

Timing-Reasoning-Based Delay Fault Diagnosis

Kai Yang, Kwang-Ting Cheng
University of California, Santa Barbara
{kyang, timcheng}@ece.ucsb.edu

Abstract

In this paper, we propose a timing-reasoning algorithm to improve the resolution of delay fault diagnosis. In contrast to previous approaches which identify candidates by utilizing only logic conditions, we propose a timing-simulation-based method to perform the candidate reasoning. Based on the circuit timing information, we identify invalid candidates which cannot maintain the consistency of failure behaviors. By eliminating those invalid candidates, the diagnosis resolution can be improved. We then analyze the problem of circuit timing uncertainty caused by the delay variation and the simulation model. We calculate a metric, named *invalid-probability*, for each candidate. Then we propose a candidate-ranking heuristic which is robust with respect to such sources of timing uncertainty. By ranking the candidates based on their *invalid-probability*, we can improve the candidate first-hit-rate of the traditional critical path tracing (CPT) technique. To demonstrate the efficiency of the proposed method, we have developed a timing diagnosis framework which can simulate the real diagnosis process to evaluate and compare different algorithms.

1 Introduction

Due to various deep submicron effects, a circuit may fail to operate at the desired clock frequency. Timing failure analysis is the procedure used to locate the source of timing failures. The *resolution* and the *first-hit-rate* of the candidates which are reported by the delay-fault diagnosis process will determine the efficiency of timing failure analysis. The resolution is defined as the ratio of the number of real fault sites to the total number of the reported candidates. The first-hit-rate is defined as the number of candidates we need to investigate before hitting the real one. Unfortunately, the existing delay-fault diagnosis methodologies suffer from poor resolution or low scalability.

The existing methods can be classified into two categories: (1) Fault-model-based diagnosis [1–5], and (2) Reasoning-based diagnosis [6–8]. For fault-model-based diagnosis, the cause of a timing failure is modeled as a fault such as a transition fault or a path-delay fault. The diagnosis procedure is invoked to identify the faults that match the measured responses with respect to the applied test vectors. For reasoning-based diagnosis, the cause of a timing failure is assumed to be extra delay at one or more locations in the circuit. The diagnosis procedure is implemented to identify all possible locations of the extra delay.

The fault-model-based diagnosis approaches can be further classified according to the applied fault model. The method reported in [1] utilizes a transition fault simulator as the fundamental engine. The reported transition fault

candidates are then further refined by removing those faults which cannot be observed due to the circuit timing constraints. However, this method must make an assumption about the fault size. Moreover, the applied timing model is fixed, which may affect the result in the presence of delay variations. The methods proposed in [3–5] apply the path-delay-fault model. In [3], the authors propose an efficient algorithm to identify the sensitized paths based on ZBDD. However, the number of reported candidates are very large. The work reported in [4, 5] describes an approach based on statistical timing information to guide the path-delay-fault diagnosis. In this method, it assumes that the delay probability density function (*PDFs*) of each circuit element (i.e. cell or interconnect) is available. This assumption may not be valid as an accurate statistical model is still not available in practice. Moreover, the proposed method is path-based, which may not be scalable for circuits with a huge path count.

The reasoning-based algorithm is primarily based on the critical-path-tracing (CPT) technique [6–8] used to identify all possible locations where a defect may incur extra delay. It back-traces the sensitized paths from failing primary outputs to extract all possible candidates. In contrast to diagnosis based on a transition-fault-model, which considers only stable signal values in the two cycles, CPT considers all possible transitions including static and dynamic hazards. Therefore, CPT is a more general methodology than the fault-model-based techniques. This implies that the transition-fault-based method might miss a real defect in the presence of static hazards. However, in practice, CPT could report too many candidates. To bound the defect size, the method reported in [8] utilizes the information from fault-free robustly testable paths. Based on such timing information, some of the candidates can be removed to improve the resolution. However, extracting robustly testable paths might not be scalable and the fixed-delay model (without considering delay variations) could lead to inaccurate diagnosis results.

In this paper, we propose a timing-reasoning algorithm to improve the resolution of delay-fault diagnosis. Based on the circuit timing information obtained, we identify *invalid candidates* which cannot maintain consistent failure behaviors. To increase the robustness of the method with respect to timing uncertainty (due to circuit timing variations and/or timing model inaccuracy), we define a metric called *invalid-probability*. By ranking the candidates with respect to the probability of their being invalid, we can improve the first-hit-rate of traditional CPT. To demonstrate the efficiency of the proposed method, we have developed a diagnosis framework which simulates the real diagnosis process. Using this framework, we can compare and evaluate various algorithms.

The rest of the paper is organized as follows. In Section 2, we review the CPT algorithm. In Section 3, we

describe the new delay-fault diagnosis algorithm based on timing reasoning. We give the definition of invalid candidate and propose a simulation-based method to calculate the timing information. In Section 4, we analyze the problem caused by circuit timing uncertainty. We propose a candidate ranking heuristic which can tolerate timing variations caused by such uncertainties to the diagnosis process. In Section 5, we describe the diagnosis framework which is used to simulate the real diagnosis process and evaluate the proposed algorithm. In Section 6, we show some experimental results.

2 Critical-Path-Tracing

Critical-path-tracing (CPT) [6] is a general method for delay fault diagnosis. It utilizes the simulation result of test vectors and traces all possible candidates along sensitized paths from failing primary outputs. It assumes that the failure has been caused by a single defect in the circuit. Unlike transition-fault-model based diagnosis, which considers only the stable signals values in the two cycles under consideration, CPT considers all possible transitions including static and dynamic hazards.

To simulate the transition behaviors of a circuit, CPT utilizes a multi-valued logic system to encode various transitions. The symbols representing different signals are shown in Figure 1. S0 and S1 represent static zero and one, respectively. T0 and T1 represent the transition or dynamic hazard, respectively. P0 and P1 represent static zero and one hazards, respectively.

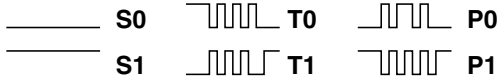


Figure 1: Multi-valued Logic System

Given a logic gate and input symbols, the output symbol is calculated by the multi-valued simulation which can be implemented by table lookup [6]. The multi-valued simulation is a timing-independent process which considers all possible transitions. By applying the CPT algorithm to the multi-valued simulation results, we can extract the candidates along the sensitized paths. The tracing process starts from a failing primary output and tries to extend the primary output's critical state as far as possible toward the primary inputs.

A fanin of a gate is considered critical if the output stable-value and stable-time can be determined by this fanin alone. If the transition at a gate's output arrives late, then its critical fanins are the cause of the output's late arrival. Those critical fanins are then marked as sensitive lines (inputs) that represent the sources from which the delay fault may be propagated. The sensitive lines are the candidates which might be responsible for the failure. Figure 2 (a)(b) shows an example of an AND gate. If the output arrives late, then the dots represent the sensitive lines (inputs) under different input symbols.

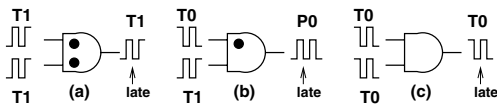


Figure 2: Sensitive Lines

In the case of a gate having no sensitive inputs (Figure 2 (c)), a delay-fault can be propagated through the gate only if the fault affects more than one input simultaneously. In this case, the output arrives late only if these fanins all arrive late. CPT performs the re-convergent loop analysis [7] to identify the source stem backward from these fanins and marks the stem as a sensitive line.

Since the diagnosis process assumes the error source is a single defect, the candidate list for a given test vector is derived from the intersection of the candidates extracted from all failing primary outputs. The final candidate list is the intersection of the candidates extracted from all test vectors.

Figure 3 shows an example. After simulating the test vector, sensitive lines are marked from failing primary output *G*. CPT then reports segments *BE*, *CE*, and *EG* as candidates.

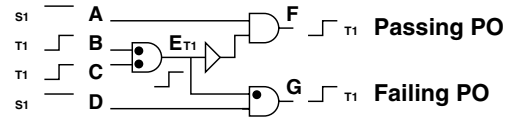


Figure 3: CPT Example

3 CPT with Timing Reasoning

CPT extracts all possible candidates based on the logic conditions. However, since CPT utilizes information only from failing patterns, the number of reported candidates is usually very large. Moreover, since CPT is a timing-independent process, some of the candidates can never be valid if circuit timing information is taken into account.

To utilize the timing information, we propose a timing-reasoning approach which is illustrated in Figure 4. It consists of two phases: the *candidate extraction* phase and the *candidate ranking* phase. The objective of the candidate extraction phase is to efficiently extract possible candidates without using timing information. Based on the test vectors and the failure report, we utilize the traditional critical-path-tracing (CPT) technique to extract possible candidates. In the second phase, we utilize the circuit timing information to further rank the candidate list. A candidate with a higher ranking is more likely to be a valid candidate.

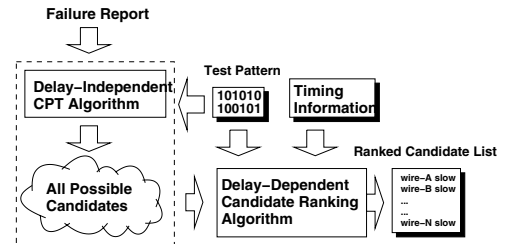


Figure 4: CPT with Timing Reasoning

3.1 Timing Reasoning

By incorporating timing information of both passing and failing primary outputs of each pattern, some of the candidates reported by CPT cannot consistently explain the failing and passing behaviors simultaneously.

Figure 3 shows an example. Traditional CPT reports segments BE , CE , and EG as candidates. Suppose accordingly to the given timing model of the circuit, the delay of segment EF is longer than the delay of segment EG . With this timing knowledge, segment BE and CE must be fault-free. Otherwise, the failing and passing primary output behaviors cannot be consistent with the measured results. Therefore, based on the circuit timing information, segment EG is the only candidate.

In the following section, we define *invalid candidates*. By removing invalid candidates, the diagnosis resolution can be improved.

{Definition} Given a wire f and a test vector v , if there exists no defect-size (extra-delay assignment) on f that can maintain the consistency of passing and failing primary output behaviors, then f is an *invalid candidate*.

Assuming the detection is monotonic with respect to timing defect size (i.e. if a wire f with a timing-defect size δ can be detected by vector v , then v can also detect f with a timing-defect size larger than δ), then the problem of checking the validity of a candidate can be formulated as follows:

{Checking the Validity} A candidate f is invalid if the minimum defect-size (extra-delay) of f , which makes all failing primary outputs fail the test, cannot make all passing primary outputs pass the test.

Please note that this assumption may not be true in the presence of hazards. Therefore, the refinement technique must be able to tolerate this uncertainty. Our candidate ranking heuristic addresses this problem.

The minimum defect size is determined by the *failing sensitized paths*, which are the paths going through f and ending at a failing output. The delay of every failing path plus the minimum defect size must be longer than the clock period. On the other hand, to make all the passing outputs pass the test, the minimum defect size plus the delay of any *passing sensitized path* must be shorter than the clock period. The passing paths are the paths which go through f and end at a passing output. As a result, if any delay of a passing sensitized path is longer than the delay of any failing path, then it is impossible to find a minimum defect size which can maintain the consistency. Therefore, f is invalid.

The following section gives the details of deriving required timing information.

3.2 Minimum Defect Size

The minimum defect size of a wire f , under test vector v , which makes a failing primary output, po_i^{fail} , arrives later than the clock clk is determined by the delay of the longest sensitized path going through f to po_i^{fail} , denoted as $dlf(v, f, po_i^{fail})$.

The minimum defect size of a wire f , under test vector v , which makes all failing primary outputs arrive later than the clock clk is determined by the smallest value of $dlf(v, f, po_i^{fail})$ among all failing primary outputs, denoted as $dlf(v, f)$.

As shown in Figure 5, $dlf(v, f, po_i^{fail})$ can be calculated as the summation of the delay of the activation path and the delay of the failing propagation path. The delay of

the activation path is the arrival time of candidate f under vector v , which is denoted as $at(v, f)$. The delay of the failing propagation path under vector v , which is the delay of sensitized segment from f to po_i^{fail} , is denoted as $df(v, f, po_i^{fail})$.

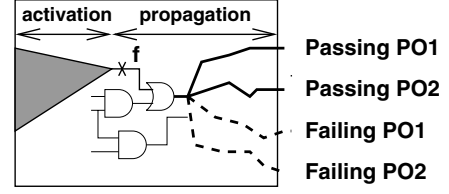


Figure 5: Deriving Timing Information

For each candidate f , we calculate the $at(v, f)$ as the latest arrival time by timing simulation. However, as the extra delay introduced by the defect is unknown, we cannot calculate $df(v, f, po_i^{fail})$ by timing simulation. Under test vector v , the possible failing propagation paths from f to failing primary outputs are those segments starting from f , along with the sensitive lines extracted by CPT, and ending at a failing primary output. Note that without accurate timing information, we cannot tell exactly which failing propagation paths are sensitized. Therefore, we make a conservative assumption for the calculated minimum defect size. By assuming that the longest failing propagation path is sensitized by vector v , the derived $dlf(v, f, po_i^{fail})$ is a lower bound of minimum defect size of f under v . The $dlf(v, f, po_i^{fail})$ can be calculated by updating the longest delay from f to po_i^{fail} along the sensitive lines. The $dlf(v, f)$ is the minimum value of $dlf(v, f, po_i^{fail})$ among all failing primary outputs. These result in the following equations:

$$dlf(v, f, po_i^{fail}) = at(v, f) + df(v, f, po_i^{fail}) \quad (1)$$

$$dlf(v, f) = \min(dlf(v, f, po_i^{fail})) \forall po_i^{fail} \quad (2)$$

3.3 Delays of Passing Paths

The delay of the longest sensitized path which goes through wire f and pass test vector v can be calculated in a similar fashion. As shown in Figure 5, $dlp(v, f)$ can be calculated as the summation of the delay of the activation path and the delay of the passing propagation path ($dp(v, f)$). Again, we calculate the passing delay conservatively: we utilize the *semi-robust sensitization criterion* to calculate the delay of the longest passing propagation path.

To calculate the $dp(v, f)$, we update the longest delay from f to passing primary outputs along the *semi-robust-sensitive lines*. The paths along the semi-robust-sensitive lines are the possibly sensitized paths. Given a gate G and the simulated values, the semi-robust-sensitive lines can be identified as follows:

- If all inputs of gate G have a non-controlling final value, then every input which is non-stable is a semi-robust-sensitive-line.
- If only one input in has a non-stable value with a controlling final value, and other inputs have a stable value or a non-stable value with a non-controlling final value, then in is a semi-robust-sensitive-line.

Since one and only one of the semi-robust-sensitive lines actually dominates the gate delay, then $dp(v, f)$, which is calculated by the semi-robust criterion, is a lower bound value of the delay of the propagation path from f to any passing primary output. The $dlp(v, f)$ then can be calculated by the following equation for each v :

$$dlp(v, f) = at(v, f) + dp(v, f) \quad (3)$$

Please note that because of the strict logic constraints, we may not be able to derive the $dp(v, f)$ for every f . If this is the case, we set the $dlp(v, f)$ is equal to zero.

Given a candidate f , if there exist vectors v_i and v_j so that $dlp(v_i, f) > dl f(v_j, f)$, then f is an invalid candidate.

4 Timing Uncertainties and Candidate Ranking

Based on the passing and failing timing information, we can refine the list of candidates reported by CPT. However, there are several sources of uncertainty during the process of calculating the timing information. Therefore, the refinement technique must be able to tolerate these uncertainties. There are three major sources of uncertainty which are: (1) Delay variation, (2) Timing simulation mismatch, and (3) Sampling invalidation.

Due to the process variation, the delay values of circuit components vary from chip to chip. Therefore, the calculated delay values may not be the same for different circuit instances. To handle this variation, instead of using one fixed delay value, we utilize the min-max bounded delay model to calculate the timing information in the timing reasoning process. This delay model has been widely used in industrial standard-cell libraries.

The second source of uncertainty is due to the limitation of timing simulation algorithms. In order to efficiently utilize timing simulation to calculate the arrival time, timing simulation calculates the latest stable time. However, the real-chip timing behavior is better explained by waveform propagation, and the latest stable time might be too conservative. Figure 6 shows an example of the difference between latest-stable-time simulation and waveform-based simulation.

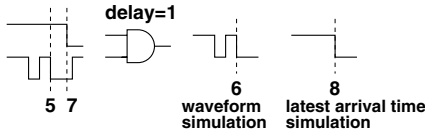


Figure 6: Different Timing Simulation Algorithms

The third source of uncertainty, the sampling invalidation, is illustrated in Figure 7 (a). In the presence of static and dynamic hazards, using only one sampling clock may incorrectly classify some failing primary outputs as passing primary outputs. We call this phenomenon the *sampling invalidation*.

This mis-classification may result in incorrect diagnosis result. To solve this problem, instead of sampling under one clock period, we could perform multi-sampling with various clock frequency. As shown in Figure 7 (b), we slightly increase the sampling clock period by Δ and mark

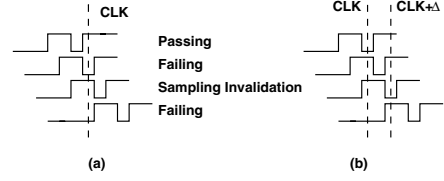


Figure 7: Sampling Invalidation

all the failing primary outputs. We keep sampling until no failing primary output can be observed. Then, we consider the set of failing primary outputs as the union of all failing outputs under different sampling clocks. The value Δ could be determined by the resolution of the tester.

4.1 Candidate Ranking

Due to the above sources of uncertainty, the calculated timing information may mislead the timing reasoning. To increase the tolerance to these uncertainties, instead of removing those candidates which cannot maintain the consistency, we propose a candidate ranking heuristic.

For each vector v , we calculate the $dl f(v, f)$ and $dlp(v, f)$ under minimum and maximum delay corners. $dl f_{min}(v, f)$ and $dlp_{min}(v, f)$ assume all pin-to-pin delays are at minimum delay corner. $dl f_{max}(v, f)$ and $dlp_{max}(v, f)$ assume all pin-to-pin delays are at maximum delay corner.

We assume that $dl f(v, f)$ and $dlp(v, f)$ are two independent random variables with a normal distribution. The delay values calculated under minimum and maximum delay corners assumed to be the values of 3σ bound. The mean μ and the variance σ of these normal random variables can be calculated.

The probability of $dlp(v_i, f) > dl f(v_j, f)$, which indicates the probability that the candidate might be invalid under v_i and v_j , can be calculated by Equation 4. We denote this probability as the *invalid-probability*. Φ denotes the cumulative distribution function of a standard normal random variable. The higher probability implies that the candidate f is more likely to be invalid.

$$Pr(dlp(v_i, f) > dl f(v_j, f)) = \Phi\left(\frac{\mu_{dlp} - \mu_{dlf}}{\sqrt{\sigma_{dlp}^2 + \sigma_{dlf}^2}}\right) \quad (4)$$

Candidates then are ranked by their invalid-probability. The ranking procedure for a candidate f consists of three stages: (1) Given a vector v , if both $dlp(v, f)$ and $dl f(v, f)$ are available ($dlp(v, f) \neq 0$), then we rank the candidates based on the maximum value of the invalid-probabilities among all v 's. In this case, the failing and passing paths share the same activation path. Therefore, the uncertainty caused by the timing simulation mismatch can be eliminated. (2) For candidates with a tie based on the above criterion, we use the following metric. We select the k largest $dlp_{max}(v, f)$ and $dlp_{min}(v, f)$ among all v 's and the k smallest $dl f_{max}(v, f)$ and $dl f_{min}(v, f)$ among all v 's. We obtain k random variable pairs based on these k pairs of values and calculate the invalid-probability based on these k pairs. We then utilize the average value of these derived invalid-probabilities as the second ranking metric. Note that using the average would increase the tolerance to timing uncertainties. (3) If there is still a tie among candidates, we then rank the candidates based on their logic level. Based on the proposed heuristic, every candidate has a unique rank.

5 Timing Diagnosis Framework

To demonstrate the diagnosis process and evaluate the quality of different diagnosis algorithms, we develop a timing diagnosis framework which is shown in Figure 8. Due to the absence of real test chips, we develop a statistical timing simulator - DSIM-v40, to simulate the real timing behavior of a test chip [11]. The simulator first generates a circuit instance whose timing behavior is based on a given delay library and randomly injected delay defects. Given a test vector, the simulator then performs timing simulation and produces the corresponding output waveforms. According to the sampling clock, the failure report indicates which primary output samples the wrong value (i.e. failing primary output). Different diagnosis algorithms then can be applied to extract candidates based on the failure report and the test vectors. The quality of various diagnosis algorithms can then be evaluated.

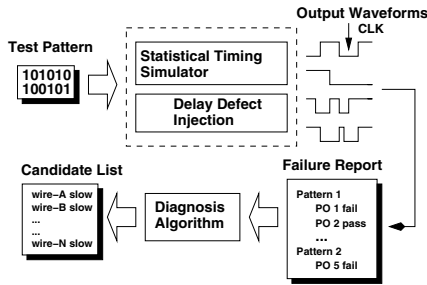


Figure 8: Timing Diagnosis Framework

5.1 Statistical Timing Simulator

Due to the process variation, the delay configuration of a circuit differs from chip to chip. In the statistical timing simulator, the delay of a cell or an interconnect is modeled as a random variable with a known probability density function (*PDF*) which is extracted from Monte-Carlo-based SPICE simulation [9]. To mimic test chips, based on the given statistical timing models, we generate a number of *sample chip instances*. Each sample instance has a different but fixed delay configuration. Different types of delay defects with specified locations and sizes can be injected into a circuit for simulation. Given a set of test vectors, the simulator performs timing simulation for each sample instance. Unlike the previous simulator [10, 11], which calculates the latest arrival time, DSIM-v40 performs the waveform simulation that can provide more precise timing information. Please refer to [11] for the implementation details of the statistical timing simulator.

5.2 Sampling Invalidation

Based on the result of waveform-based simulation, sampling invalidation (shown in Figure 7) can be simulated by the proposed framework. Sampling invalidation occurs often for those circuits with a huge number of reconvergent paths. Figure 9 shows the example of circuit c6288. The x-axis shows the test vector index and the y-axis shows the number of failing primary outputs. The first curve shows the number of primary outputs which arrive later than the clock. The second curve shows the number of failing primary output under multi-sampling. And the third curve shows the number of failing primary outputs by using only one sampling clock.

In this example, we assume the resolution of the tester is 100ps. As it can be seen, using only one sampling clock

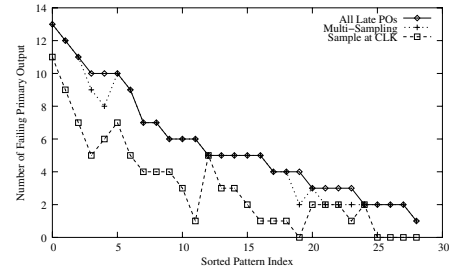


Figure 9: Sampling Invalidation for c6288

may have a non-trivial number of mis-classified primary outputs. However, even with multi-sampling, there is still some mis-classification. Therefore, the proposed diagnosis algorithm must have the capability to tolerate such mis-classification. Our candidate ranking heuristic addresses this problem.

6 Experimental Results

We conducted experiments to evaluate the proposed diagnosis algorithm using the diagnosis framework. To construct the statistical delay library for DSIM-v40, we apply Monte-Carlo-based SPICE (ELDO) simulation [9] to extract each cell's pin-to-pin delay *PDF*'s using a 0.25μm, 2.5V CMOS technology, which characterizes about 15~25% (in the worst case) inter-die variations for pin-to-pin delays. We then extract the min-max delay value for each cell which is applied to the timing-reasoning procedure.

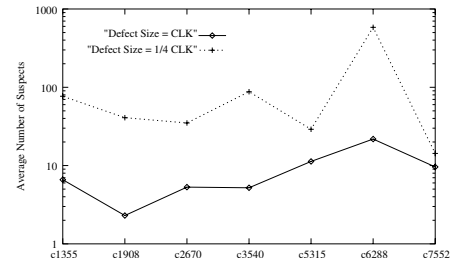


Figure 10: Defect size v.s. Number of candidates by CPT

We first demonstrate the relation between defect size and diagnosis resolution. Figure 10 shows the diagnosis result based on traditional CPT with respect to different injected defect sizes. We generate 100 defective chip instances with random defect injection and apply transition-fault test vectors. The x-axis shows the circuit name, and the y-axis shows the average number of candidates reported by CPT. The first curve shows the results for the experiment where the size of the injected defect is equal to the circuit clock period. The second curve shows the results for the experiment where the defect size is equal to a quarter of the circuit clock period. Not surprisingly, the smaller the defect size, the larger the number of reported candidates. That is, the diagnosis resolution is lower for smaller delay defects.

To demonstrate the effectiveness of the proposed algorithm, we randomly generate chip instances by injecting a single delay defect. Based on the transition-fault test vectors, we compare the diagnosis result with that of traditional CPT algorithm. Figure 11 shows an experimental result for the combinational part of s13207. We injected

Table 1: Experimental Result

CKT	CLK(ps)	# Ptn	Defect Size	Avg # Candidate	Avg First-Hit-Rate	Avg CPU(sec)
c2670	32	277	16 / 32	35.0 / 5.3	5.9 / 1.4	0.33
c3540	49	418	24 / 49	9.2 / 5.2	1.9 / 2.1	0.66
c5315	48	275	12 / 48	29.1 / 11.3	5.0 / 3.2	1.14
c6288	140	97	35 / 140	583.4 / 21.8	124.4 / 8.0	0.53
c7552	42	484	10 / 42	14.3 / 9.6	2.8 / 5.5	3.52
s9234comb	75	2026	9 / 75	34.4 / 8.3	13.2 / 2.6	15.78
s13207comb	75	1685	9 / 75	60.5 / 9.6	15.8 / 5.0	63.34
s15850comb	93	1191	24 / 93	141.8 / 21.4	40.6 / 5.5	361.45
s35932comb	21	142	2.5 / 21	23.2 / 3.5	14.6 / 2.1	7.65
s38417comb	31	2369	8 / 31	27.7 / 1.4	6.2 / 2.3	132.7
s38584comb	68	1685	8 / 68	70.2 / 7.2	28.9 / 4.5	229.6

a single delay defect at a random location with a defect size of 1/8 of the clock period. The x-axis shows the index of the defective chip instance and the y-axis shows the number of reported candidates. The first curve shows the results of CPT. The second curve shows the rank of the real candidate after applying the ranking heuristic. We set the parameter k to 3 for the ranking process in the experiment. For example, CPT reports 150 candidates for the first defective instance. But after applying the ranking heuristic, we can locate the faulty wire in the candidate list at the 30-th place. Therefore, the first-hit-rate is improved.

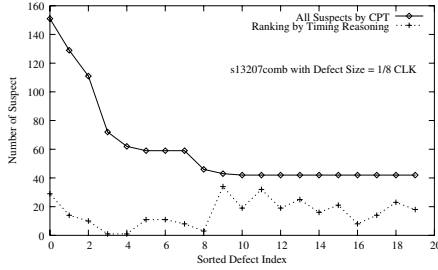


Figure 11: Experimental Result - s13207comb

Table 1 shows the experimental results for some public benchmark circuits. We generate 20 defective chip instances for each circuit and apply both the traditional CPT algorithm and the proposed ranking heuristic. We set the parameter k to 3 in the experiment for all cases. The first column shows the circuit name. For sequential benchmarks, we only use the combinational part. The second column shows the clock period for the corresponding circuit. The third column shows the number of transition-fault test vectors we applied. The fourth column shows the injected defect size. We injected two different sizes of defect: one which is equal to the clock period and another which is much smaller than the clock period. As indicated in Figure 10, small defects have a higher probability of resulting in a low diagnosis resolution. The fifth column shows the average number of candidates which are reported by traditional CPT. Traditional CPT can produce high resolution diagnosis results for a large defect size, whereas the diagnosis resolution for small defects is low. The sixth column shows the average first-hit-rate if we follow the rank produced by the procedure. The final column shows the total CPU time for the diagnosis process.

Overall, the results indicate that the proposed ranking heuristic can efficiently improve the first-hit-rate for small defect sizes. For large defect sizes, the CPT alone already produce high-resolution results.

7 Conclusions

In this paper, we propose a timing-reasoning algorithm to improve the resolution of delay-fault diagnosis. Based on the circuit timing information, we identify invalid candidates which are candidates that cannot consistently explain the failure behaviors. By eliminating those invalid candidates, the diagnosis resolution can be improved. We further analyze the problem resulting from the circuit timing uncertainties caused by delay variations and inaccurate simulation models. We calculate the invalid-probability for each candidate and propose a candidate ranking heuristic to increase the robustness of the diagnosis process with respect to those uncertainties. By ranking the candidates based on their invalid-probability, we can improve the candidate first-hit-rate of traditional CPT.

References

- [1] Z. Wang, *et al*, *Delay Fault Diagnosis Using Timing Information*. Proc. ISQED, 2004.
- [2] H.B. Wang, *et al*, *Gate-Delay Fault Diagnosis Using the Inject-And-Evaluate Paradigm*. Proc. DFT, 2002.
- [3] S. Padmanaban, *et al*, *Non-Enumerative Path Delay Fault Diagnosis*. Proc. Design, Automation and Test in Europe (DATE), 2003.
- [4] A. Krstic, *et al*, *Delay Defect Diagnosis Based Upon Statistical Timing Models - The First Step*. Proc. Design, Automation and Test in Europe (DATE), 2003.
- [5] A. Krstic, *et al*, *Enhancing Diagnosis Resolution For Delay Defects Based Upon Statistical Timing and Statistical Fault Models*. Proc. IEEE/ACM Design Automation Conf. (DAC), 2003.
- [6] P. Girard, *et al*, *Delay-Fault Diagnosis by Critical-Path Tracking*. IEEE Design & Test of Computers, 1992.
- [7] P. Girard, *et al*, *A Reconvergent Fanout Analysis for the CPT Algorithm Used in Delay-Fault Diagnosis*. Proc. IEEE European Test Conference, 1993.
- [8] J.G. Dastidar, *et al*, *Adaptive Techniques for Improving Delay Fault Diagnosis*. Proc. IEEE VLSI Test Symp. (VTS), 1999.
- [9] *Eldo v4.4.x User's Manual*, 1996.
- [10] K. Yang, *et al*, *TranGen: A SAT-Based ATPG for Path-Oriented Transition Fault*. Proc. ASP DAC, 2004.
- [11] K. Yang, *et al*, *On Statistical Correlation Based Path Selection for Timing Validation*. IEEE VLSI-TSA, 2005.