# Faster Defect Localization in Nanometer Technology based on Defective Cell Diagnosis

Manish Sharma , Wu-Tung Cheng and Ting-Pu Tai [1]
{manish_sharma,wu-tung_cheng,ting-pu_tai}@mentor.com

Y.S. Cheng and Will Hsu [2]
{yscheng,will_hsu}@tsmc.com

Chen Liu and Sudhakar M. Reddy [3]
{cheliu,reddy}@engineering.uiowa.edu

Albert Mann [4]
albert.man@amd.com

1. Mentor Graphics Corporation, 8005 SW Boeckman Rd., Wilsonville, OR 97070, USA
2. Taiwan Semiconductor Manufacturing Company, 8, Li-Hsin Rd. 6, Hsinchu Science Park, Hsinchu, Taiwan 300-77, R.O.C.
3. Department of ECE, University of Iowa, Iowa City, IA 52242
4. AMD, Inc., 1CV Commerce Valley Drive E., Markham, Ont Canada, L3T 7N6

## Abstract

*In this paper we present practical techniques that enable diagnosis of defective library cells in a failing die. Our technique can handle large industrial designs and practical situations like compressed test patterns with multiple exercising conditions per pattern and sequence dependent defects. Being able to accurately differentiate between cell-internal and interconnect defects leads to a faster root cause failure analysis at a reduced cost. This capability was applied on an AMD graphics chip using 90nm at TSMC. In all of the failing dies that underwent physical failure analysis, the defective library cell identified by diagnosis was verified to be correct by failure analysis. Currently this capability is successfully used to diagnose another design using TSMC's 65nm technology.*

## 1. Introduction

When a manufactured die fails scan based structural testing, logic diagnosis is typically used to determine the source of failure inside the die. This information enables root cause analysis which can lead to fabrication process and/or design changes that result in an overall higher yield. Logic diagnosis tools today [1][2][3][4][5][13][18] can determine the most likely location inside a failing die from which the failures originate. However, this location information (which is typically a pin or a net in the design) does not tell whether the real defect is on the interconnecting wire (also called *back end defects*) or inside the library cell (also called *cell internal defect or front end defects*) associated with the identified location.

With integrated circuit fabrication technology advancing from 90nm to 65nm and beyond, the ability to distinguish between a cell internal defect and an interconnect defect is becoming critical for faster defect localization. The main reason for this is that for 90nm and beyond, a significant number of manufacturing defects and systematic yield limiters lie inside library cells. One of the causes of this is the increasing use of custom designed cells to cope with higher process variations. Hence it is important to know that, for a failed die, the defect lies inside a cell instead of on an interconnecting wire. First, this leads to a faster and cheaper physical failure analysis (PFA) process. With the above knowledge the failing die can be directly de-layered all the way down to metal layer 1 (where all the intra-cell connections are) without the need to look at metal layer 2 or higher. This speed-up is becoming more significant with shrinking fabrication technologies that use more and more metal layers. Having fewer layers to examine during PFA also reduces the overall cost of the process. Second, knowing cell internal defects greatly helps in collecting defect statistics that can point to systematic yield limiting issues in library cells. As an example, for a low yielding wafer, if the majority of the defects are in cells then this can point to certain process steps without going through PFA.

Previous work on cell internal diagnosis can be put in two general categories. Conventional diagnosis works on a logic level model of the design which may not preserve the actual physical implementation of library cells. Hence, the first category of techniques [6][7] use an enhanced model of the design to perform diagnosis. We refer to these as *defect model based diagnosis* techniques. This model preserves enough physical level information to represent a class of defects inside a cell e.g. transistor level bridges, or transistor stuck opens etc. Doing so enables diagnosis to identify defects inside a cell. Another variant of such techniques use a *pattern fault model* which models specific defect behaviour by pre-specifying conditions under which a defect may be excited [8][9]. These conditions can be determined by simulating likely defect types. The main drawback of such techniques is the dependence on a specific defect model for diagnosis. This is risky since unknown defect types, not represented in the model, may go undiagnosed.

The second category of techniques, which we refer to as *excitation condition based diagnosis*, does not require any special circuit model [10][11][12]. It is based on the realistic assumption that the excitation of a defect inside a cell


1-4244-1128-9/07/$25.00 © 2007 IEEE

will be highly correlated to the logic values at the input pins of the cell. On the other hand an interconnect defect's (e.g. a bridge) behaviour will depend more on the logic values on the nets neighbouring the defective wire. Based on this assumption, *failing patterns* (test patterns that fail on the ATE for a failing die) are used to determine input logic value combinations that potentially excite a cell internal defect, also referred to as *failing excitation conditions*, for selected candidate cells. Similarly, *observable passing patterns* (test patterns that pass on the ATE for the failing die, however are capable of observing a fault effect at the defect site) are used to determine *passing excitation conditions* (conditions that do not excite or propagate the defect inside the cell) for candidate cells. Based on the assumption above, a defective cell can be isolated from interconnect defects by correlating the passing and failing conditions. Furthermore, the excitation conditions determined in this process can be used along with SPICE or switch level simulations to determine the actual defect inside the cell [11].

Considering the advantages of the excitation condition based diagnosis we started off by implementing this technique and testing it in a controlled experiment in which cell internal defects were injected and corresponding fail logs generated by simulation for an industrial design. These fail logs were then diagnosed using the excitation condition based strategy. Surprisingly we found that the technique was able to correctly identify the defective cell in only 25% of the cases. Upon investigation the main problem that was identified was that for industrial designs, it is very common to have a defect site being exercised multiple times in different ways during the *capture phase* of a test pattern.

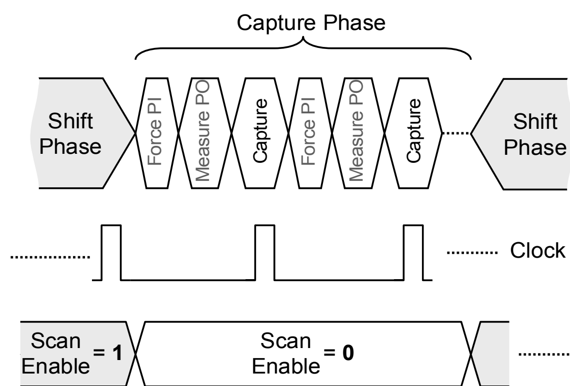



**Figure 1: Scan Test Process.**

In order to understand this let us look at the process of scan based logic test. Figure 1 shows the sequence of events in a typical scan test pattern Test data is first loaded into the scan chains during the *shift phase* in which the scan enable signal is forced high. Following the shift phase there is a repeating sequence of the following operations: force the primary inputs to appropriate values, measure the primary outputs and pulse the capture clock to capture the system logic response to the test pattern in the scan chains. This phase of the testing process is referred to as the *capture phase*. In this paper we are concerned only with failures in the system logic, in other words failures that occur during the capture phase. Failures during shift phase are caused by defects on the scan chain shift path and are a subject of another research topic: scan chain diagnosis [15][16][17].

During the capture phase of a test pattern a defect site may be exercised multiple times. This is due to various reasons like multiple capture cycles, the presence of both leading and trailing edge flops in the design etc. This makes the mapping from failing (observable passing) test patterns to failing (passing) excitation conditions a non-trivial task. This is because conventional stuck-at fault simulation, which is the basis for most diagnosis algorithms, does not provide information on which exercising conditions in a single test pattern are true cell internal defect failing excitation conditions. A *passive excitation condition extraction* strategy, which assumes all exercising conditions in a failing (observable passing) pattern are true failing (passing) excitation conditions, does not work as shown by our experiments. To the best of our knowledge none of the previous work has addressed this issue.

In order to overcome the problem, in this paper we present a practical *active excitation condition extraction* algorithm to heuristically determine the true failing and passing excitation conditions for candidate defective cells from test patterns with multiple exercising conditions with high accuracy. Our technique does not change the stuck-at fault simulation in any way and is still based on the logic gate level model of the design. Hence it does not incur any significant performance penalties. Also, it can handle sequence dependent defects like transistor stuck-opens which require two values for excitation. Finally, this technique works with compressed test patterns for designs with on-chip test compression.

The rest of this paper is organized as follows. We begin by defining the terms used throughout the paper in Section 2. Next, Section 3 describes the excitation condition based cell internal diagnosis algorithm and discusses the problem caused by multiple exercising conditions. Section 4 describes our proposed technique to overcome this problem. We first applied our technique in a controlled environment where known cell internal defects were injected

and fail-logs were created by simulation. Then the technique was applied to real fail logs for an AMD graphics chip and the results verified by PFA. This design had on chip EDT compression logic [13] with a 77x compression ratio. These experiments are presented in Section 5. The final section concludes the paper.

## 2. Terms and Definitions

The following are definitions of terms used throughout this paper:

**Shift Phase:** The phase in the operation of a scan based test during which the test pattern values are loaded into the scan chains, and, at the same time the captured response values for the previous pattern are unloaded.

**Capture Phase:** The phase in the operation of a scan based test during which the response of the logic under test to the test values is captured into the scan chains by pulsing capture clocks. The response values on the primary outputs are also measured.

**Failing Die:** A die that fails structural ATPG testing.

**Cell Internal Defect:** A manufacturing defect inside a library cell. Also called front-end defects.

**Defective Cell:** A library cell with a defect inside it.

**Interconnect Defect:** A manufacturing defect on the wires interconnecting library cells. Also called back end defects.

**Failing Patterns:** Test patterns that fail on the ATE for a failing die.

**Passing Patterns:** Test patterns that pass on the ATE for a failing die.

**Observable Passing Pattern:** An observable passing pattern is defined with respect to a defective library cell candidate. If a passing pattern detects a stuck-at-1 or stuck-at-0 fault on the cell output pin then it is called an observable passing pattern for the cell.

**Exercising Condition:** A binary logic value combination that gets applied to the input pins of a library cell in the design during the capture phase of a test pattern. Note that, for simplicity, we restrict a majority of the discussion to a single input value combination as an excitation condition. However, as mentioned before certain defects like transistor stuck opens may require a sequence of input value combinations to excite the defect. All the discussion in this paper easily extends to such defects. We discuss this in more detail in Section 4.

**Failing Excitation Condition:** An exercising condition of a defective cell that excites the cell internal defect and propagates the faulty value to the cell output pins.

**Passing Excitation Condition:** An exercising condition of a defective cell that does not excite the cell internal defect, or does not propagate the faulty value to the cell output pins.

## 3. Excitation Condition Based Cell Internal Diagnosis Algorithm

The excitation condition based cell internal defect diagnosis technique [10][11][12] is based on the basic premise that if a cell is defective then the defect inside the cell will be excited and observed on some cell output pins only by certain specific input values to the cell. For other input values to the cell the defect will remain unexcited or its effect will not propagate to a cell output pin, which means the overall cell function will remain error free. On the other hand, if a defect is on the interconnecting wires between cells, then its excitation should be relatively independent of the input values to cell that drives the defective wire. In such cases the defect excitation will likely be strongly dependent on the values on the wires neighbouring the defective wire. As an example, consider a two input XOR cell driving a net, named /net1, as shown in Figure 2. The net named /net2 neighbours /net1 in the physical layout of the circuit.

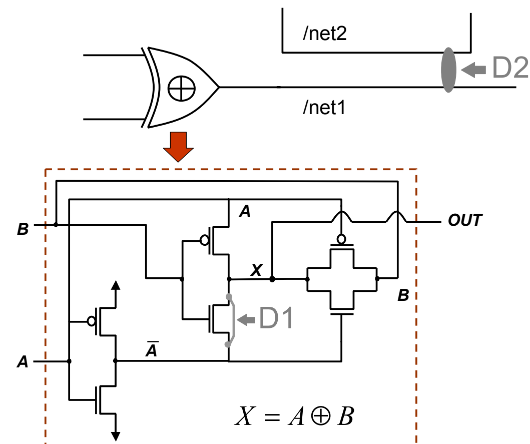



**Figure 2: Cell Internal -vs- Interconnect Defect.**

Consider two defects as shown in Figure 2: D1, a bridge defect inside the XOR cell and, D2, a dominant bridge from /net2 to /net1 where /net2 dominates /net1. In this case, it can be seen that in order to excite the defect D1, the following two conditions are required. First, the input B to the XOR cell has to be **0** so that the NMOS transistor connected to the defect site is OFF. This is because when the NMOS transistor, whose source and drain are bridged, is ON, the behaviour of the defective cell is identical to that of a defect free cell. Second, it requires the A input to be **1** so that the pass gate structure that connects B to X is disabled and the inverter structure between A and $\overline{A}$ (of which the defective transistor is a part of) is enabled. Under these conditions a defect-free cell will have an output of **1**. However, for the defective cell the output will be **0** since it will be pulled down by the short D1 in the

NMOS transistor. For the remaining three input value combinations on A and B (**00**, **11** and **01**) the cell functionality will remain error free. Hence, in the presence of D1, only those test patterns will potentially fail that apply a **10** to the two inputs of the XOR gate. On the other hand, the bridge defect, D2, will be excited only when the two nets, /net1 and /net2 assume opposite logic values. In this case the defect behaviour will be relatively independent of the values at the cell inputs.

The above described distinguishing factor can be used to differentiate between a cell internal defect or an interconnect defect by utilising the following strategy: use the failing patterns to hypothesize what input value conditions excite potential defective cell candidates and then test the hypothesis against observable passing patterns. The hypothesis should be confirmed for actual cell internal defects while it should be rejected for non cell-internal defects. The failing excitation conditions determined from the failing patterns can then be compared against predetermined defect truth tables obtained by injecting defects into cell models and simulating them [11]. This way we can not only determine which cell is defective, but also pin-point the defect inside the cell.

### *The Problem Posed by Multiple Exercising Conditions per Pattern*

Needless to say, the ability to determine failing and passing excitation conditions from failing and observable passing test patterns is critical to the success of the above algorithm. However, this is not a trivial task for patterns that exercise defect sites multiple times, which is typically the case for industrial designs. This situation is mainly caused by three reasons. Firstly, most industrial designs are not 100% full scan; they also contain certain non-scan elements. Multiple cycles of the capture clock are needed to test the faults around these non-scan elements. Another reason for having multiple capture cycles is to detect defects, such as resistive open, transistor stuck open etc., which require a sequence of values to excite them. Multiple capture cycles in a test pattern obviously lead to multiple exercising conditions in the pattern. As an example, consider the hypothetical circuit shown in Figure 3. In this case, the flop A is a non-scan flop, while all the other flops are scan flops. In order to test the stuck-at-1 fault on the output of flop A, a test pattern with two capture clock cycles will be required. As shown, the values **1**, **0, 0** and **1** are scan loaded into the scan flops B, C, D and E, respectively. The first capture clock cycle applies the excitation value of **0** to the fault site and second capture cycle captures the faulty response in scan flop D, which is then scanned out. As can be seen this two capture clock cycle pattern applies two exercising conditions to the NAND cell (**10** and **01**) in the capture phase.

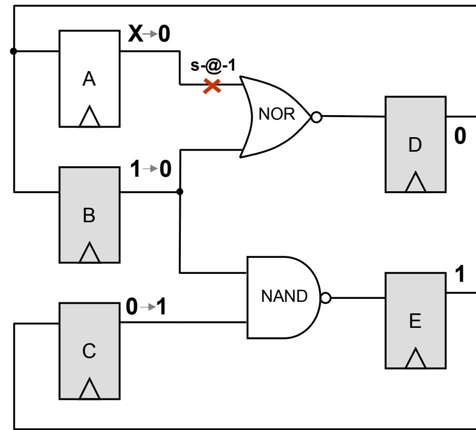



**Figure 3: Example of Multiple Exercising Conditions due to Multiple Capture Clock Cycles.**

Secondly the presence of a mix of leading edge and falling edge triggered flops in the design also lead to multiple exercising conditions in a test pattern. Let us consider the circuit in Figure 4. In this case the flop A is a leading edge triggered flop while flop B is trailing edge triggered. Assume that both the flops are scanned and consider a test pattern that loads a **0** in both the flops. Now, when the capture clock, CLK, is pulsed the flop A will capture the effect of the exercising condition **00** for the NAND cell on the leading edge.

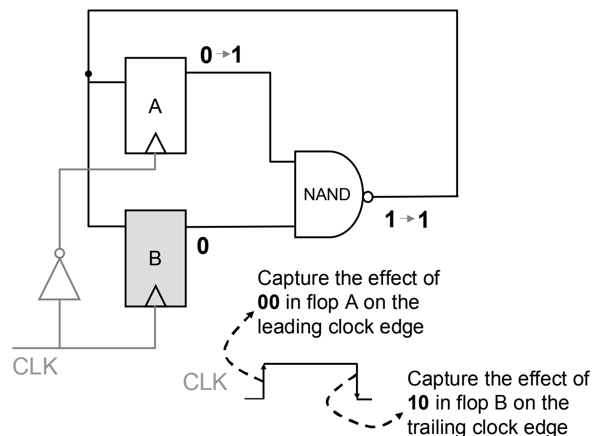



**Figure 4: Example of Multiple Exercising Conditions due to a mix of Leading, Trailing edge flop.**

Next, on the trailing the edge the flop B will capture the effect of the exercising condition **10**. Hence this is another example of a pattern with multiple exercising conditions.

Finally in some cases the clock signal feeds into the system logic under test. In such cases the effect of logic values in the design when the clocks are OFF can be captured in leading edge scan flops or observed at primary outputs, hence resulting in yet another set of exercising conditions.

In order to understand the problem caused by multiple exercising conditions in test, consider the example circuit in Figure 5. Assume that the test patterns applied have three capture clock cycles each. Further assume that for some failing chip there is a single failing test $ft_1$, whose logic simulation values are as shown in Figure 5, and that the failing behaviour is explained by the defect location: D stuck-at-1. In this case the cell MUX_1 is a defective cell candidate and the algorithm would use $ft_1$ to determine the failing excitation conditions for MUX_1. This is where the problem occurs due to multiple exercising conditions, because in this case there are three exercising conditions (corresponding to the three capture clock cycles) to choose from: **101**, **010** and **001**. Any one or moreof these can be the real cell failing excitation condition(s). Since stuck-at-fault simulation does not keep track of the origination and propagation of events by capture cycle, the only information we get from it is whether a fault is detected by a pattern or not. In particular, fault simulation does not tell which cycle was the origination point for a stuck-at-fault activation event which was eventually detected at an observation point (primary output or scan flop).

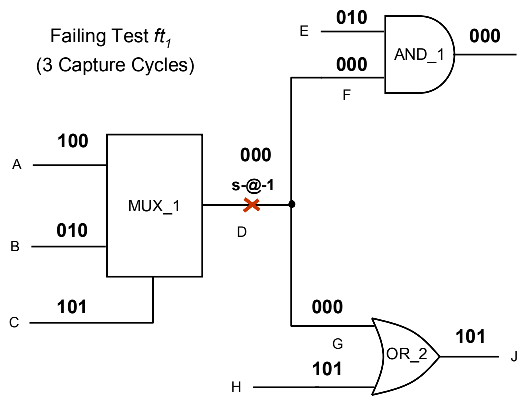



**Figure 5: Simulation Values for Multiple Capture Cycle Pattern.**

For example, from Figure 5 it can be seen that the fault D stuck-at-1 will be excited in all three cycles, however in the first and the last cycles the fault effect is blocked from propagation at the AND_1 and OR_2 gates This means that from this failing test we can only reliably conclude that **010** is a defect excitation condition. **101** and **001** may or may not be excitation conditions in reality since the fault effect originating at the corresponding cycles was not observed.

Ideally, we would want to enhance fault simulation so that it can provide us with the information as to the fault effect in which cycles actually make it to an observation point. However, doing so will be an impractical solution. Firstly, this would require back-tracing from observation points after faulty machine simulation for all failing and observable test patterns for all defective cell candidates, making it an expensive operation. Secondly, for some patterns it may still be an incomplete solution. The reason is that in some cases the fault effect from one capture clock cycle may get mixed together with fault effects from other cycles due to re-convergence. In such cases it will become impossible to determine the fault effect from which cycle is observed at the end of the capture phase. As an example consider the circuit in Figure 6. Assume that flop F is a scan flop. Consider a test pattern with two capture cycles and a stuck-at-1 fault at the output of the AND gate. The values shown in Figure 6 are the simulation values for the test pattern just before each capture clock pulse. As can be seen from the figure, the fault effect from the two different capture clock cycles gets mixed together in the second cycle at the OR gate. Therefore, in this case, if the AND cell is a candidate defective cell it will be impossible to tell from fault simulation which of the two exercising conditions: **01** or **10** are true failing conditions. Hence, this failing pattern may not be usable.

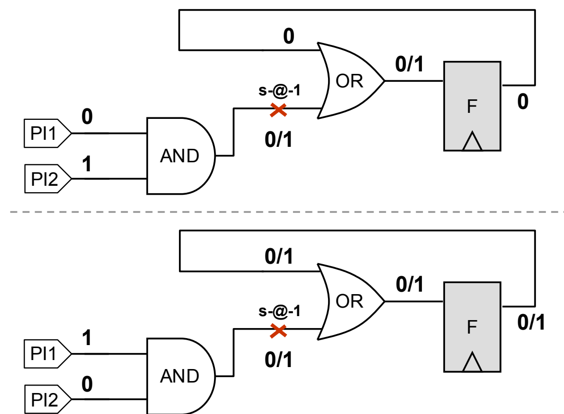



**Figure 6: Example of Mixing of Faulty Values.**

None of the previous approaches presented in the literature address this issue. Therefore, in this paper we have developed a practical heuristic algorithm, which we call *active excitation condition extraction*. This algorithm can determine which exercising conditions in a multiple exer-

cising conditions pattern are actual failing excitation conditions with high accuracy. It does not require any changes to the fault simulation engine and can also handle sequence dependent defects. Furthermore, the algorithm works when using compressed patterns.

## 4. Determining Excitation Conditions from Patterns with Multiple Exercising Conditions

The active excitation condition extraction algorithm starts by extracting all exercising conditions for potential defective cells in a test pattern. This information is recorded for all the failing and observable passing patterns for each candidate cell. Next, this information is used to determine which exercising conditions are the actual failing excitation conditions for the cell internal defect, and which are not. In order to describe how this is done, let the set of exercising conditions that are extracted from a test pattern for a candidate defective cell be referred to as the *Exercising Conditions Collection (ECC)* for that test pattern. Let $ECC_1^f, ECC_2^f, \cdots, ECC_k^f$ denote the *distinct* ECCs for all the failing patterns for a candidate cell. Note that $k \leq$ the number of failing test patterns, and typically it is much less than that. As an example, consider a scenario in which a two-input multiplexer cell, MUX_1, is a candidate defective cell, and, that there are four failing test patterns with simulation values as shown in Figure 7. In this case there will be three failing pattern ECCs:

$$ECC_1^f = \{101,010\} \rightarrow ft_1, ft_4$$
$$ECC_2^f = \{000\} \rightarrow ft_2$$
$$ECC_3^f = \{101,000\} \rightarrow ft_3$$

In a similar fashion, let $ECC_1^p, ECC_2^p, \cdots, ECC_k^p$ denote the distinct ECCs for all the observable passing patterns for a candidate cell. For the observable passing pattern ECCs the number of observable passing patterns associated is also stored. As an example consider the case in Figure 7 and further assume that there are four observable passing patterns that detect the stuck-at-1 fault at the MUX_1 output. Let the simulation values for these be as shown in Figure 8. In this case the observable passing pattern ECCs will be:

$$ECC_1^p = \{101,001\} \rightarrow pt_1, pt_4$$
$$ECC_2^p = \{101,010,000\} \rightarrow pt_3$$
$$ECC_3^p = \{001,101\} \rightarrow pt_2$$

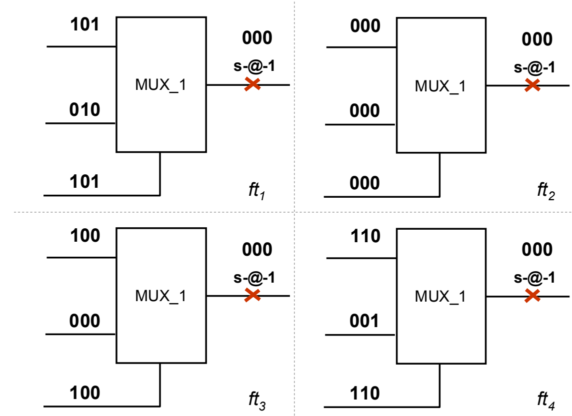



**Figure 7: Simulation Values for Four Failing Tests**

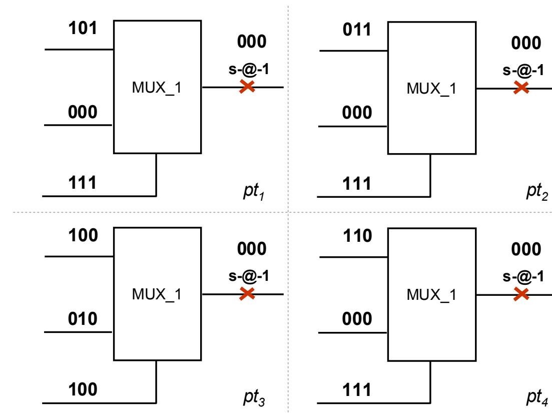



**Figure 8: Simulation Values for Four Observable Passing Tests.**

The ECCs are used to determine the actual failing excitation conditions using the following heuristic algorithm:

**Step 1.** Divide all the exercising conditions into three categories. Categorize all those exercising conditions that occur only in failing pattern ECCs as *failing excitation conditions*. All those exercising conditions that occur only in observable passing pattern ECCs are categorized as *passing excitation conditions*. All the remaining exercising conditions are placed in the *undecided* category. These undecided exercising conditions will be placed in the other two categories in the subsequent steps.

**Step 2.** For the failing pattern ECCs, determine those that contain exactly one undecided exercising condition, say *i*, and no failing excitation condition. This means that the exercising condition *i* must activate the cell internal defect in some failing test pattern; hence

its category is changed to a failing excitation condition.

**Step 3.** For the observable passing pattern ECCs, determine those that contain exactly one undecided exercising condition, say *i*, and no passing excitation condition. This means that the exercising condition *i* must not activate the cell internal defect; hence its category is changed to a passing excitation condition

**Step 4.** If there are still some undecided exercising conditions left, choose one which is associated with the largest number of observable passing patterns and change its category to passing excitation condition. The reasoning behind this is that if an exercising condition occurs in a large number of observable passing patterns then it is likely that it does not excite the cell internal defect.

**Step 5.** If any undecided exercising condition was converted to a passing excitation condition in the previous step, then go back to **Step 2**, otherwise end.

So, for our example, the algorithm will produce the following results: After **Step 1** the exercising conditions **101**, **010** and **000** are placed in the undecided category. Exercising condition **001** is placed in the passing excitation category since it does not occur in any failing test ECC. In **Step 2** the exercising condition **000** will be categorized as a failing excitation condition since it occurs by itself in $ECC_2^f$. At this point we are left with two undecided conditions **101** and **010**. Since the exercising condition **101** occurs in more observable passing patterns than **010** it is more likely to be a passing excitation condition. Hence in **Step 4** this exercising condition's category is changed to passing excitation condition. In the next iteration through the loop the last remaining undecided condition **010** will be changed to a failing excitation condition in **Step 2** because this condition is now the only undecided condition in $ECC_1^f$. Hence, the heuristic will conclude that **000** and **010** are failing excitation conditions and **101** is a passing excitation condition. This means that there will be no conflict between passing and failing excitation conditions for this candidate defective cell.

### *Handling defects requiring a sequence of exercising conditions*

Certain cell internal defects like transistor stuck-opens may require a sequence of exercising conditions at the cell inputs to excite them. As a simple example, consider a NAND cell with an open defect, as shown in Figure 9. As can be seen from the figure a sequence of two exercising conditions are required to excite this defect. The first exercising condition (**00** in Figure 9) is required to charge the output capacitance to a **1.**

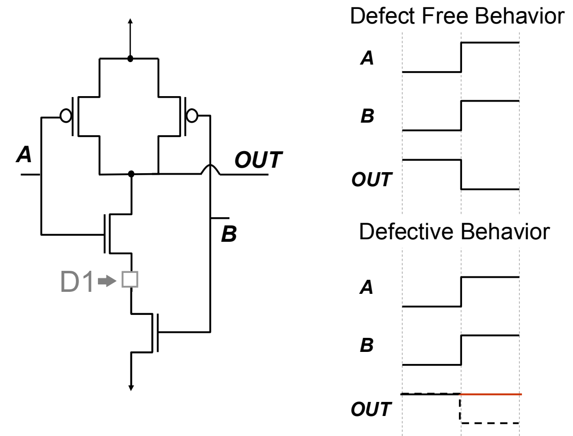



**Figure 9: Open Requires Sequence of Values.**

Note that this can also be achieved by **01** or **10**. The second condition, **11** turns ON both the NMOS transistors so that output will discharge to **0** in a defect free cell, however it will not do so in the defective cell. Hence, in this case the failing conditions will be: **00-11**, **01-11** and **10-11**. The passing conditions will be **XX-10**, **XX-01** and **XX-00**. The behavior of the defective cell under the remaining condition, **11-11**, will depend on whether the output capacitance was charged up before for example during the shifting phase.

Active excitation condition extraction can be extended to handle such situations. The change will be to extend the definition of an exercising condition to include a sequence of *two* binary input value combinations instead of *one*. The rest of the discussion will then automatically cover sequence dependent defects. As an example, with the above change, for the case shown in Figure 7, the failing pattern ECCs will be as shown below:

$$ECC_1^f = \{101-010, 010-101\} \rightarrow ft_1$$
$$ECC_2^f = \{000-000\} \rightarrow ft_2$$
$$ECC_3^f = \{101-000, 000-000\} \rightarrow ft_3$$
$$ECC_4^f = \{101-101, 101-010\} \rightarrow ft_4$$

Note that for the second failing pattern there is no transition. Hence, for this pattern we will also consider the last shift value, since the transition must have come from there. Similarly, the observable passing pattern ECCs will contain a sequence of input value combinations. Once the ECCs have been determined, the same algorithms can be

used to determine true failing and passing excitation conditions.

Now, the nature of the defect, i.e. whether it is sequence dependent or not will not be known beforehand. Hence, in our diagnosis algorithm we always start assuming a sequence dependency of two. The reason for this is that *even if the real defect is not sequence dependent, it will still be identified when extracting a sequence of two conditions.* This is because the passing and failing excitation conditions will still be disjoint by virtue of the second exercising condition in the sequence. When a cell internal defect is identified assuming a sequence dependency of two, an additional analysis step, this time assuming no sequence dependency, can be performed to determine whether the identified defect is sequence dependent or not.

## 5. Experiments and Results

### *Experiments with simulated cell internal defects*

In order to test the effectiveness of the active excitation condition extraction algorithm developed in this paper, we conducted controlled experiments in which the behavior of a failing chip with a cell internal defect during test was emulated. This was done by injecting cell internal defects in a pre-determined list of library cell instances. In order to emulate cell internal defect behavior, the target cell instance was replaced with a *modified* cell in the netlist. This modified cell represented a defective version of the original cell, by having a randomly changed truth table in order to emulate defective behavior. The modified netlist was then simulated against a stuck-at test pattern set to produce a cell internal defect fail log. Active excitation condition extraction based cell internal diagnosis was then run on the fail logs thus produced, and the effectiveness of the technique in isolating the cell instance with the injected failure was studied. The experiment was performed on a 2.1M gate design. A stuck-at fault test set with a 1000 patterns was used. The test patterns had up to two capture clock cycles, as well as a mix of leading and trailing edge flops and clock driven logic. Hence, each test pattern had either three or six exercising conditions depending on the number of times the capture clock was pulsed in the pattern.

A total of 383 cell internal defect fail logs were generated using the above method and were diagnosed based on the active excitation condition extraction algorithm. In order to validate the diagnosis strategy, the percentage of fail logs for which the defective cell was correctly identified was recorded. Furthermore, active excitation condition extraction was validated by matching the failing and passing excitation conditions determined by the algorithm for the defective cell, with those that were used to inject the cell internal defect to begin with. To prove the value of active excitation condition extraction as a key enabler of excitation condition based diagnosis of defective cells (as well as highlight the problem posed by multiple exercising conditions test patterns) the fail logs were also diagnosed based on the passive excitation condition extraction strategy. Recall that this means that it was assumed that all the exercising conditions in the failing patterns are failing excitation conditions and all those in the observable passing patterns are passing excitation conditions.

Results of these experiments are summarized in Table 1. These results clearly prove the effectiveness of our proposed technique. Diagnosis was considered to be correct if the defective cell was identified in the diagnosis report with the highest rank among all candidates. Ranking was performed based on passing and failing pattern information as is routinely done in most diagnosis techniques. As reported in the first row, using passive excitation condition extraction, the excitation condition based cell internal diagnosis algorithm is able to diagnose the correct defective cell instance in only 25% of the cases. In the remaining 75% cases, the diagnosis algorithm was not able to establish whether the defect lies inside the cell or on the interconnecting wire. However, with active excitation condition extraction, the percentage of correctly diagnosed cases goes up to 94%. This means that for 94% of the cases, the correct defective cell was identified and the failing and passing excitation conditions extracted by our algorithm exactly matche (last row in Table 1) with those that were used for defect injection.

**Table 1: Results of Cell Defect Injection Experiments.**

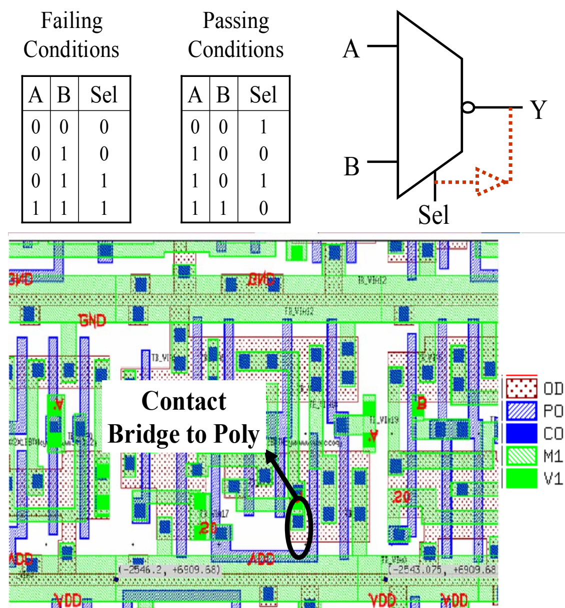
| | |
|---|---|
| % Correctly Diagnosed with **Passive** Excitation Condition Extraction | 25% |
| % Correctly Diagnosed with **Active** Excitation Condition Extraction | 94% |
| Correct failing and passing excitation conditions extracted by **active** excitation extraction? | Yes, for all the correctly diagnosed cases |

For the remaining 6% of the cases for which diagnosis was unsuccessful the main reason was that for these cases a majority of failing and observable passing patterns had *identical* set of exercising conditions. In such a case active excitation condition extraction is not able to distinguish between failing and passing conditions. However, test patterns are highly optimized which means that each pattern attempts to exercise defects in different ways. This results in the rate of occurrence of this special situation to be very small, as is apparent from the experimental results. These results highlight both the severity of the limitation imposed by multiple exercising conditions in test patterns on excitation condition based diagnosis, and the

effectiveness of active excitation condition extraction in solving this problem.

### *Real silicon diagnosis data*

Having validated our technique using controlled experiments, we applied active excitation condition extraction based diagnosis on a set of failing die of an AMD graphics chip fabricated at TSMC using 90nm technology. This was a ~33M gate design with on chip EDT compression logic [13] with a compression ratio of 77x. The analysis was done for two different revisions of the same design. The test sets used had 500 and 1000 compressed test patterns, respectively, for the two revisions. Out of the failing dies that were diagnosed, 7 different die were selected for detailed physical failure analysis (PFA). These 7 die all had a cell internal defect as one of the top candidates reported by diagnosis. PFA results showed that in 7 out of the 7 cases, the failing die indeed had a defect inside the cell instance that was reported as a top candidate in the diagnosis report. Out of these 7 cases, 4 could not be correctly diagnosed with passive excitation condition extraction. This provides further validation of our cell diagnosis approach in a real industrial application. In all of these cases, the PFA process was faster and less costly because diagnosis of defective cells eliminated the need for examining any layers above metal layer 1. Next we provide details on two out of these seven successfully diagnosed cases. **Case-I:** In the first case, the defect was found to be inside a two input inverting multiplexer cell.



Failing Conditions

| A | B | Sel |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |

Passing Conditions

| A | B | Sel |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



**Figure 10: Case-I Defect Site and Gate Model.**

The defective cell had a contact bridge to poly between the output and the select line of the MUX, as shown in Figure 10. At a logic level this bridge caused the defective cell to behave as if there was a buffer from the select input to the output of the MUX, as shown in Figure 10. The figure also shows the failing and passing excitation conditions extracted for this cell by active excitation condition extraction. The reader can easily verify these to be correct. Figure 11 shows a PFA picture of the defect showing the bridge from contact to poly.

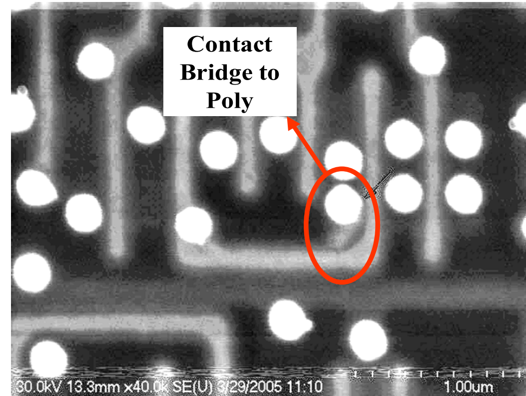
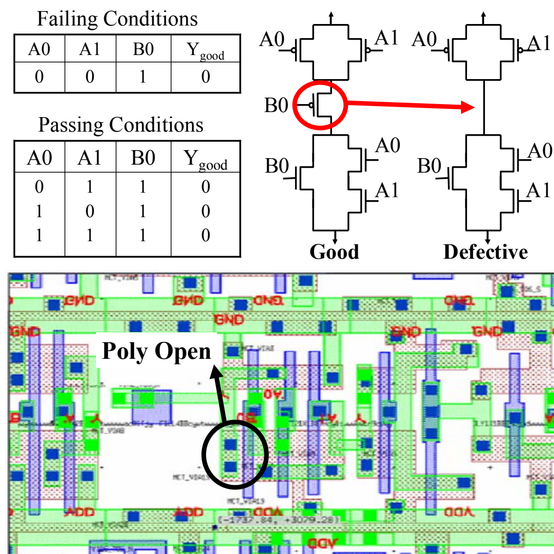




**Figure 11: PFA Image of Case-I Defect.**

**Case-II:** In the second case, the defective cell was an AND-OR-Invert cell whose transistor level schematic is shown in Figure 12.



Failing Conditions

| A0 | A1 | B0 | $Y_{good}$ |
|---|---|---|---|
| 0 | 0 | 1 | 0 |

Passing Conditions

| A0 | A1 | B0 | $Y_{good}$ |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |



**Figure 12: Case-II Defect Site and Transistor Model.**

In this case the defect was a poly open which resulted in a missing connection from input B0 to the corresponding PMOS (Figure 12), with the result that the PMOS was always ON. The failing and passing conditions extracted by our tool for this case are also shown in Figure 12. In

this case it can be easily verified that the extracted failing condition **A0 A1 B0 = 001** is indeed correct, since this leads to a path from power to ground in the defective cell due to the B0 PMOS being always ON. However, when we analyze the extracted passing conditions, two of the conditions **011** and **101**, apparently should also have been failing conditions from a purely switch level simulation point of view. A little more detailed look at the circuit easily explains this anomaly. For the input condition **001** both the PMOS transistors corresponding to A0 and A1 are ON. This creates a stronger pull to Vdd and hence results in the output of the cell being a faulty **1**. The input conditions **011** and **101** both result in only one of the PMOS transistors (corresponding to A0 or A1) being ON. Hence this may result in the output voltage corresponding to a logic value of **0**, the correct value.

## 6. Conclusion

In this paper we present the application of diagnosis techniques that differentiate between cell-internal and interconnect defects to speed-up defect root cause analysis for industrial designs. We identified multiple exercising conditions in test patterns as a major roadblock to cell internal diagnosis for industrial design. We also presented a practical technology, active excitation condition extraction, to correctly determine passing and failing excitation conditions from such pattern. This technique works for designs with compressed patterns. It also handles defects requiring a sequence of values for excitation. Experimental results, both in a controlled simulated environment and on real industrial failing devices, prove the effectiveness and accuracy of our technique and hence its ability to speed up the PFA process, and the overall yield loss factor analysis. Due to the success of this technique on 90 nm, it is now being used on another design using TSMC's 65 nm technology.

Finally we observe that the application of active excitation condition extraction is not only limited to cell internal defects, but can also be directly applied to other types of defects such as interconnect opens and bridges where the defect excitation may depend on the neighboring net values [12].

## References


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

[2] J. A. Waicukauski and E. Lindbloom, "Failure Diagnosis of Structured VLSI", in *IEEE Design and Test of Computers*, Aug. 1989, pp. 49-60.

[3] T. Bartenstein, D. Heaberlin, L.Huisman and D. Sliwinski, "Diagnosing Combinational Logic Designs using the Single Location At-A-Time(SLAT) Paradigm," in *Proc. Intl. Test Conf.,* 2001, pp. 287-296.

[4] S. Venkataraman and S. B. Drummonds, "POIROT: A Logic Fault Diagnosis Tool and Its Applications", *International Test Conference*, 2000. pp. 253-262.

[5] W. Zou, W.-T. Cheng, S. M. Reddy, and H. Tang, "On Methods to Improve Location Based Logic Diagnosis," in *Proc. VLSI Design*, 2006, pp. 181-187.

[6] X. Fan, W. Moore, C. Hora, M. Konijnenburg and G. Gronthoud, "A Gate-Level Method for Transistor-Level Bridging Fault Diagnosis," in *Proc. VLSI Test Symp.*, 2006.

[7] X. Fan, W. Moore, C. Hora and G. Gronthoud, "A Novel Stuck-at Based Method for Transistor Stuck-Open Fault Diagnosis," in *Proc. Intl. Test Conf.*, 2005, pp. 253-262.

[8] R. D. Blanton, J. T. Chen, R. Desineni, K. N. Dwarakanath, W. Maly and T. J. Vogels, "Fault Tuples in Diagnosis of Deep-Submicron Circuits," in *Proc. Intl. Test Conf.*, 2002, pp. 233-241.

[9] I. Pomeranz, S. Venkataraman, S. M. Reddy and E. Amyeen, "Defect Diagnosis Based on Pattern-Dependent Stuck-At Faults," in *Proc. VLSI Design*, 2004.

[10] Y. Higami, K. Saluja, H. Takahashi, S. Kobayashi and Y. Takamatsu, "Diagnosis of Transistor Shorts in Logic Test Environment," in *Proc. Asian Test Symp.,* 2005.

[11] E. Amyeen , D. Nayak and S. Venkataraman, "Improving Precision Using Mixed-level Fault Diagnosis," in *Proc. Intl. Test Conf.*, 2006.

[12] R. Desineni, O. Poku and R. D. Blanton, "A Logic Diagnosis Methodology for Improved Localization and Extraction of Accurate Defect Behavior," in *Proc. Intl. Test Conf.*, 2006.

[13] D. B. Lavo, I. Hartanto and T. Larrabee, "Multiplets, Models and the Search for Meaning: Improving Per-Test Fault Diagnosis, in *Proc. of Intl. Test Conf.*, 2002, pp. 250-259.

[14] J. Rajski, J. Tyszer, M. Kassab and N. Mukherjee, "Embedded Deterministic Test", in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. Vol. 23, Issue 5, May 2004, pp. 776 – 792.

[15] Yu Huang, Wu-Tung Cheng and J. Rajski, "Compressed Pattern Diagnosis for Scan Chain Failures", in *Proc. Intl. Test Conf.*, 2005.

[16] A. Leininger, M. Goessel, P. Muhmenthaler, "Scan chain diagnosis using IDDQ current measurement", in *Proc. of Design Automation and Test in Europe*, 2004, pp. 1302-1307.

[17] Ruifeng Guo and S. Venkataraman, "An Algorithmic Technique for Diagnosis of Faulty Scan Chains", in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. Vol. 25, Issue 9, Sept. 2006, pp. 1861 – 1868.

[18] X. Wen, H. Tamamoto, K. K. Saluja and K. Kinoshita, "Fault Diagnosis for Static CMOS Circuits", in *Proc. Asian Test Symposium*, 1997, pp. 282-287.