Computer-Aided Fault to Defect Mapping (CAFDM) for Defect Diagnosis

Zoran Stanojevic

Dept. of Electrical Engineering
Texas A&M University
College Station TX 77843-3124
Tel: (979) 862-6610
Fax: (979) 847-8578
Email: zoran@cs.tamu.edu

Hari Balachandran

Texas Instruments, Inc.
Mixed Signal Products
Dallas TX
Tel: (214) 480-7602
Fax: (214) 480-7676
Email: harib@ti.com

D. M. H. Walker

Dept. of Computer Science
Texas A&M University
College Station TX 77843-3112
Tel: (979) 862-4387
Fax: (979) 847-8578
Email: walker@cs.tamu.edu

Fred Lakhani

International Sematech
Yield Management Tools
2706 Montopolis Drive
Austin TX 78741
Tel: (512) 356-7011
Fax: (512) 356-7640
Email: fred.lakhani@sematech.org

Sri Jandhyala

Texas Instruments, Inc.
Mixed Signal Products
Dallas TX
Tel: (214) 480-7598
Fax: (214) 480-7676
Email: srij@ti.com

Jayashree Saxena

Texas Instruments, Inc.
P.O. Box 660199 MS 8645
Dallas TX
Tel: (214) 480-1646
Fax: (214) 480-2356
Email: j-saxena@ti.com

Kenneth M. Butler

Texas Instruments, Inc.
P.O. Box 660199 MS 8645
Dallas TX
Tel: (214) 480-1431
Fax: (214) 480-2356
Email: kenb@ti.com

Abstract

Defect diagnosis in random logic is currently done using the stuck-at fault model, while most defects seen in manufacturing result in bridging faults. In this work we use physical design and test failure information combined with bridging and stuck-at fault models to localize defects in random logic. We term this approach computer-aided fault to defect mapping (CAFDM). We build on top of the existing mature stuck-at diagnosis infrastructure. The performance of the CAFDM software was tested by injecting bridging faults into samples of a streaming audio controller chip and comparing the predicted defect locations and layers with the actual values. The correct defect location and layer was predicted in all 9 samples for which scan-based diagnosis could be performed. The experiment was repeated on production samples that failed scan test, with promising results.

I. Introduction

Fault localization or fault isolation is the process of identifying a region within an integrated circuit that contains a circuit fault, such as a short or open circuit. This region must be small enough that the defect causing the fault can be found and analyzed. This is very important for quickly debugging new products, ramping yields, identifying test and reliability problems in customer returns, and resolving quality assurance (QA) part failures. Advancing integrated circuit technology has resulted in more interconnect layers, smaller geometries, greater chip complexity, and flip-chip packaging. All of these greatly increase the complexity of fault isolation. Often a direct view of the defect from the front or backside of the chip is not available. This makes it increasingly difficult to locate the defect using defect localization methods that detect light or heat given off by the defect. Fault isolation has become the most time-consuming part of defect diagnosis, and often a diagnosis cannot be performed since the fault cannot be localized. As a result, fault isolation is listed as a difficult challenge in the International Technology Roadmap for Semiconductors, with its complexity projected to grow by 142 times in the next 15 years [1].

The alternative to defect localization techniques is to use electrical tests to isolate the circuit fault, with the assumption that the defect is closely associated with the fault. In memory arrays this localization process is straightforward since there is a direct mapping between the bit fail map and the possible defect locations. In logic circuits such a straightforward mapping between electrical failures and locations is not available.

Current state-of-the-art practice for localizing faults in logic circuits is to use a stuck-at fault diagnosis approach, which can be applied to circuits that are full scan or mostly scan. In this approach, a set of passing and failing vector outputs is captured, and fed into the diagnosis system. Diagnosis can be done with a variety of techniques such as Boolean difference [2] and critical path tracing [3], to identify a list of suspect nets where stuck-at faults best explain the observed failure patterns. The nets are ranked based on how many faulty output patterns they explain. The suspect list includes logically equivalent nets. For example, a stuck-at-0 fault on an AND gate output is equivalent to a stuck-at-0 fault on any of the AND gate inputs. Stuck-at diagnosis tools are commercially available, typically as part of an automatic test pattern generation (ATPG) system. Two examples are Mentor Graphics FastScan^{TM} and IBM TestBench^{TM}. TestBench also supports the more general pattern fault model.

The list of suspect nets from the diagnosis system can be visualized in the chip layout database using a tool such as Knights Technology LogicMap^{TM}, which integrates the Knights CADNAV^{TM} layout navigation system with FastScan [4][5]. The top few suspect nets are then examined with a scanning electron microscope (SEM) in order to locate the fault within the net. Voltage contrast is often used to locate open circuits within these nets. This

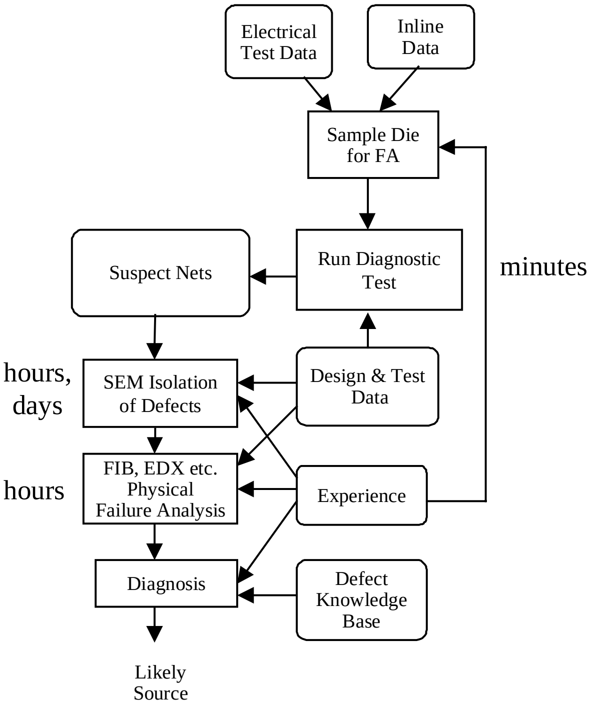


Figure 1. Current Defect Sourcing Approach

current defect sourcing approach is shown in Figure 1. As shown, most of the time is spent in the SEM search.

The drawback of using a stuck-at diagnosis approach is that most circuit faults are bridging faults, and the stuck-at fault model is not a very accurate model for them [6][7]. It is well known that the more accurately the fault model matches actual fault behavior, the better the diagnosis. The result of this limitation is that a significant fraction of the time, the real faulty nets do not appear on the list of suspects, or the faulty net is far down the ranked list of suspects, and so is never examined due to resource limitations.

A second problem with the stuck-at approach is that it does not provide any ranking in terms of chip locations or process layers. Even if only one suspect net is reported, if it is a very large net, it may not be feasible to locate the fault with the SEM. This is particularly true if the net traverses primarily interior interconnect layers, and so is not directly visible from the top or backside of the chip. The increased number of interconnect layers in deep submicron circuits means that most signal routing is on an interior layer. Often such cases can only be diagnosed if, during fabrication, defect inspection detected a defect at the same location as one of the suspect nets. However inspection information is only available for a small fraction of chips, and is rarely available for customer returns.

Overall, the result of these limitations is that the fault cannot be located in about 30% of all chips. This is not such a problem for low yield analysis, where more faulty chips are available, but is of serious concern when diagnosing quality assurance (QA) part failures or customer returns. In these cases it is a high priority to quickly determine the root cause of the failure, and inability to do so can put a large amount of product at risk.

Table 1. Project Metrics

Description	Baseline	Goal
Failed Diagnosis	30%	<15%
Physical Layer Ranking	No	Yes
Physical Location Ranking	No	Yes
Small Suspect List (<10)	80-95%	90-99%
Average Localization Time	Days	Hours

Rather than replace the mature stuck-at diagnosis infrastructure, the goal of this project was to improve the performance of existing tools. We term our approach *computer-aided fault to defect mapping* (CAFDM). This approach uses physical design information to identify locations and layers within the circuit where short circuit faults can occur. This information is used to filter and add location and layer information to the suspect net list provided by stuck-at diagnosis, as shown in Figure 2, reducing the SEM search time. The project metrics are shown in Table 1. The location and layer ranking are required to reduce SEM search time. The small suspect list is required since resource limitations rarely permit looking at more than the top ten suspects. These metrics were determined by polling semiconductor manufacturers on what they would consider a substantial improvement over current practice. We do not have a misleading diagnosis metric, but instead subsume it into the failure rate and average localization time. One of the goals of this project was to determine what knowledge sources are most useful in reducing the SEM search area, which reduces the SEM search time.

The remainder of the paper describes the CAFDM approach to defect diagnosis. Section 2 describes previous work on defect diagnosis. Section 3 presents the CAFDM approach. Section 4 describes experimental results and a comparison with previous work, and Section 5 describes conclusions and directions for future work.

II. Previous Work

Several methods have been suggested to improve diagnosis of bridging faults using the stuck-at fault model. A vector that detects a bridging fault also detects an associated stuck-at fault [8]. Composite signatures can be constructed for each potential bridging fault by the union of the four associated single stuck-at signatures. However this technique results in an unacceptable rate of misleading diagnoses. An improvement is to use only realistic faults,

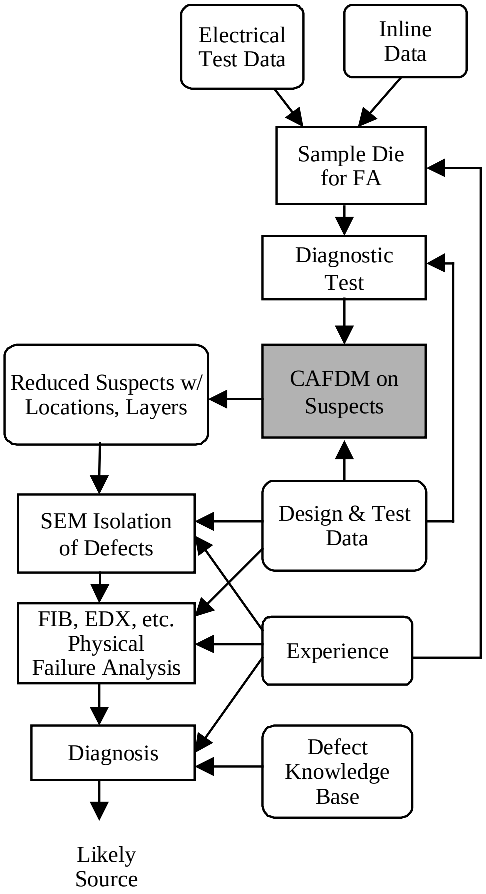


Figure 2. CAFDM modification of existing fault diagnosis approach

and better match restriction [9], which was demonstrated experimentally [10]. One drawback of this improved approach is the need to build a large fault dictionary.

There has been prior work on fault diagnosis using more accurate fault models, including wired-OR and wired-AND bridging fault models [11] and voting models [12]. These models improve the accuracy of the suspect net list, but suffer from two problems. First, they usually require a completely different diagnosis infrastructure, replacing the mature stuck-at test and diagnosis software. The use of different ATPG and diagnostic engines may require two different design descriptions or cell libraries, leading to substantial engineering effort to describe the design a second time, and potential error introduction. Second, such models do not provide physical location or layer information. The latter can mean that all possible bridges to the stuck-at suspect nets must be considered, which is infeasible in a large circuit. A related problem is that some approaches require building a large fault dictionary in advance, which is very costly for large circuits.

The best bridging fault diagnosis results can be achieved by modeling the bridging fault behavior at the circuit level, and using layout information to identify adjacent nets that could be involved in a bridge [13][14]. The drawback is the need for a bridging fault simulator [15]. IDDQ tests alone can achieve small suspect lists [16]-[19]. The drawback is the need for large numbers of IDDQ tests. Since these are not normally used in production test, a separate diagnostic test set is needed.

III. Computer-Aided Fault to Defect Mapping

The key assumption in CAFDM is that by using physical design information and a combination of bridging and stuck-at faults, we can accurately model the behavior of most faults, and achieve sufficient location and layer resolution to substantially reduce SEM search time. We restrict our work to faults that cause functional failures, and so ignore faults causing parametric failures. Thus we set a goal of reducing the number of failed diagnoses in half from current practice. We also make the assumption that if the fault behavior is not modeled, there will be no good suspect net matches, and so the average localization time will not significantly increase due to long fruitless searches due to misleading diagnoses. These assumptions will be tested experimentally.

The initial CAFDM implementation is shown in Figure 3. A fault extractor identifies the critical areas where short circuits could occur on the suspect nets, including their locations and layers. We initially used the DEFAM (Defect to Fault Mapper) tool [20] due to its familiarity. We only extracted short circuits since open circuits tend to behave like stuck-at faults, and so the FastScan stuck-at diagnosis algorithm behaves well on them. In addition, an open can occur anywhere on a suspect net, so the search area is the entire net, and this can already be visualized with existing tools like LogicMap. In this work only intralayer bridges were considered, since they are much more probable than interlayer bridges.

The DEFAM system works well on very regular layout designs. However in an ASIC design style, there is little regularity in the logic sections of a chip layout. DEFAM cannot handle this design style in reasonable memory or time. It is these sections of the chip that are the subject of our work, since the memory arrays can be diagnosed by bitmapping. To solve this problem, we have developed a new fault extractor, FedEx, which uses a scanline approach to circuit and fault extraction. It generates a list of two-node bridging fault sites that are possible for a maximum defect size. These include bridging pairs that would actually have to be three-way or more bridges due to intervening lines. For example, given three adjacent parallel lines A, B, and C, the tool will report bridges A-B, B-C, and A-C, even though the latter should be A-B-C.

Each bridging fault site includes the layer, bounding box of the critical area, and weighted critical area. There can be multiple sites between a pair of nets. The critical

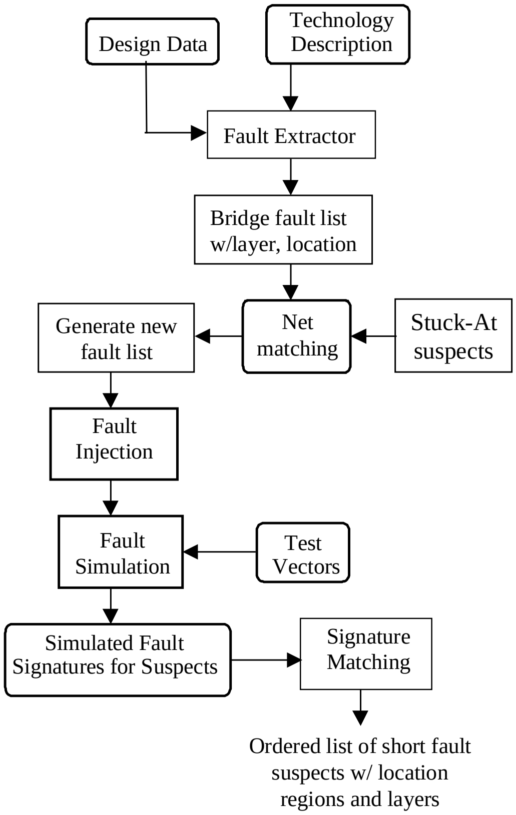


Figure 3. CAFDM Implementation

area weighting is done using the $1/x^3$ defect size distribution, so that when divided by the chip area, the weighted area is the relative probability of failure.

For test generation or diagnosis it is more important to identify all possible realistic bridging faults than to obtain accurate critical areas. Thus FedEx is optimized for speed while obtaining critical areas accurate to within 20%. Its performance is such that it can extract the faults of our target design in one hour of CPU time on an engineering workstation. This is more than an order of magnitude faster than fault extraction with an accurate critical area computation [21].

The potential bridging fault list can be quite large. We have found that there are about 5 fault sites per transistor, and about 100 per labeled net. For a design with 50M transistors of random logic, there are about 250M possible bridging fault sites. Simulating all of these is infeasible, which is why we filter the list using the stuck-at suspects.

For each chip being diagnosed, a set of passing and failing results is captured from the tester, and fed into the FastScan diagnosis system. Stuck-at diagnosis takes less than one minute on million-transistor designs on a typical workstation. Given the mostly-scan nature of the designs, this time grows sublinearly for larger designs, and is not a bottleneck in the diagnosis procedure.

The stuck-at nets from FastScan include logically equivalent nets and are ranked based on how many faulty patterns they explain. These nets (including the equivalent nets) are then matched against the bridging fault list to identify all the potential bridging faults associated with a given suspect net. This matching first requires a translation between the net name specified in the Verilog netlist and that specified in the layout, using interconnect verification information stored in the netlist. The design methodology must provide this mapping, such as via a layout-versus-schematic step. The conversion from stuck-at suspects to Verilog bridge sites is done using a combination of C programs and Perl scripts, and takes several minutes per faulty chip, due to the size of the bridging fault list.

The list of potential bridging faults and stuck-at faults is injected into the netlist, one at a time and fault simulated with a variety of fault models. The results are ranked based on how well they match against measured results. The procedure is explained below.

Each fault is injected via a Perl script by making a copy of the symbol definition in which a fault will be injected, and using that new definition at the symbol instance where the fault occurs. Thus if a fault occurs within an AND gate, a new faulty AND gate is created and used at the fault location.

For each bridging fault a Wired-AND, Wired-OR and Wired model are used, since any one of them may best describe the fault behavior. The Wired model propagates an X condition on the bridged nodes if they are driven to opposite values. Each faulty circuit is simulated with FastScan using the test vectors, and the resulting outputs captured. One issue with this approach is that the FastScan startup overhead can be a significant part of the total simulation time.

The Hamming distance (the number of bit positions in all vectors that differ) between the simulated faulty circuit output and the tester output is computed, and sorted in increasing order of distance. For an X output (from a Wired fault) matched against a tester value of 0 or 1, a distance of 0.5 is used, since the X could resolve to a 0 or 1. The result is a ranked list of bridging fault suspects, along with layer and location information for each fault. For the multiple bridging fault models, the smallest Hamming distance and bridging fault model is recorded.

We chose a Hamming distance match rather than the number of matching vectors (as used in FastScan) for two reasons. First, the Hamming distance metric has higher resolution than just a vector-by-vector match. In the latter approach, two injected faults could have the same number of matching vectors, but one could have many more incorrect bit positions in the unmatched vectors. Intuitively, the vector with the fewer incorrect bit positions is “closer” to the actual fault behavior. Second, the Hamming distance can better deal with the X values resulting from the Wired model. The X values cannot be treated as Don't Care values when performing a match, since this would suggest that a vector with all X values would be good match for any measured vector, which is clearly wrong. On the other hand, an X is "closer" than a 0 to a 1, since there is some probability that it resolves as a 1, and should be ranked accordingly.

For faults with the same Hamming distance, further ranking is done using the weighted critical area, with faults with larger critical area being ranked first, since they are more likely to occur. Rather than using equal Hamming distance, it may be more appropriate to use a weighted combination of metrics. This deals with the case where one fault has slightly worse Hamming distance, but much larger critical area, and so is likely to be a better candidate for diagnosis. Rather than modify the ranking, all this information is provided to the user and left to their judgement. Additional ranking could be done using defect inspection data, expected fault localization time, layer, or other database parameters, but was not done in this study. The ranked list of suspects is then used to perform the SEM search.

One project metric is to generate fewer than ten suspects almost all the time. This requires discarding matches that are worse than some threshold (in terms of Hamming distance or a weighted combination of metrics). For our prototype implementation we did not set a predefined threshold, but simply ranked suspects based on Hamming distance. The user could then discard suspects whose Hamming distance is too large, or when a large jump occurs from one suspect to the next. This issue will be revisited in the experimental results.

One concern about the stuck-at diagnosis is that the nets involved in a bridging fault may not be on the suspect list, or so far down the list that the stuck-at diagnostic tool discards them. This situation is identified when all of the original bridging and stuck-at fault suspects have a Hamming distance that exceeds a user-defined threshold. The reason for the user-defined threshold is that the "normal" Hamming distance depends on how well the modeled faults match faulty circuit behavior, and this can vary with technology and circuit design styles. Note that the large Hamming distance could also mean that the fault behavior does not match the modeled faults, rather than that the faulty net is not on the stuck-at suspect list.

To handle the case of the stuck-at diagnosis missing the defective net, we implemented two approaches. First, we considered bridging faults that are N bridges away from the stuck-at suspect nets. In other words, if net A is a stuck-at suspect, and one potential bridging fault is between nets A and B, then if a potential bridge exists between nets B and C, it too is added to the list, and so on. In this work we considered up to three bridges away from the stuck-at suspects. The general idea is that there is likely to be some spatial correlation between reported suspect nets and actual faulty nets, since for a single fault, the location must lie within the intersection of all the logic fan-in cones of all the faulty outputs for all faulty vectors.

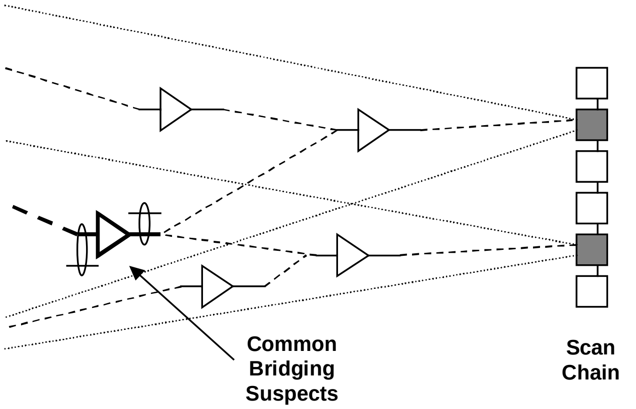


Figure 4. Backconing to identify possible bridging faults.

If the neighboring bridges do not explain the fault (in terms of the Hamming distance threshold), then we use a backconing procedure. For each faulty output, all the nets that are in the logic cone that feeds the output are identified with a Perl script. All potential bridging faults to the nets in each cone are identified. All of these bridging fault lists are intersected to identify potential bridges in which one side or the other has a structural path to all faulty outputs. This situation is shown in Figure 4. The fault list could include bridges between logic cones that do not share any nets. The resulting fault list is used as input to fault simulation. The key assumption of this approach is that one fault will explain the faulty behavior. The advantage of this approach is that it fits into the existing fault simulation framework, and the fault list can be generated via a Perl script. The disadvantage is that the potential fault list is typically much larger than obtained using critical path tracing. It is also about 5-10 times larger than suspect bridge lists obtained using stuck-at suspects.

IV. Experimental Results

A case study using the CAFDM software was carried at out at Texas Instruments (TI), Mixed Signal Products division, Dallas TX. The target chip for diagnosis was a streaming audio controller. It is a 1M-transistor, mixed-signal design, implemented in a 250-nm technology. This design has 1.7M drawn rectangles, and is about half memory and half logic, with a small amount of analog circuitry. The die size is 3.75 mm by 3.75 mm. The digital logic uses a full scan design, which permits FastScan diagnosis. There are about 27,000 labeled nets subject to diagnosis.

Using a maximum defect size of 2.25 microns, 2.8M bridging fault sites were extracted from the layout with memory arrays removed. Of these there are 1.6M unique bridging pairs for simulation. The average critical area for each fault site is 5 μm^2. The 9.5 mm^2 total area for all sites is two-thirds the total chip area. The fault site area tends to be larger on the upper metal layers due to the long parallel runs that occur there. The largest area is 0.64 mm^2, which occurs on metal4. This large area is caused by the fact that the area is computed as the bounding box containing all critical areas in that region between the net pair at the maximum defect size. Thus two adjacent L-shaped lines will result in an area that is much larger than the actual critical area. The weighted critical area is computed using the actual critical area shapes. More complex critical area shapes could be reported, but large areas are rare and we felt this would be too complex to interface with SEM navigation software.

IV.A Controlled Experiments

A set of controlled and production experiments were conducted. The controlled experiments used a focused ion beam (FIB) to inject bridging faults at known locations within the chip and determine how well the CAFDM software performed in identifying the locations. Due to the limitations of the FIB, all faults were injected on the metal4 layer. Faults were injected into 13 chips, which were then tested, and the test outputs used for diagnosis. For the production experiment, 10 chips that failed wafer scan test were analyzed with CAFDM and one sent to failure analysis. The chip sample sizes were limited by available personnel and equipment time.

The goal of the experiments was to determine the diagnostic contribution of each knowledge source, and how each contributed to project metrics, and whether the metrics were achieved. The knowledge sources included:

1. Stuck-at suspects from FastScan diagnosis
2. Potential bridges to the stuck-at suspects
3. Suspects filtered by fault simulation
4. Location information of filtered suspects
5. Layer information for filtered suspects
6. Ranking based on critical area

Table 2. Controlled Experiment Diagnosis Results

Chip	FastScan™	CAFDM	Search Area	Layers
FIB1	one node (#1)	bridge (#1)	27 μm^2	M4
FIB2	FIB scan chain			
FIB3	one node (#6)	bridge (#1)	263 μm^2	M4
FIB4	two nodes (#4, #12)	bridge (#1)	86 μm^2	M4
FIB5	FIB scan chain			
FIB6	none	bridge (#1)	91 μm^2	M1, M2, M4
FIB7	two nodes (#1, #20)	bridge (#1)	61 μm^2	M4
FIB8	two nodes (#16, #21)	bridge (#1)	54 μm^2	M4
FIB9	FIB scan chain			
FIB10	one node (#1)	bridge (#1)	4 μm^2	M2, M3, M4
FIB11	two nodes (#1, #2)	bridge (#1)	57 μm^2	M4
FIB12	scan chain problem			
FIB13	one node (#1)	bridge (#1)	400 μm^2	M4

It is not possible to measure a reduction in diagnosis time on the injected faults, since their location is already known. Similarly, equipment limitations prevented direct use of the predicted layer and location information during failure analysis of the production sample. Instead we use the predicted fault site area as a surrogate for search time. Given the small number of samples, only rough estimates of metrics can be made.

The original project plan was to interface the CAFDM software to LogicMap to provide access to chip layout and visualize the suspect locations. LogicMap was not available in time for the case study, so a temporary interface was developed to read the layout into the CAFDM software and visualize the result.

The results of the controlled experiments are summarized in Table 2. The first column identifies the chip. The second column lists how many of the bridged nets (nodes) appear on the FastScan stuck-at fault list, and their rank on the list. For example, for FIB8, both of the bridged nets appear on the suspect list at positions #16 and #21. The third column lists whether the bridge was correctly identified by CAFDM (bridge) or not (none) and its rank on the filtered suspect list. The fourth column lists the critical area. As noted above, the search area is relatively large compared to the average due to the long parallel runs on metal4. Still these areas are within a few SEM fields of view. For example, the top-ranked critical area for FIB8 is a 65 μm long parallel run of two metal4 wires. The last column lists the predicted layers. In the cases where multiple layers occurred, the critical areas on different layers were all in the same general location.

Of the ten chips, FIB2, FIB5 and FIB9 had a fault injected into the scan chain by mistake (the wrong bridging fault locations were selected). FIB12 failed its scan chain test after packaging. In these cases FastScan could not be run and so CAFDM could not be applied. The chain failures can be easily located with binary search of the scan cells using a SEM while shifting in an alternating 0/1 pattern. This left 9 samples for CAFDM analysis.

In eight chips, one or both of the bridged nets appear on the stuck-at suspect list from FastScan. There were 18 to 184 potential bridges to each stuck-at suspect. All of these chips were successfully diagnosed by CAFDM, with the real bridge being ranked #1 on the suspect list, with a Hamming distance of about 5. Compared to the thousands of bits in the output vectors, this is a very close match. In seven of those chips the Hamming distance of the #2 suspect was much larger (500-600) than that of the #1 suspect, and so would be discarded with any reasonable user-defined Hamming distance threshold. In the eighth chip the top two suspects had the same small Hamming distance. The critical area of the real bridge was much larger than the other bridge, and so the real bridge was ranked #1. Only two samples predicted layers in addition to metal4, and these critical areas were a subset of the metal4 critical area. This is useful in failure analysis since if the defect is not found on metal4, stripping can be done down to the next predicted layer. In all samples the Wired-AND bridging model resulted in the smallest Hamming distance, and the Wired-OR model gave poor results.

Neither of the bridged nets appeared in the stuck-at suspect list for chip FIB6. Considering bridges three away from the stuck-at suspects also did not identify the bridge. Backconing identified 66 potential bridges, one of which was the real bridge, which was ranked #1 by fault simulation. As in the first 8 samples, the other potential bridges had a much larger Hamming distance.

Comparing the results from FastScan and CAFDM one can see that the stuck-at suspects contained one of the bridged nets in all but one case, but in three cases they were down the ranked list. Furthermore, since we selected nets that ran on metal4 some of the time, they were all relatively long regional nets, and would take a significant time to search to find the defect. Thus the stuck-at suspects alone do not result in a small search area. Figure 5 shows the FIB site of chip FIB3. The potential bridges to the stuck-at suspects are highlighted in Figure 6. The arrow points to the bridge location. As can be seen, the suspect nets extend over much of the chip area, and would take a very long time to search by SEM. Thus bridges to the stuck-at suspects only make matters worse in terms of search time. Figure 7 highlights the top-ranked bridged nets found by CAFDM. The cross in the middle of the figure is the bridge location. One net is about 3 mm long, and would take a while to search by SEM. Thus the bridge pair alone is not sufficient to guarantee a small search area. But since the two nets are adjacent in only a small region, the critical area is only 263 μm^2, about 5 fields of view in the SEM. This fault site is seen in Figure 8, with the cross marking the FIB location. So location information is very important in reducing search time. The layer information also greatly reduces the search space. Out of the 6 potential layers, only one layer was predicted most of the time. The critical area information proved somewhat useful in one sample, but in general one must analyze all fault sites that closely model the observed chip behavior.

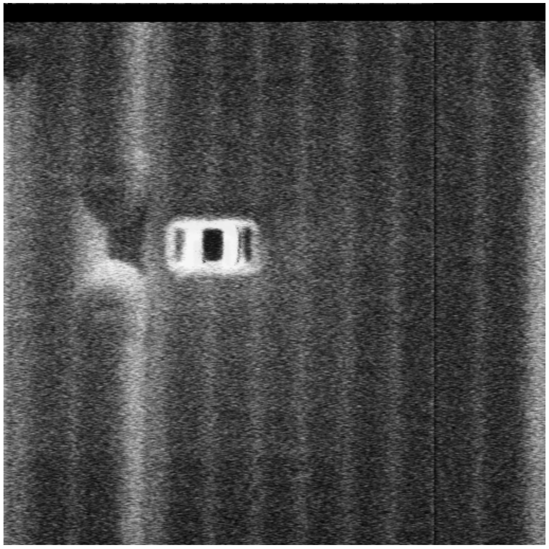


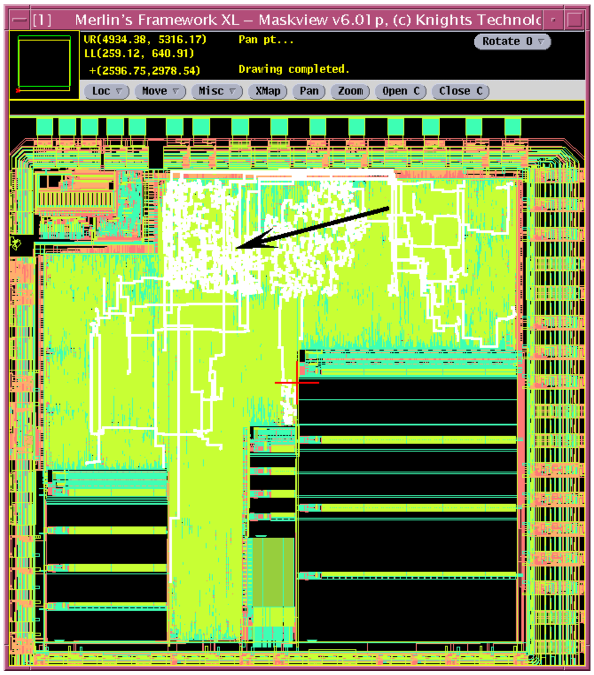
Figure 5. SEM of FIB3 bridging fault site

In the controlled experiment, the CAFDM software was able to achieve or exceed all of our metrics: 0% failure rate, a small suspect list 100% of the time, the correct layer, and correct small search area for all bridging fault suspects. Since the FIB locations were known, we were not able to directly measure defect localization time, but we assume it would be small given the small search area. However it should be kept in mind that the FIBed bridges have low resistance and behave reasonably like one of the modeled faults (wired-AND). The results are unlikely to be as good for real defects, such as resistive bridges or resistive opens.

The fault injection and simulation time per chip ranged from about 20 minutes on a SPARC 20 workstation for the cases in which the bridge appeared on the stuck-at suspect list to about 500 minutes for the backcone case. About half this time was Perl scripts, and half FastScan execution. The Perl scripts and simulation procedure can be redesigned to greatly speed them up.

IV.B Production Experiments

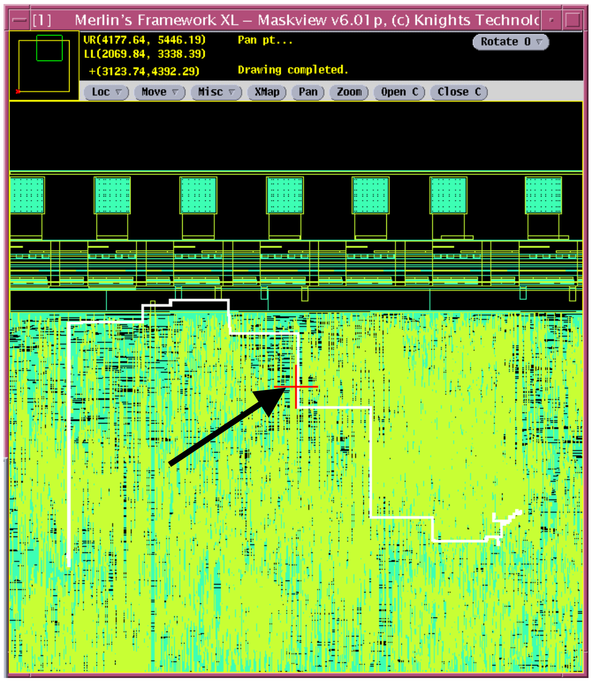
The production experiments were conducted on 10 devices that failed wafer scan test. These devices passed



[1] Merlin's Framework XL – Maskview v6.01p, (c) Knights Technolc

UR(4934.38, 5316.17) Pan pt...
LL(259.12, 640.91)
+(2596.75,2978.54) Drawing completed.

Figure 6. Potential bridging fault nets


[1] Merlin’s Framework XL – Maskview v6.01p, (c) Knights Technolc

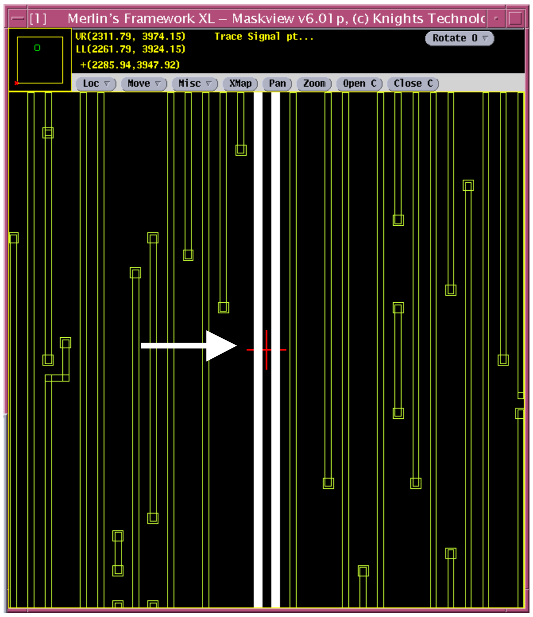
UR(4177.64, 5446.19)
LL(2069.84, 3338.39)
+(3123.74,4392.29)

Pan pt...

Drawing completed.

Figure 7. CAFDM predicted and actual bridging fault location

DC parametrics, memory test, and scan integrity test. These devices also passed the functional patterns used to test a specific portion of the device. Out of the 10 devices, 3 fail only under maximum voltage conditions. These are being reserved for later analysis. For 3 of the remaining 7 devices, stuck-at diagnosis did not provide a suspect candidate list, so backconing must be applied. Due to time constraints, these devices were also deferred until later. CAFDM analysis was performed on the remaining 4 devices. Sproing [22] analysis is also being performed to confirm CAFDM results and to determine whether it can identify suspects in the devices where FastScan diagnosis failed.


[1] Merlin’s Framework XL – Maskview v6.01p, (c) Knights Technolc

UR(2311.79, 3974.15)
LL(2261.79, 3924.15)
+(2285.94,3947.92)

Trace Signal pt...

Figure 8. Layout of FIB3 injected bridging fault site

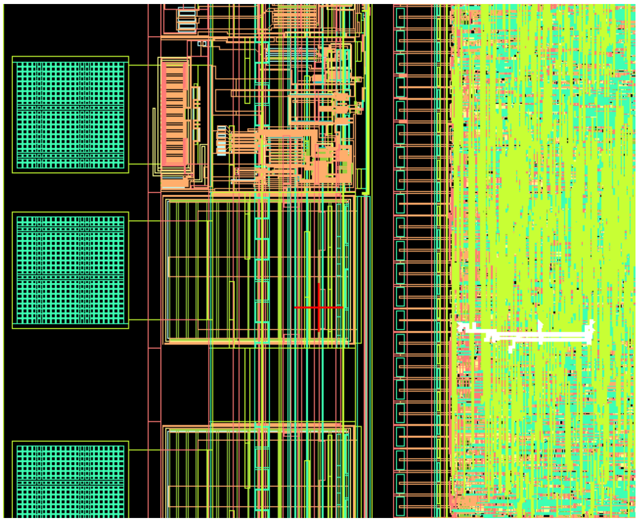


Figure 9. CHIP4 potential bridging fault nets

CHIP4 was selected for failure analysis. In this device two bridges were ranked as the top candidates, with much smaller Hamming distance than other candidates. The first bridge explained 30 of 31 failing vectors, while the second explained 28 of 29 failing vectors. The remaining candidates could not explain any failing vectors. The bridged lines are shown in Figures 9 and 10. The first bridge is predicted to be on the metal2 or metal3 layers, while the second bridge is predicted to be on the metal1 or poly layers. As can be seen, the search region is small enough to perform physical failure analysis (PFA) within hours, as opposed to days. Moreover, the layer information further reduces the search time and increases the chance of PFA success.

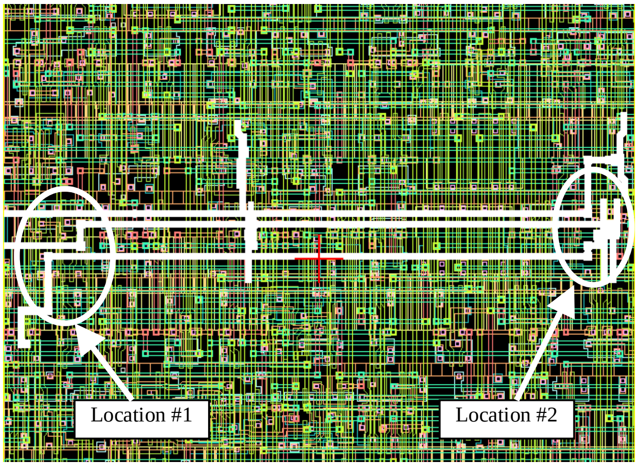


Figure 10. CHIP4 fault sites

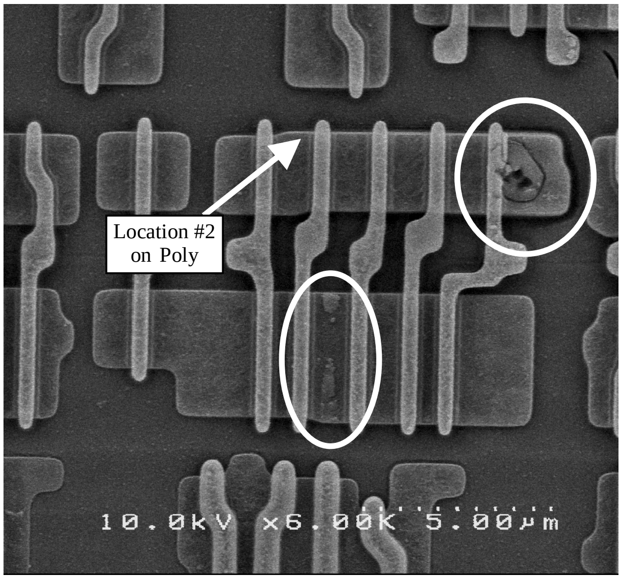


Figure 11. SEM of defects at location #2

Failure analysis of CHIP4 was inconclusive. A misunderstanding combined with deprocessing problems resulted in the metal1 and metal2 layers being removed prior to inspection. Thus it could not be determined if any defect was present in the predicted locations on those layers. No defect was observed on the metal3 or poly layers. As shown in Figure 11, defects were observed in nearby diffusion regions, but these are several transistors away from the predicted nets and should not affect their behavior. We do not know if these defects were part of a larger defect that was removed.

We plan on performing failure analysis on the remaining devices. Despite the inconclusive failure analysis results in Figure 11, we believe that the CAFDM approach is promising. On the devices to which it was applied, it identified a small number of suspects as being a good explanation for the faulty behavior, and much better than other suspects. This is similar to what we found with the controlled experiments. Thus we expect to get similar successful failure analysis results on the remaining devices.

Table 3. Sproing DS Diagnosis Results

Chip	Sproing Results without Layout Information			
	Stuck-at	Bridging	Mixed	Comments
FIB1	1 node (#1)	1 node (#1)	1 node (#1)	Both nodes in input
FIB2	Not Applicable			
FIB3	None			
FIB4	1 node (#1)	1 node (#4)	1 node (#4)	One node in input
FIB5	Not Applicable			
FIB6	None			
FIB7	1 node (#1)	1 node (#5)	1 node (#2)	
FIB8	Not Applicable			Neither node in input
FIB9	Not Applicable			
FIB10	1 node (#10)	1 node(#1)	1 node(#1)	Both nodes in input
FIB11	1 node(#9)	None	None	One node in input
FIB12	Not Applicable			
FIB13	1 node(#6)	None	None	Both nodes in input

IV.C Comparison to Sproing

In order to compare our results to more recent diagnostic approaches, we performed diagnosis of our controlled samples using the diagnosis tool Sproing [22]. Sproing can perform stuck-at, bridging, or mixed model (both stuck-at and bridging) diagnosis. It uses a training set of faults to build a dictionary, and then ranks the list of suspects for each of the model categories. Sproing uses a subset of the entire fault list. Approximately 22,000 collapsed, detected (DS) stuck-at faults were used in this experiment. Sproing can accept a bridging pair list, such as generated by FedEx, but format problems precluded its use.

The Sproing diagnostic results without layout (bridging pair) information are summarized in Table 3. The analysis was first done with the DS faults, and then logically-equivalent (EQ) faults were added to the training set. With DS faults only, Sproing was able to correctly diagnose the fault for six chips, with the fault being ranked #1 by at least one of the models in 4 cases. In three cases the diagnosis failed. Sproing requires at least one of the bridged nodes to be in the input training list. Such was not the case for FIB8, so it could not be diagnosed. FIB4 and FIB11 had only one of the bridged nodes in the input list. This can reduce the accuracy of the diagnosis. In the case of FIB4, stuck-at diagnosis had one of the bridged nodes as the #1 suspect. The other side of the bridge was not in the top 10. The other

Table 4. Sproing EQ Diagnosis Results

Chip	Sproing Results Without Layout Information			
	Stuck-at	Bridging	Mixed	Comments
FIB1	1 node (#1)	1 node (#1)	1 node (#1)	One node in input
FIB2	Not Applicable			
FIB3	1 node (#3)	None	None	Both nodes in input
FIB4	2 nodes (#1, #7)	Both (#5)	Both (#5)	One node in input Other node was EQ. Suspect was the DS of the EQ.
FIB5	Not Applicable			
FIB6	None			
FIB7	1 node (#1)	1 node (#2, #5, #6)	1 node (#2, #5, #6)	1 node paired with other nodes
FIB8	1 node (#1)	1 node 8/10 suspects were the bridged node.	1 node 8/10 suspects were the bridged node.	FIBed nodes were EQ. DS in input, was same for both FIBed nodes. Another EQ of DS was suspect #1. EQ got in input because SA0 of the node was a DS while SA1 was an EQ to the FIBed nodes.
FIB9	Not Applicable			
FIB10	1 node (#10)	1 node(#1)	1 node(#1)	One node input
FIB11	3 nodes (#3, #4)	Both (#2)	Both (#2)	EQ analysis showed both nodes in input
FIB12	Not Applicable			
FIB13	2 nodes (#1, #9)	None	None	Both nodes in input

side was ranked #4 in the bridging and mixed diagnosis suspect lists, while the bridged node was not in the list. The other side of the FIB4 bridge was classified by FastScan as an equivalent (EQ) fault and so not included in the Sproing dictionary. After adding equivalent faults to the training set, Sproing analysis improved, with the results shown in Table 4. For FIB4, the stuck-at model identified both nodes with rankings #1 and #7 respectively. The bridged nodes were ranked #5 in both the bridging and mixed model models.

A direct comparison of Sproing and CAFDM results is not possible since CAFDM had layout information and Sproing did not. One advantage of CAFDM is that it is not necessary to build a large dictionary in advance.

V. Conclusions and Future Work

In this work we have described a computer-aided fault to defect mapping (CAFDM) approach to defect diagnosis, that combines existing stuck-at diagnosis infrastructure with physical design information to reduce diagnosis time. Our controlled experiments met or exceeded the project metrics. Our production experiments were promising but inconclusive. More production experiments are being undertaken to completely evaluate the CAFDM approach.

The experiments highlighted several improvements needed to make CAFDM practical in a production environment. The first is to include more bridging fault models, to improve the diagnostic accuracy. For example, dominant faults were found to be the most common type in a recent microprocessor [23]. The second improvement is to reduce the per-chip CAFDM diagnosis time. For failure analysis the simulation can be overlapped with sample preparation. However manufacturing monitoring applications [5] require per-chip diagnosis time of less than a minute. Our current fault simulation approach cannot achieve this, particularly on global nets. The Reset net of the streaming audio controller has 93,376 bridging pairs, which would take CPU months to simulate. The solution we are pursuing is to modify the netlist so that the bridging behaviors look like stuck-at faults, and the stuck-at diagnosis engine can accurately rank suspects. The pattern fault capability of TestBench avoids the need for netlist modification, since it can model many bridging fault behaviors. Finally, the CAFDM software must be tightly integrated with a design database and SEM navigation system, in order to permit rapid translation from tester failures to fault site locations and layers.

VI. Acknowledgements

This work was funded by International Sematech under project DRTB002 and by an internship for Zoran Stanojevic in the Design and Test Technology Group, Mixed Signal Products, Texas Instruments, Inc. This work benefited from the advice and support of Craig Force of Texas Instruments, and Sagar Sabade, a Texas A&M student interning at TI.

References

[1] Semiconductor Industries Association, *International Technology Roadmap for Semiconductors*, 1999.

[2] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*, IEEE Press, 1990.

[3] M. Abramovici, P. R. Menon, and D. T. Miller, “Critical Path Tracing: An Alternative to Fault Simulation*”, IEEE Design & Test of Computers*, vol. 1, 1984, pp. 89-93.

[4] H. Balachandran, J. Parker, G. Gammie, J. Olson, C. Force, K. Butler and S. Jandhyala, “Expediting Ramp-to-Volume Production”, *IEEE Int’l Test Conf.*, Atlantic City, NJ, Sept. 1999, pp. 103-112.

[5] H. Balachandran, J. Parker, D. Shupp, S. Butler, K. Butler, C. Force and J. Smith, “Correlation of Logical Failures to a Suspect Process Step”, *IEEE Int’l Test Conf.*, Atlantic City, NJ, Sept. 1999, pp. 458-466.

[6] C. Hawkins, J. Doen, A. Righter, and F. Ferguson, “Defect Classes - An Overdue Paradigm for CMOS IC Testing”, *IEEE Int’l Test Conf.*, Washington DC, 1994. pp. 413-425.

[7] D. Lavo, T. Larrabee, and B. Chess, “Beyond the Byzantine Generals: Unexpected Behavior and Bridging Fault Diagnosis”, *IEEE Int’l Test Conf.*, Washington DC, 1996, pp. 611-619.

[8] S. Millman, E. McCluskey, and J. Acken, “Diagnosing CMOS Bridging Faults with Stuck-at Fault Dictionaries”, *IEEE Int’l Test Conf.*, Washington DC, 1990, pp. 860-870.

[9] D. Lavo, T. Larrabee, B. Chess, and F. Ferguson, “Diagnosing Realistic Bridging Faults with Single Stuck-at Information”, *IEEE Trans. CAD*, pp. 255-268, 1998.

[10] J. Saxena, K. Butler, H. Balachandran, and D. Lavo, ”On Applying Non-Classical Defect Models to Automated Diagnosis”, *IEEE Int’l Test Conf.*, Washington DC, 1998, pp. 748-757.

[11] C. Timoc, et al., “Logical Models of Physical Failures”, *IEEE Int’l Test Conf.*, 1983, pp. 546-553.

[12] Y. Gong and S. Chakravarty, “Locating Bridging Faults Using Dynamically-Computed Stuck-At Fault Dictionaries”, *IEEE Trans. CAD*, Vol. 17, No. 9, Sept. 1998, pp. 876-887.

[13] W. Maly, “Defect and Design Error Localization Procedure – Theoretical Basis”, *VLSI Design Conf.*, India, 1991, pp. 243-248.

[14] S. Naik and W. Maly, “Computer-Aided Failure Analysis of VLSI Circuits Using IDDQ Testing,” *VLSI Test Symp*, Atlantic City NJ, 1993, pp. 106-108.

[15] V. Sar-Dessai and D. M. H. Walker, “Resistive Bridge Fault Modeling, Simulation and Test Generation,” *IEEE Int’l Test Conf.*, Atlantic City NJ, 1999, pp. 596-605.

[16] P. Nigh, D. Forlenza and F. Motika, “Application and Analysis of IDDQ Diagnostic Software”, *IEEE Int’l Test Conf.*, Washington DC, 1997, pp. 319-327.

[17] R. Aitken, “Fault Location with Current Monitoring,” *IEEE Int’l. Test Conf.*, Washington DC, 1991, pp. 623-632.

[18] C. Thibeault and L. Boisvert, “Diagnostic Method Based on Δ IDDQ Probabilistic Signatures: Experimental Results*”, IEEE Int’l Test Conf.*, Washington DC, Oct. 1998, pp. 1019-1026.

[19] D. Lavo, T. Larrabee and J. Colburn, “Eliminating the Ouija Board: Automatic Thresholds and Probabilistic IDDQ Diagnosis”, *IEEE Int’l Test Conf.*, Atlantic City NJ, Sept. 1999, pp. 1065-1072.

[20] D. Gaitonde and D. M. H. Walker, “Hierarchical Mapping of Spot Defects to Catastrophic Faults - Design and Applications”, *IEEE Trans. Semiconductor Manufacturing*, vol. 8, no. 2, 1995.

[21] S. Zachariah, S. Chakravarty and C. Roth, “A Novel Algorithm to Extract Two-Node Bridges*”, Design Automation Conference*, Los Angeles, CA, June 2000, pp. 790-793.

[22] D. Lavo, T. Larrabee, B. Chess and I. Harinto, "Probabilistic Mixed-Model Fault Diagnosis," *IEEE Int’l. Test Conf.*, Washington DC, Oct. 1998, pp. 1084-1093.

[23] S. Ma, I. Shaik and R. Fetherston, “A Comparison of Bridging Fault Simulation Methods”, *IEEE Int’l Test Conf.*, Atlantic City NJ, 1999, pp. 587-595.