Test Generation for Global Delay Faults

G. M. Luong and D. M. H. Walker
Dept. of Computer Science
Texas A&M University
College Station TX 77843-3112
Tel: (409) 862-4387
Fax: (409) 862-2758
Email: {gmluong, walker}@cs.tamu.edu

Abstract

This paper describes test generation for delay faults caused by global process disturbances. The structural and spatial correlation between path delays is used to reduce the number of paths that must be tested. Results are given for the ISCAS85 benchmarks.

Introduction

As time-to-market becomes increasingly important, automatic generation of integrated circuit delay tests becomes a essential. A large investment in hand-crafted delay vectors can only be justified today for very high volume parts. Many researchers have studied the problem of delay test [1] [2] [3] [4] [5] [6]. This prior work is based on the *gate delay fault* or *path delay fault* model, in which a gate or a path can have an arbitrary delay increase. The implication is that all gates or paths must be tested in order to achieve 100% fault coverage, and a great deal of effort must be placed on ensuring that the tests are robust. In practice it is very difficult to achieve 100% fault coverage using these metrics. To be even more accurate, the gate delay values can be bounded, or be modeled as a function of manufacturing process parameters [5] [7], incorporate factors like the state of the gate and coupling capacitance [8] [9], and the operating environment including supply voltage and temperature [10].

Previous work on delay test generation does not take advantage of all of the manufacturing knowledge in order to simplify the test problem. Delay faults are caused by both local and global process disturbances during IC manufacturing [ref Maly ProcIEEE]. An example of a local disturbance is a particle that causes a resistive bridge between two nodes, or a resistive contact. Since the disturbance is local in nature, the delay increase is local to a gate or net, An example of a global disturbance is variation in furnace temperature that causes a variation in transistor gate oxide thickness and drive current. Since the disturbance affects large parts of the chip, the delay increase is distributed over entire signal paths. We term delay faults caused by local disturbances *local delay faults* [11] and those caused by global disturbances *global delay faults*. Local and global delay faults approximately correspond to the traditional gate and path delay fault models. There are several test methods that identify most local delay faults, such as Iddq [ref Iddq], low voltage [ref Liao VTS96, HaoMcCluskey ITC93], and gate delay testing [ref gate delay tests]. In this paper we focus on generating path tests for global delay faults.

In order to incorporate knowledge of global disturbances into the test generation problem, the gate delay models must be a function of the manufacturing process parameters. Unlike prior work on physically-based delay test generation [10] [17], we have found the need for fairly complex models. The limited range of process parameter disturbances implies that path delays are also constrained. In other words, it is not possible for the delay of a 10-gate path to exceed that of a 100-gate path, assuming equal average gate delays. The implication is that if the 100-gate path is tested and passes the specification, there is no need to test the 10-gate path. For a given confidence level in the process parameter distributions, this can be thought of as the min-max gate delay model. This is shown conceptually in Figure 1. If all paths are enumerated and sorted by their worst-case length under process variations, only those paths that have some reasonable probability of exceeding the maximum allowable delay T_{max} must be tested. Typically this is a small fraction of all paths.

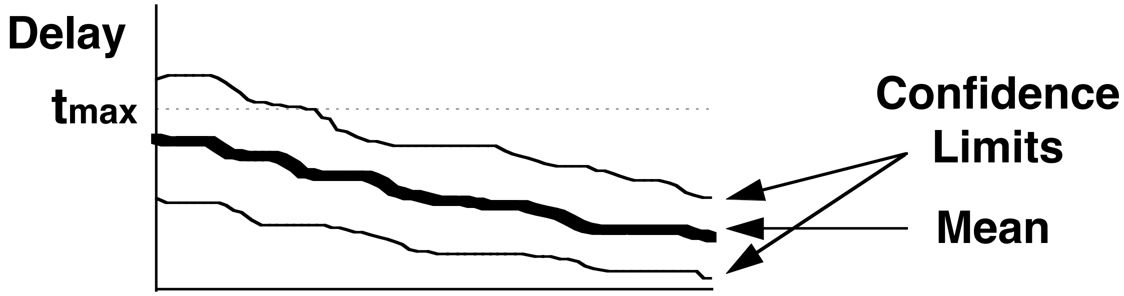

Figure 1: Paths longer than T_{max} under the min-max gate delay model are tested.

The above analysis is very pessimistic in that it ignores the structural correlation among paths and the spatial correlation of the process parameters. Structural correlation occurs when two paths share a common segment. Clearly the delay on the common segment will be nearly the same for both paths. Global process disturbances cause parameter changes that occur slowly across the chip. The result is that two nearby points will have very similar parameters, so for example, the delays of two neighboring gates will track closely. This constrains path delays in two ways. First, neighboring gates on a path track. It is not possible for one gate to have maximum delay, the next minimum delay, the next maximum delay, etc. Second, knowledge of neighboring path delays helps determine the delay on the current path. In the limit, if all gates had the same delay sensitivity to all process parameters, and there was 100% spatial correlation among process parameters (e.g. no spatial variation), then a test of the longest path would give 100% delay fault coverage of the entire chip. In practice, the sensitivity tracking and correlation are not this good. In this paper we focus on delay test generation for three situations. In the first case, we assume no spatial correlation among process parameters, so gate delays are independent variables. This is equivalent to the bounded delay test problem. In the second case, we assume 100% correlation among process parameters. This does not imply 100% correlation among gate delays since delay sensitivities vary by gate type and surrounding circuitry. The third case considered is partial spatial correlation, where the correlation is a function of distance. This is similar to the mismatch problem in analog circuit design.

The approach we take to test generation is to build physically-based delay models, incrementally generate the longest paths, and use them in an optimization-based test generation. If over the range of process variations, a path can exceed the delay specification, it is added to the test set. Each path in the test set and the spatial correlations form constraints on the process parameter space, since by implication all tested paths meet the timing specification. This constrained process parameter space in turn constrains the delays on untested paths. Paths are added to the test set until the parameter space is so constrained that no other paths can exceed the delay specification. Figure 2 shows this concept for a simple two-parameter case. The ellipses are confidence levels for the process parameters based on the mean and variance of the Normally-distributed parameters. The path 1 test constrains the parameters so that the low-low and high-high parameter combinations cannot occur. When path 2 is considered, its constraints are looser than those of path 1, which results in its maximum delay under the path 1 constraint being less than the specification. Therefore it is not added to the test set.

The paper is organized as follows. Section 2 describes construction of the physically-based gate delay models. Section 3 describes the test generation algorithm. Section 4 describes the results on the ISCAS85 benchmarks. Section 5 describes conclusions and future work.

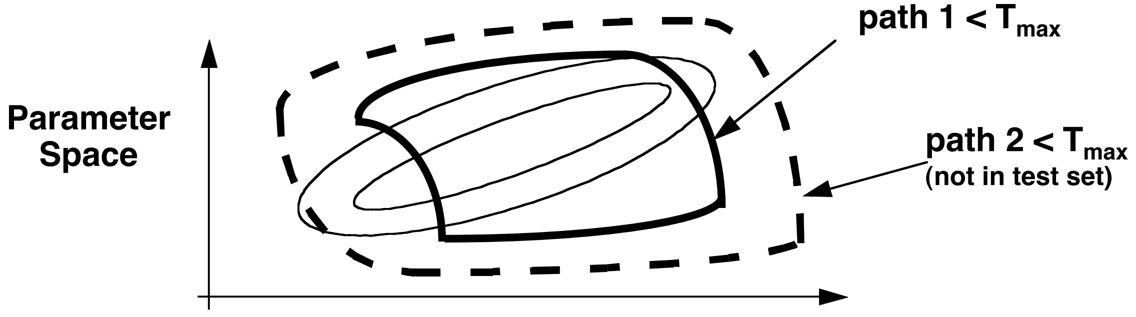

Figure 2: Path tests constrain process parameter space. Path 1 constrains the parameters so that path 2 meets the T_{max} specification.

Physically-Based Delay Models

Depend on the application and the required accuracy, the complexity of delay models can vary widely. A classification of delay models by accuracy is presented in [8]. The precise definition of delay can affect the model [12] [13]. In this work we measure delay from the midpoint of the input transition to the midpoint of the output transition. The input and output transition times are calculated using the 10% and 90% voltage levels. Separate output rising and falling delays and transition times are calculated for each input transition. We assume that a transition only occurs on a single gate input, although this is known to be optimistic [8]. The gate delay and output transition time are also a function of the input transition time and output load capacitance. In our delay models we assume the output load is constant capacitor. The gate delay and output transition time are also functions of the process parameters [14] [15]. Process disturbances are independent and Normally-distributed, which simplifies the model development.

Global process disturbances cause spatial variation in process parameters, resulting in spatial variation in transistor parameters, and thus gate delays. This variation becomes more important with larger die sizes and smaller features. A number of component mismatch models have been developed to describe this spatial variation [16] [17] [18]. Some models attempt to factor out the systematic mismatch from the random mismatch, approximating the systematic mismatch as a "linear gradient best fit plane" [18]. This has the disadvantage of requiring knowledge of the plane in order to compute actual mismatch values. Pelgrom models the mismatch as a Normal distribution whose variance is a function of the distance between two transistors [17]. We use the Pelgrom model to describe mismatch of a process parameter at different locations. Since we consider mismatch over the entire chip, we have modified the model so that the mismatch saturates at a background level, rather than continuing to rise with increasing distance. The Pelgrom model was developed to describe device parameter mismatch, but we believe it is also an adequate model for describing the underlying process parameter mismatch. We make the approximation that all transistors within a given gate or standard cell are subject to the same process parameters, only considering mismatch between gates or cells. This greatly simplifies the cell macromodel building described below.

Test generation requires that the delay model be compact and efficient. Based on the second-order nature of the MOS current equation, and previous modeling work [19] [15], we use a second-order polynomial macromodel to approximate the gate delay and transition time as a function of the process parameter disturbances. It is also known that the delay is monotonic with respect to the input transient time and the load capacitance [20] and that these relationships are adequately modeled by a second-order equation. With a slow input transition and fast output transition, it is possible for the gate delay to be negative if the logical threshold is not Vdd/2. With realistic transition times and output loads, this will not occur, and in particular no path delays will be negative.

Delay modeling is done in two steps: data collection and curve fitting. In the first step, an experimental design plan is performed. A sensitivity analysis is done to identify the most important process parameters that affect delay and transition time. A center cube composite plan [21] for these parameters is used to collect the data for model building. The *pdFab* [22] technology CAD framework is used here to construct and implement this plan, generating BSIM simulation model parameters for each data point. This simulation approach is used since it is very expensive to execute the plan using measured process data, and the simulations can provide good accuracy if the simulation models are tuned to the process. For our example process, seven sensitive process parameters are identified. Three are temperature parameters related to formation of the gate oxide. Two are doping concentration parameters related to Nwell formation and threshold voltage adjustment. One is a temperature parameter related Nwell annealing. One photolithography mask size parameter affects the transistor gate length. HSPICE simulations of the cell library are performed using the BSIM models to compute delay and output transition times. For the second step, SAS is used to perform stepwise regression for the second-order equations.

We found that the variation in input transition time and output load capacitance has much more influence on the gate delay and output transition time than do the process parameter variations. It is not possible to accurately incorporate all of these parameters into a single model equation. Instead we use a two-layer model-building approach. First, an experimental plan is built for the input transition time and load capacitance assuming nominal process parameters, and a second-order model built. The range of values is determined by the design applications. Second, for each point of the previous plan, another experimental plan for the process parameters is performed, and a second-order model built. This two-layer model can be thought as the delay variation caused by the process parameters superimposes on the primary delay determined by the input transition time and the load capacitance.

Delay Test Generation

Circuit optimization tends to compress the distribution of path delays in a circuit, so many paths are close to the maximum length [23]. In Figure 1, the result would be a flatter set of curves. Process variations mean that if each path is considered independently, a large number will have a worst-case delay that could exceed the specification, and therefore must be tested. By exploiting the correlations among path delays, we can avoid the need to test large numbers of paths. The idea of exploiting circuit structural and process parameter spatial correlation to reduce the number of paths that must be tested is known as *predictive subset testing* [24]. The concept is illustrated in Figure 3, which shows the delay relationship between two paths. The rectangle represents the confidence level of the two path delays if there is no correlation. Each path delay is independently determined by the process disturbances. If there is some correlation, the confidence region will be described by a “sausage shape”. The delay on one path will restrict the range of possible delays on other path. If the correlation is 100%, then the relationship will be a single curve, since the delay on one path will determine the delay on the second path.

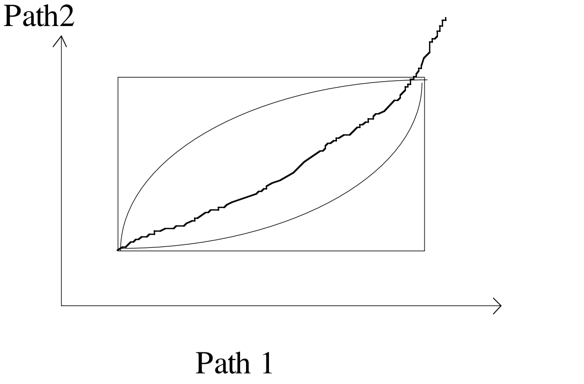

Figure 3: Delay relationship between two paths with different correlations

If the component delay values are constrained, any path make up of a combination of these components will also have its delay constrained. A simple bounded delay model shows this property [25]. For a given timing specification, a path needs to be tested or not depends on whether its maximum delay exceeds the design specification under the problem constraints. As described above, the constraints we consider are structural correlation, process parameter disturbances, and the spatial correlation among process parameters. Since our delay models are second-order equations, path delays composed of these models are highly nonlinear. The reason is that the gate delay and transition time are functions of the input transition time, which in turn is a second-order equation. We approximate the entire path as second-order by discarding higher-order terms of the composite delay equation. The problem of computing the maximum delay on a path then becomes a constrained nonlinear optimization problem. In order to fully explore the effect of spatial process parameter correlation, we consider three cases. The first case assumes no correlation among parameters. The second case assumes 100% correlation. The third case assumes spatial correlation using our modified Pelgrom model. process parameters are 100% correlated, thus all are keep track each other completely. The third case assumes the process parameters are spatial correlated, so the variations between them are limited.

The flowchart of the test generation algorithm is shown in Figure 4. For a given cell library, the worst-case gate delays are computed using the delay models and process parameter variations. These are used to incrementally generate statically-sensitizable paths in the circuit. The path generation algorithm is the topic of another research project and beyond the scope of this paper. We generate paths in batches of 250 since this is reasonably small, but turns out to be sufficient for most problems. A nonlinear optimizer NPSOL [26] then computes the upper and lower delay bounds, and resorts the paths based on these more accurate delay values. The only thing required of the path generator is that it be sufficiently accurate so as not to miss any long paths.

Paths are placed into the test set one at a time, using the heuristic of selecting the paths in order of decreasing length. Each path placed into the test set becomes an additional constraint. In addition to the path constraints, the optimizer also uses constraints on the absolute values of process parameters, and in the case of partial correlation, the distance-based correlations. These constraints are used to compute the maximum delay of the next path to be considered. Paths that do not exceed the delay specification are skipped. The algorithm halts when it has sufficient confidence that all remaining untested paths will meet the specification. The heuristic we use is to fetch another batch of paths if any of the last 10 paths in a batch are placed in the test set, or if the delay of the shortest path in the batch is within a guardband of the specification.

Results

We applied the delay model building and test generation algorithm to the ISCAS85 benchmarks, using standard cell implementations from MCNC [27] implemented in a 0.8um Nwell CMOS process. Delay models were built for each standard cell in the library. The models were built to handle the range of load capacitance observed in the circuits. The range of the input transition time was obtained as follows. The minimum was obtained by applying a step function to the largest cell driving the smallest load, and measuring the output transition time. The maximum input transition time was obtained by applying a step function to the smallest cell driving the largest load, measuring the output transition time, feeding that transition into the cell again, and measuring the output transition again. The process parameters were allowed to vary over their 3 σ range. In order to check the accuracy of the model, the longest path in the C880 circuit was computed using both the delay models and HSPICE simulation using BSIM models generated by *pdFab* for 400 random process parameter combinations. The load capacitances were extracted from the layout. The comparison of the model prediction vs. HSPICE simulation is shown in Figure 5. The correlation between the model and the simulation is 0.99, with outliers being within ±22 percent of the actual value. Most of the residual error is due to the exclusion of less sensitive process parameters from the model. Additional parameters can be added to obtain higher accuracy, but at higher cost.

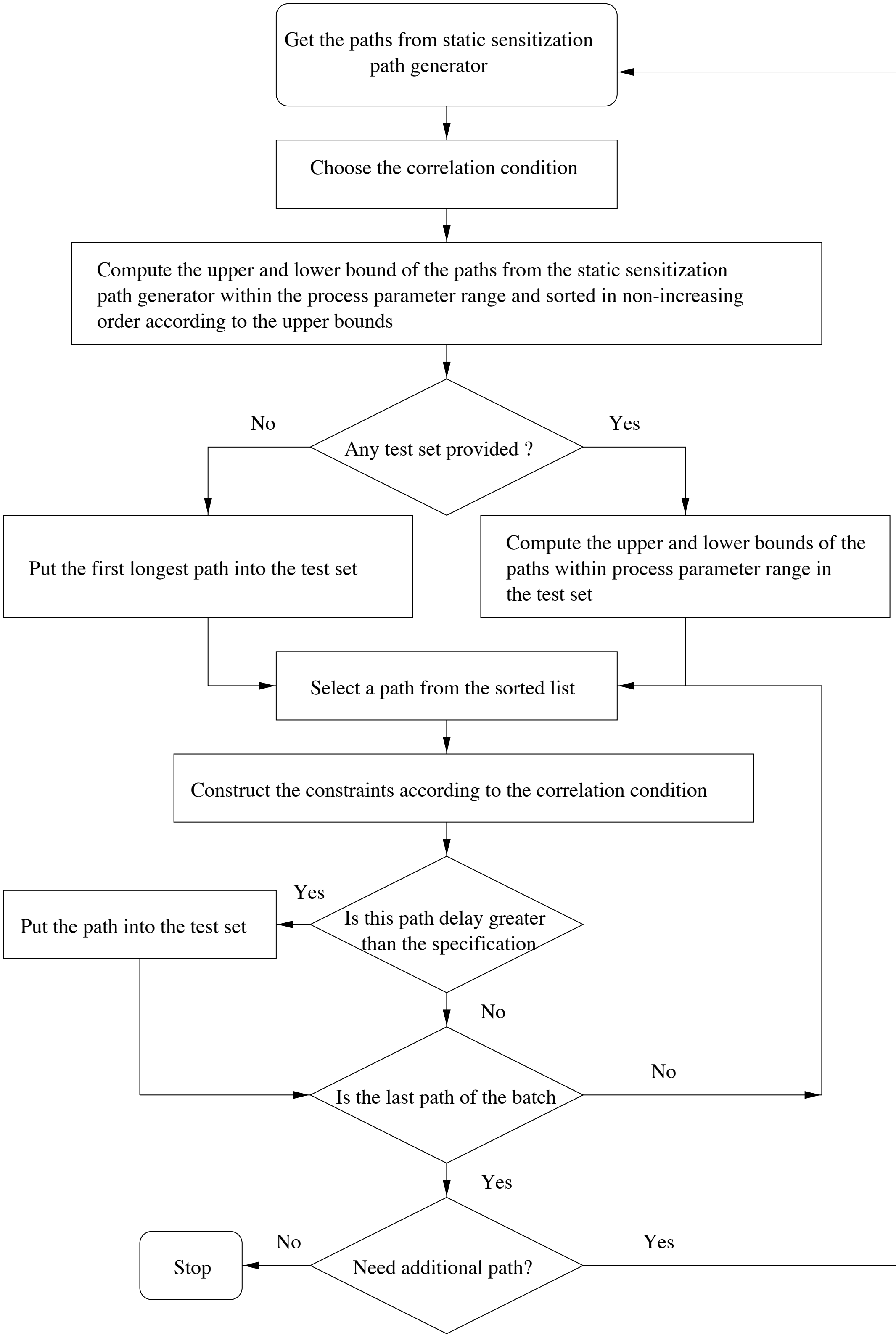
Get the paths from static sensitization path generator
Choose the correlation condition
Compute the upper and lower bound of the paths from the static sensitization path generator within the process parameter range and sorted in non-increasing order according to the upper bounds
Any test set provided ?
No
Yes
Put the first longest path into the test set
Compute the upper and lower bounds of the paths within process parameter range in the test set
Select a path from the sorted list
Construct the constraints according to the correlation condition
Yes
Put the path into the test set
Is this path delay greater than the specification
No
Is the last path of the batch
No
Yes
No
Stop
Need additional path?
Yes

Figure 4: Flowchart for path analysis

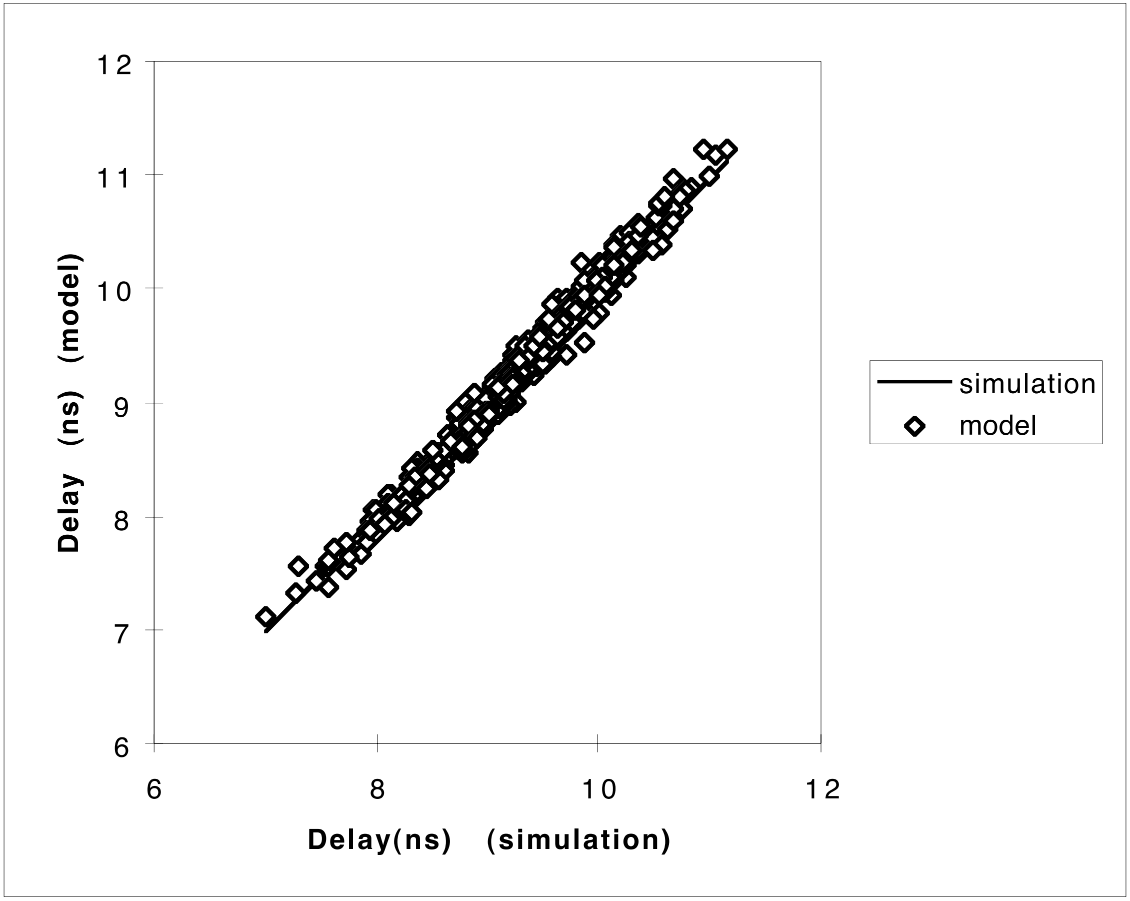

Figure 5: Model vs. simulation predictions of the longest path delay in C880 for 400 random process parameter combinations.

Test generation was performed for each ISCAS85 circuit using two different maximum delay specifications, corresponding to 50 percent yield and 25 percent yield for the no correlation case. The mismatch coefficients used in the partial correlation case are very conservative, so transistors on each side of the circuit are at the background correlation level. This background level is set to 50% of the total process parameter range. The constraints for this case approximate the sausage shape confidence level in Figure 3 by a set of hyperplanes. The test generation results for the two timing specifications are shown in Tables 1 and 2. CPU times are on a SPARC 5. As expected, case 1 (no correlation) results in the most paths that must be tested, while case 2 (100% correlation) requires the fewest paths, while case 3 (partial correlation) is somewhere in between. The path set in case 1 is the superset of the path set in cases 2 and 3. Note that even those the level of spatial correlation is very low, case 3 is much closer to case 2 than it is to case 1 in terms of the number of paths that must be tested. This indicates that the min-max delay model is too conservative and pessimistic. The results also show that a tighter test specification usually, but does not always increase the size of the test set. In several cases, the tighter specification results in a tighter path constraint, reducing the number of other paths that must be tested. For each case, the set of paths for each timing specification are very similar. The implication is that taking the union of both test sets to form a two-bin speed test will not significantly increase the test set size.

Tables 1 and 2 do not contain results for the C6288 circuit for cases 1 and 2, due to the high cost of the test generation. The reason is primarily that this circuit has four times more gates on its long paths than does the next largest circuit. It also contains a larger number of different gates. This combination causes the optimizer to take a large amount of memory and CPU time.

Table 1: No. of tested paths in the first 250 longest paths under timing specification I

Circuit	Specification (ns)	No. of paths tested	CPU time (hh:mm:ss)
c432	18.3635	132	30:01:06
c499	10.1104	142	4:39:05
c880	10.0352	74	5:08:03
c1355	15.4645	140	30:16:02
c1908	15.1	209	56:59:12
c2670	26.71	90	12:43:36
c3540	27.8143	208	36:27:53
c5315	20.9036	163	27:07:46
c7552	39.3016	175	24:22:03

(a) Case 1 (no correlation between circuit components)

Circuit	Specification (ns)	No. of paths tested	CPU time (hh:mm:ss)
c432	18.3635	1	0:30:38
c499	10.1104	3	0:14:07
c880	10.0352	2	0:25:27
c1355	15.4645	2	0:35:34
c1908	15.1	1	0:19:30
c2670	26.71	7	0:58:50
c3540	27.8143	2	1:26:53
c5315	20.9036	3	0:49:43
c6288	57.69	1	8:32:03
c7552	39.3016	1	0:11:40

(b) Case 2 (100 percent correlation between circuit components)

Circuit	Specification (ns)	No. of paths tested	CPU time (hh:mm:ss)
c432	18.3635	5	16:54:01
c499	10.1104	74	27:56:36
c880	10.0352	10	3:52:24
c1355	15.4645	13	30:21:51
c1908	15.1	27	90:22:14
c2670	26.71	8	6:48:42
c3540	27.8143	23	67:03:15
c5315	20.9036	13	39:00:44
c7552	39.3016	13	68:24:43

(c) Case 3 (partial correlation between circuit components)

Table 2: No. of tested paths in the first 250 longest paths under timing specification II

Circuit	Specification (ns)	No. of paths tested	CPU time (hh:mm:ss)
c432	15.1707	132	40:50:05
c499	8.29385	156	4:36:14
c880	8.2548	74	8:47:30
c1355	12.9358	140	22:16:03
c1908	12.35	210	95:07:05
c2670	21.8792	91	14:11:47
c3540	22.8154	207	45:10:42
c5315	17.3307	171	31:46:30
c7552	32.121	174	27:49:39

(a) Case 1 (no correlation between circuit components)

Circuit	Specification (ns)	No. of paths tested	CPU time (hh:mm:ss)
c432	15.1707	1	0:41:53
c499	8.29385	2	0:13:57
c880	8.2548	3	0:43:17
c1355	12.9358	1	0:33:49
c1908	12.35	1	0:15:48
c2670	21.8792	5	1:51:36
c3540	22.8154	2	1:31:37
c5315	17.3307	2	0:33:01
c6288	47.3355	2	15:12:04
c7552	32.121	1	0:06:44

(b) Case 2 (100 percent correlation between circuit components)

Circuit	Specification (ns)	No. of paths tested	CPU time (hh:mm:ss)
c432	15.1707	7	28:34:46
c499	8.29385	93	35:46:34
c880	8.2548	10	4:29:36
c1355	12.9358	13	28:45:34
c1908	12.35	32	86:48:16
c2670	21.8792	9	10:42:09
c3540	22.8154	34	85:54:30
c5315	17.3307	19	48:37:10
c7552	32.121	18	10:05:10

(c) Case 3 (partial correlation between circuit components)

Conclusions and Future Work

We have described a method for test generation method for delay faults caused by global process disturbances. The structural and spatial correlation between path delays is used to reduce the number of paths that must be tested. An investigation of three different process parameter correlation conditions and two different timing specifications was performed and applied to the ISCAS85 circuits. The results show that our physically-realistic modeling approach greatly reduces the number of paths that must be tested, compared to a path delay or min-max model.

Our algorithm currently only considers statically-sensitizable paths. Dynamic paths have the property that they may be always, sometimes, or never sensitizable depending on the process parameters. This can be handled in the test generation code in a straightforward manner, but the primary problem is supplying the path generator useful information that it can incrementally generate all the longest paths that might be dynamically-sensitizable.

Our path-by-path approach to testing is known to be optimistic in that multiple input transitions are required for the worst-case gate delay. When the clock and power supply network parasitics are considered, the worst case may require many paths to simultaneously be tested. Our path-by-path approach also behaves poorly in circuits where many paths may exceed the specification, even accounting for constraints. The solution in these cases is to approximate a complete test set by testing the longest paths through each gate.

Our delay model does not include the operating environment such supply voltage and operating temperature. In addition, we model interconnect as a constant capacitor. In reality it is a general impedance that is also affected by the process parameters. We are currently working on incorporating these additional effects, using our current model-building methodology.

Essentially all of the space and CPU time in the test generation goes to the nonlinear optimizer. The space complexity of the optimizer is $O(n^2)$ where n is the number of variables in the equations. We are exploring a number of approaches to dramatically improve performance and reduce space. We can use delay sensitivity and the results of previous optimizations to set the optimizer initial guess. In case 1, we can remove paths from the constraint set that do not share gates with the current path. In case 3, we can drop paths from the constraints that are beyond some distance away. Initial results show that rather than using the full set of constraints in case 3, it is sufficient to only use constraints between neighboring gates on the current path and neighboring gates in the test set paths. Internal optimizer mechanisms can be used to abort if a path exceeds the specification, rather than continuing the maximization. Similarly, if the optimizer is converging far below the specification, it should abort. We are still studying these optimizations to determine their benefit, and any error introduced by them.

References

[1] C. J. Lin, Sudhakar, and M. Reddy, "On delay fault testing in logic circuits," *IEEE Trans. CAD*, vol. 6, pp. 694-703, Sept. 1987.

[2] E. S. Park and M. R. Mercer, "Robust and nonrobust test for path delay faults in a combinational circuit," *Proc. Internation Test Conf.*, pp. 1027-1034, 1987.

[3] S. M. Reddy, C. J. Lin, and S. Patil, "An automatic test pattern generator for the detection of path delay faults," *Proc. International Conf. on CAD*, pp. 284-287, Nov. 1987.

[4] K. T. Cheng and H. C. Chen, "Delay testing for non-robust untestable circuits," *Proc. International Test Conf.*, pp. 954-961, Oct. 1993.

[5] M. Sivaraman and A. J. Strojwas, "Test Vector Generation For Parametric Path Delay Faults," *SRC Pub C95135*, May 1995.

[6] Y. Wu and A. Ivanov, "Accelerated path delay fault simulation," *Proc. IEEE VLSI Test Symposium*, pp. 1-6, April 1992.

[7] M. Sivaraman and A. J. Strojwas, “Timing Modeling for Accurate Timing Verification,” *SRC Pub C94372*, August 1994.

[8] P. Franco and E. J. McCluskey, “Three-pattern tests for delay faults,” *IEEE VLSI Test Symposium*, pp. 452-456, April 1994.

[9] A. Pierzynska and S. Pilarski, “Non-robust versus robust,” *Proc. International Test Conf.*, pp. 123-131, 1995.

[10] K. Krishna and S. W. Director, “The linearized performance penalty(lpp) method for optimization of parametric yield and reliability,” *SRC Pub C94501*, Oct. 1994.

[11] D. M. H. Walker, “Tolerance of delay faults,” *Proc. of the 1992 IEEE International Workshop on Defect and Fault Tolerance in VLSI Systems.*, November 1992.

[12] D. J. Pilling and J. G. Skalnik, “A circuit model for predicting transient delays in LSI logic systems,” *Proc. 6th Asilomar Conf.*, pp. 424-428, November 1972.

[13] A. F. Schwarz, “Propagation delay and threshold voltages in timing simulation,” in *Circuit theory and design*, R. Boite and P. Dewilde, Eds.: North-Holland, Amesterdam, August 1980, pp. 575-580.

[14] H.-F. Jyu and S. Malik, “Statistical delay modeling in logic design and synthesis,” *Proc. 31st Design Automation Conference*, pp. 126-130, 1994.

[15] V. Ramakrishnan and D. M. H. Walker, “IC performance prediction system,” *Proc. International Test Conf.*, pp. 336-344, 1995.

[16] C. Michael and M. Ismail, “Statistical modeling of device mismatch for analog MOS integrated circuits,” *IEEE J. of Solid-State Circuits*, vol. 3.27, Feb. 1992.

[17] M. J. Pelgrom, A. C. J. Duninmaijer, and A. P. G. Welbers, “Matching properties of MOS transistors,” *IEEE J. of Solid-State Circuits*, vol. 3.24, pp. 1433-1440, Oct. 1989.

[18] E. Felt, A. Narayan, and A. Sangiovanni-Vincentelli, “Measurement and modeling of MOS transistor current mismatch in analog IC's,” *Proc. IEEE ICCAD*, pp. 272-277, November 1994.

[19] M. Sivaraman and A. J. Strojwas, “Towards incorporating device parameter variations in timing analysis,” *Proc. EDAC-94*, pp. 338-342.

[20] E. D. Boskin, C. J. Spanos, and G. J. Korsh, “A method for modeling the manufacturability of IC designs,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 7, pp. 298-305, August 1994.

[21] G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for experimenters*: John Wiley & Sons, 1978.

[22] pdFab, *User's Reference Manual*: PDF Solutions.

[23] T. W. Williams, B. UnderWood, and M. R. Mercer, “The Interdependence Between Delay-Optimization of Synthesized Networks and Testing,” *Proc. 28th Design Automation Conf.*, pp. 87-92, 1991.

[24] J. B. Brockman and S. W. Director, “Predictive subset testing: optimizing IC parametric performance testing for quality, cost, and yield,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 2, pp. 104-113, August 1989.

[25] W. K. Lam, A. Saldanha, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, “Delay fault testing: trading fault coverage, test set size, and performance,” *Technical Report No. 93-26*, March 1993.

[26] P. G. etal., *User's guide for NPSOL: A FORTRAN package for nonlinear programming*. Tech. Rep. SOL 86-2: Stanford University.

[27] F. Brglez and H. Fujiwara, “A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran,” *Proc. IEEE Int. Symposium on Circuits and Systems; Special Session on ATPG And Fault Simulation*, June 1985.