistor sizes often cannot be determined solely on one critical path concerned. The close interdependency between different paths through shared gates plays an important role in both timing and area optimization. We refer to the gates not shared by more than one path as "*unique gates*". By optimizing only the unique gates in a given critical path, the interdependency between critical paths is eliminated, thus, simplifying the optimization process.

TABLE 1
The distribution of unique gates for the 10 most critical paths in C432

path number	length (gates)	number of unique gates
1	17	17
2	17	3
3	17	2
4	17	2
5	17	2
6	17	2
7	17	2
8	17	2
9	16	1
10	17	1

circuit: c432

The selection of critical paths for timing and area optimization is based on the following rules:

- Select critical paths according to their rankings in path delays, starting from the most critical path.
- Select only the paths which have at least one unique gate.

Table 1 shows the number of unique gates for each of the 10 longest critical paths in the ISCAS benchmark circuit C432. The most critical path has 17 unique gates. The next critical path has only 3 unique gates. The number of unique gates in the rest of the critical paths decreases with the increase in the number of identified critical path. This situation is very common among most of the ISCAS benchmark circuits.

The advantages of critical path selection based on both the path length and number of unique gates are twofold:

- The identification of unique gates among the critical paths allows us to dramatically reduce the search space for optimizing a given circuit, and also enables us to isolate different critical paths during optimization. Such a simplification, though may not yield a globally optimized solution, will significantly reduce the complexity of the optimization problem.
- In general, by ranking the critical paths, longer critical paths are optimized first, fixing the unique gates in the shorter critical paths, thus

TABLE 2
The number of critical paths with at least one unique gate in the ISCAS85 benchmark circuits

circuit name	number of paths with unique gates
c432	37
c499	108
c880	161
c1355	54
c1908	80
c2670	95
c3540	95
c5315	63
c7552	89
c6288	15*

*For c6288, 5000 paths are searched. For all the other circuits, 10000 paths are searched

reducing the computational complexity of the shorter critical paths without affecting the longer critical paths.

Table 2 shows the number of critical paths with at least one unique gate among 10,000 critical paths extracted for each of the ISCAS benchmark circuits.

3. A TRANSISTOR SIZING ALGORITHM BASED ON GENETIC ALGORITHMS

Genetic algorithms (GAs) are discrete, probabilistic optimization algorithms. Results have shown [7] that genetic algorithms can search for the global optimum in a non-convex, discrete searching space. The genetic algorithms can operate on a diversity of coded strings ranging from binary, integer strings, to strings of alphabets, etc. Therefore, the genetic algorithms have been proposed to generate solutions to a wide range of problems [14], [7]. In particular, several optimization problems have been investigated. These include control system [15], function optimization [16], combinational problems [17], [18], test pattern generation [17], and VLSI floorplanning [19].

The problem of transistor sizing consists of two parts: timing analysis to extract critical paths from a circuit; and optimization of transistor sizes on the extracted critical paths. We map the transistor sizing problem to a problem for the genetic algorithms as follows:

- An extracted critical path is treated as a chromosome. The length of the chromosome is determined by the number of genes on the chromosome.
- A gate on a critical path is treated as a gene of the chromosome. A gene is encoded as an integer which represents the *size* of the corresponding gate on the critical path.
- Assuming only CMOS gates are used and the risetime and falltime of the output is balanced, the size of a gate is represented by the effective transistor channel width of the n-tree in the gate. For instance, a 2-input NAND gate, the effective transistor channel width of the n-tree is half of the n-type transistor channel width.

The optimization process is divided into the following steps:

1. Extract the critical paths and rank them in the order of path delays.
2. Take one of the critical paths which does not satisfy the delay requirement and generate the initial chromosome population.
3. Calculate the load capacitance of every gate on the critical path according to gate sizes.
4. Use the analytic delay model [10] to calculate the gate delay and the fitness values of the population according to the fitness function

$$f(A,T) = \frac{1}{A * T} \tag{1}$$

where A is the total active area and T is the path delay. The input slew rate of a gate is approximated to the risetime or falltime of the previous gate.

5. Calculate the average fitness value for the entire population.
6. If at least one of the chromosomes representing the corresponding path satisfies the timing and the area constraints, or the improvement on the average fitness value of the entire population is within a specified threshold, the optimization process is completed. Otherwise, take two chromosomes from the population according to a set of selection criteria related to their fitness values, and perform crossover or mutation operations to generate a pair of new chromosomes, i.e., a pair of new path configurations.
7. If the new paths are better in terms of their fitness values, they are kept in the population and two existing chromosomes are eliminated from the population according to a set of criteria. Otherwise, the new paths are eliminated.
8. Go to Step 3.

The above optimization algorithm is illustrated in Figure 1. The critical path of a given circuit is first extracted as shown in bold lines. In Figure 1, we assume that there are 17 gates on the critical path *b*-to-*y*. The size of each gate in the critical path is then initialized to a random size and the size of a gate in the critical path (an integer) is represented in a chromosome structure according to the order of the gate in the critical path. Many such chromosomes are generated in the same way to form an initial population. Genetic algorithms are then applied to the population to obtain the final optimized gate sizes which is shown in Figure 1 as the optimized chromosome.

There are several parameters which can determine the quality of the optimization using genetic algorithms. These parameters include the population size, P_{size}, the probability of using the crossover operator, P_{xover}, and the probability of using the mutation operator, P_{mut}. The population size has the effects on convergence behavior as well as the final results. Typically, the larger the population, the better the quality of the final results and the longer it takes to reach the final results. The probabilities of applying the crossover and mutation operators also have some impact on the quality of the final results. The optimal values of these probabilities need to be experimented.

The transistor sizing algorithm based on the genetic algorithm was implemented in C programming language with about 20,000 lines of code.

4. EXPERIMENTAL RESULTS

The transistor size optimizer was applied to the ISCAS85 benchmark circuits. A commercial $1.5\mu m$ CMOS technology was used as the target technology. We compared the optimized results to two configurations. Under the first configuration, all the transistor sizes are set to the minimum size allowed by the technology. Under the second configuration, all the transistor sizes are set to the sizes of a commercial standard cell library as if these benchmark circuits were implemented using the standard cell library. In most cases, our transistor size optimizer was able to find better solutions in terms of the objective function. In some cases, the circuit size and the circuit delay were both reduced compared to two configurations. The parameters for the genetic algorithms chosen for the experiments were:

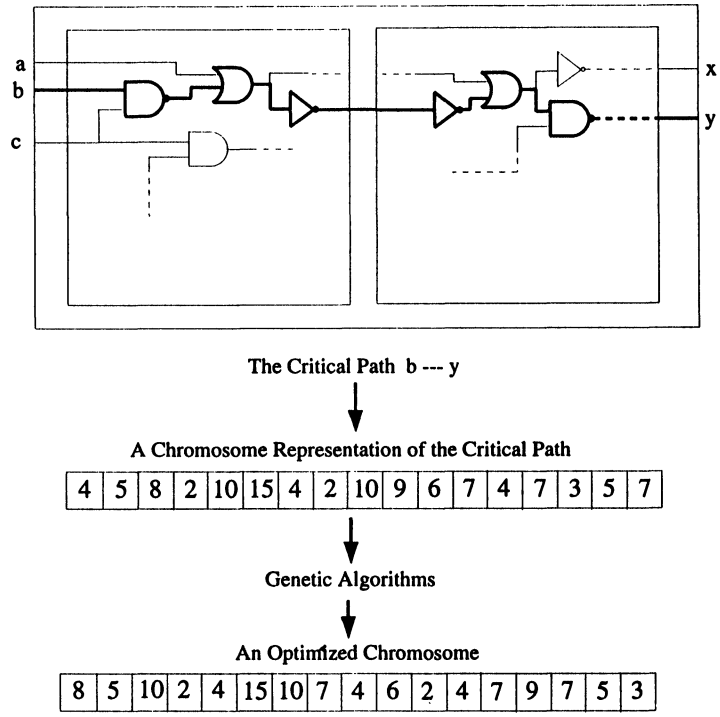

FIGURE 1 A simplified flow chart illustrating the optimization process.

TABLE 3
Results of the optimized ISCAS85 benchmark circuits with the minimum initial size for all the transistors.

circuit name	size (1cm *channel length) minimum size	optimized	FM	delay (ns) minimum size	optimized	FM
c432	0.3126	0.4408	−41.0%	39.66	26.56	38.3%
c499	0.6261	1.0141	−62.0%	24.13	19.88	17.6%
c880	0.5441	0.8333	−53.2%	29.89	24.71	17.3%
c1355	0.6873	0.6873	0%	35.21	35.21	0%
c1908	1.0054	1.1847	−17.8%	54.17	49.44	8.70%
c2670	1.5300	1.7393	−13.7%	44.03	40.75	7.40%
c3540	2.3713	2.6370	−11.2%	57.45	55.99	2.50%
c5315	3.4952	3.7007	−5.9%	69.11	63.87	7.60%
c7552	4.4694	4.7839	−7.0%	59.88	53.49	10.70%
c6288	No valid results because of memory allocation problem					

* The value of the size of a circuit is the sum of all the transistor widths in the circuit times the channel length of the transistors, so the unit for size is 1 cm (unit for width)* channel length.
** In all circuits, 10000 paths searched. Population size = 100 in the GA.

population size = 100, $P_{xover} = 0.6$, and $P_{mut} = 0.1$. The rationale for choosing these parameters will be discussed later in this section.

In order to compare the optimized results with two different configurations, we define the figure of merit (FM) as:

$$FM \text{ (figure of merit)} = \frac{\text{initial delay or size} - \text{optimized delay or size}}{\text{initial delay or size}} = \qquad (2)$$

Table 3 shows the comparison between the optimized ISCAS85 benchmark circuits and the minimum size configuration. The results are very much expected. We reduced the circuit delay at the cost of the increase of circuit size in 8 out of 10 cases. For C1355, our optimizer could not get any reduction in delay even if the circuit size is dramatically increased. Our optimizer ran out of memory during the optimization for C6288.

Table 4 shows the comparison between the optimized ISCAS85 benchmark circuits and the standard cell library implementation. In 7 out of 10 benchmark circuits, we achieved reduction in both the delay and the size of the circuits. The reduction in circuit sizes ranges from 4.6% to 31.3%. At the same time, 2.4% to 22.5%. For C499 and C1355, the circuit sizes were reduced at the cost of lesser amount of increase in circuit delay. This evidence demonstrates that the relationship between circuit size and circuit delay may not be convex as many others have originally believed.

TABLE 4
Results of the optimized ISCAS85 benchmark circuits with the initial implementation using a commercial standard cell library.

circuit name	size (1cm*channel length) standard library	optimized	FM	delay (ns) standard library	optimized	FM	% of gates size increase	size decrease	CPU time (second)
c432	0.4812	0.4408	8.4%	34.26	26.55	22.5%	43.4%	56.7%	4323.7
nc499	0.9057	1.0141	−12.0%	23.66	19.88	16.0%	50.5%	49.5%	5728.2
nc880	0.8737	0.8333	4.6%	28.79	24.71	14.2%	32.1%	67.9%	21151.7
nc1355	1.1922	0.8692	27.0%	34.05	37.38	−9.8%	39.5%	60.5%	12306.4
nc1908	1.6249	1.1847	27.1%	50.74	49.44	2.6%	38.5%	61.5%	20791.4
nc2670	2.3376	1.7393	25.6%	41.76	40.75	2.4%	49.1%	50.9%	16636.3
nc3540	3.5492	2.7225	23.3%	59.97	58.36	2.7%	45.1%	54.9%	46898.0
nc5315	5.0826	3.7007	27.2%	66.23	63.87	3.6%	40.0%	60.0%	22098.1
nc7552	6.9588	4.7839	3.13%	56.60	53.49	5.5%	29.8%	70.2%	36370.2
c6288	No valid results because of memory allocation problem								

* * The value of the size of a circuit is the sum of all the transistor widths in the circuit times the channel length of transistors, therefore, the unit for circuit size is "1 cm*channel length".
** In all circuits, 10000 paths searched. Population size = 100 in GA

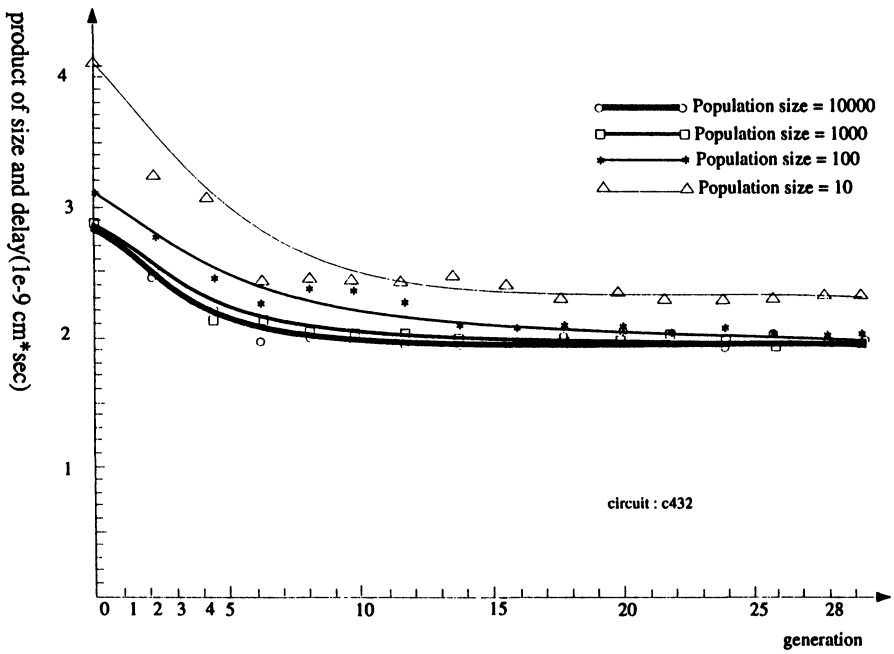

FIGURE 2 The effect of the population size on the objective values of a 17-gate path in C432.

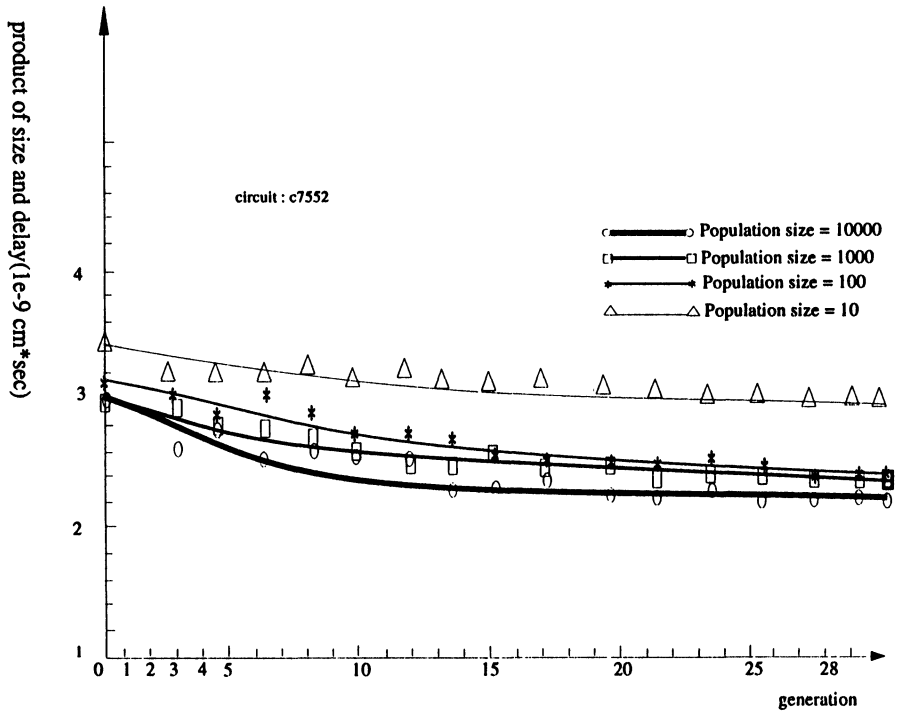

FIGURE 3 The effect of the population size on the objective values of a 43-gate path in C7552.

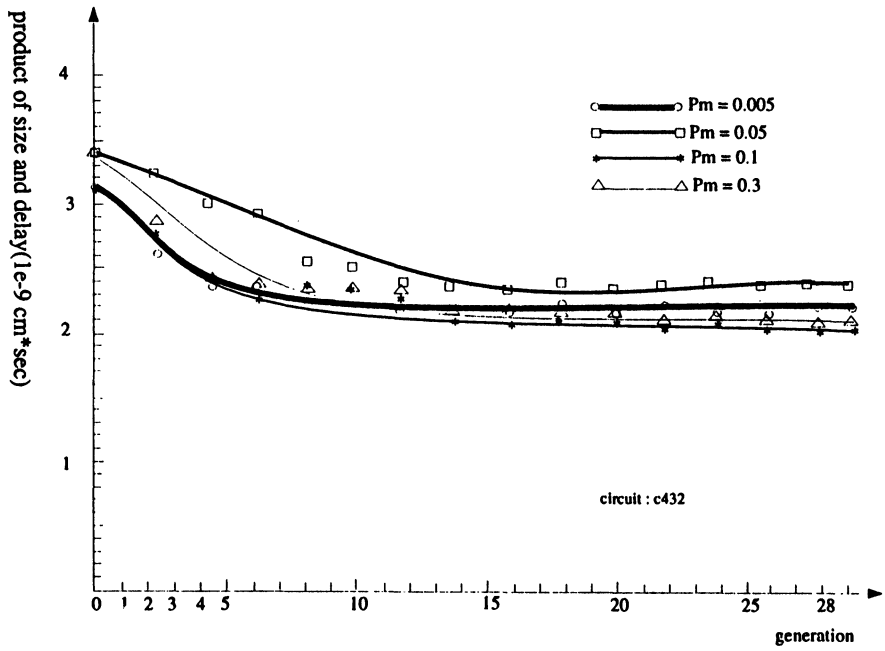

FIGURE 4 The effect of P_{mut} on the objective values of a 17-gate path in C432.

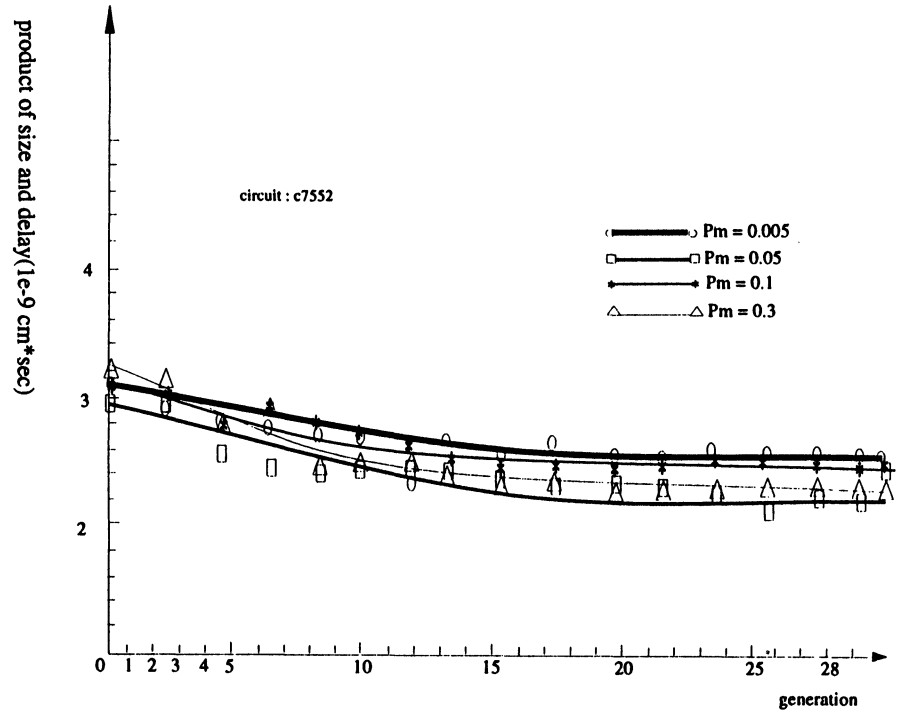

FIGURE 5 The effect of P_{mut} on the objective values of a 43-gate path in C7552.

During the optimization, we also kept track on the number of unique gates which increase in size or decrease in size as compared to the initial configuration. They are shown in the column "% of gates" with "size increase" and "size decrease" in Table 4. This is an indication of whether the genetic algorithm has successfully identified a small amount of critical gates to improve the circuit delay. For most cases, the amount of gate increase in size is smaller than the amount of gate decrease in size.

We also experimented various parameters for the genetic algorithm such as population size and the probabilities of performing crossover and mutation. Figure 2 shows the objective values, $A * T$, of a 17-gate path in C432 during the optimization in which genetic algorithms updates the generation of the population as new and better chromosomes are generated. The population size varied from 10 to 10,000 chromosomes. It is obvious that the objective values with larger population sizes are better than those with smaller population sizes. However, larger population size requires longer computation time to update each population. Our experiments show that the difference between the results with population size of 100 and the results with population size of 1000 and 10,000 are similar in terms of their objective values after a certain number of generations. This phenomenon is very typical in other benchmark circuits. Figure 3 shows the same relationship for C7552 with the path length of 43 gates.

The probability of performing crossover and mutation is less well behaved as compared to the population size in terms of finding the best value. Choosing the best value for a given circuit requires a certain amount of experiments. However, varying these two parameters does not have a significant impact on the final results as long as they are within a certain range. Figures 4 and 5 show the relationship between the objective values and the population generated by the genetic algorithm under various probabilities of performing mutation, P_{mut}. As these figures indicate, no definite conclusion can be drawn from the experiments. Our choice of P_{mut} is 0.1.

Although the delay model in [10] has taken the load capacitance into consideration, the load capacitance contributed by on-chip interconnects was not included in the experiments presented in this paper due to lack of layout information. The load capacitance contributed by the fanout gate capacitance was, however, included in our experiments.

This is the first time, to the best of our knowledge, that the entire set of ISCAS85 benchmark circuits has been used for evaluation. All previous publications on transistor sizing have given experimental results based on several non-public domain circuits to which we do not have access. Therefore, direct comparison of our results and previously published results is not possible.

5. CONCLUSION

By using a non-RC-based delay model, we have shown that the relationship between the delay and the size of a given circuit may not be convex as many others originally believed. Therefore, we generalized the transistor sizing problem as a non-convex optimization problem and used genetic algorithms to search for global optimum for both the delay and the size of a circuit. The results show that the genetic algorithms have several advantages over traditional calculus-based algorithms, such as the ability to search for global optimum in a discrete, non-convex search space. By using the proper path selection method, the search space can be greatly reduced to make it feasible for optimizing a large complex VLSI chip. We are especially encouraged by the results that show the opportunities to reduce both circuit delay and circuit size for a large of VLSI circuits of the transistor sizes are chosen properly.

One problem with our current experiments is that it fails to consider the impact of on-chip interconnects. As IC technology advances to deep-submicron feature size, the impact of on-chip interconnects on the overall chip optimization will be greatly increased. Our future work will include incorporating layout information about on-chip interconnects by back-annotating such information to the gate level. In addition, area estimation will include both the active area in terms of transistor sizes and the routing area.

References

[1] J. Shyu, A. Sangiovanni-Vincentelli, J.P. Fishburn, A.E. Dunlop. *Optimization-Based Transistor Sizing*. IEEE Journal of Solid-State Circuit, Vol. 23, No. 2 (April, 1988). pp. 285–289.

[2] J.P. Fishburn, A.E. Dunlop. *TILOS: A Posynomial Approach of Transistor Sizing*. Proc. Int. Conf. on CAD (1985), pp. 326–328.

[3] S.S. Sapatnekar, V.B. Rao, P.M. Vaidya, *A Convex Optimization Approach to Transistor Sizing for CMOS Circuits*. Proc. ICCAD 1992, pp. 482–485.

[4] K.S. Hedlund. *Aesop: A Tool for Automated Transistor Sizing*. Proc. 24th DAC (1987), pp. 114–120.

[5] S. Yen, D. Du, S. Ghanta, H.C. Du. *Efficient Algorithms for Extracting the K Most Critical Paths in Timing Analysis*. Proc. 26th IEEE DAC (1989), pp. 649–654.

[6] H. Chen, D.H.C. Du, L. Liu. *Critical Path Selection for Performance Optimization*. Proc. 28th ACM/IEEE DAC, 1991, pp. 547–550.

[7] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc., 1989.

[8] M. A. Cirit. *Transistor Sizing in CMOS Circuits*. Proc. 24th DAC (1987), pp. 121–124.

[9] A.E. Dunlop, J.P. Fishburn, D.D. Shugard. *Experiments Using Automatic Physical Design Techniques for Optimizing Circuit Performance*. Proc. Int. Symp. on Circuit and System (1990), pp. 847–851.

[10] Kelly R. Devereaux, T. Chen. *A New Model for Delay Analysis of CMOS Circuits*. Technical Report, Department of Electrical Engineering, Colorado State University, 1992.

[11] Neil Weste and Kamran Eshraghian. *Principles of CMOS VLSI Design—A System Perspective*. Second Edition, Addison-Wesley Publishing Co. 1993, pp. 208–226.

[12] M. Hofman, J.K. Kim. *Delay Optimization of Combinational Static CMOS Logic*. 24th DAC (1987), pp. 125–132.

[13] J.K. Ousterhout. *A Switching-Level Timing Verifier for Digital MOS VLSI*. IEEE Trans. on CAD of ICs, Vol. CAD-4(3) (April 1990), pp. 367–376.

[14] J.H. Holland. **Adaptation in Natural and Artificial System*. The University of Michigan Press, Ann Arbor.

[15] D.E. Goldberg. *Computer-aided gas pipeline operation using genetic algorithms and learning rules*. Ph.D. dissertation, Dept. Civil Eng., Univ. Michigan, 1983.

[16] A. Bethke. *Genetic algorithms as function optimizers* Ph.D. dissertation. Dept. Comput. Commun. Sci. Univ. Michigan, 1981.

[17] J.H. Aylor, J.P. Cohoon, E.L. Feldhousen, and B.W. Johnson. *GATE-A Genetic Algorithm for Compacting Randomly Generated Test Sets*. Int. Journal of Computer Aided VLSI Design 3, pp. 259–272 (1991).

[18] H.A.R. Davidson. *Applying genetic algorithm to the traveling salesman problem*. M.S. thesis, Dept. Computer Science, Colorado State Univ., 1989.

[19] J.P. Cohoon, S.U. Hegde, W.N. Martin, D.S. Richards. *Distributed Genetic Algorithms for the Floorplan Design Problem*. IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems, Vol. CAD-9, 1991,

[20] Xiao-dong Wang, *Optimization of VLSI System Using Genetic Algorithms*, M.S. Thesis, Department of Electrical Engineering, Colorado State University, 1993.

Biographies

TOM (WEI) CHEN received the B.Sc. degree in electronic engineering from Shanghai Jiao-Tong University, China, and the Ph.D. degree in electrical engineering from the University of Edinburgh, Scotland, U.K., in 1982 and 1986, respectively.

From 1987 to 1989, Dr. Chen was with Philips in Hamburg, Germany. From 1989 to 1990 he was with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology. He is currently an associate professor at the Department of Electrical Engineering at Colorado State University. Dr. Chen's research interests are in the areas of novel architectures in computation and signal processing, VLSI design and CAD techniques, ULSI/WSI architectures, and neural networks.

MR. XIAO-DONG WANG received his B.Sc. and M.S. degrees in electronic engineering from Qin-Hua University, China, in 1988, and Colorado State University in 1993. Currently, he is a design engineer with LSI Logic. His research interests are in the areas of CAD techniques for VLSI design.