Efficient and Accurate Method for Intra-gate Defect Diagnoses in Nanometer Technology and Volume Data

Aymen Ladhar[1,2], Mohamed Masmoudi[2], Laroussi Bouzaida[1]
[1]STMicroelectronics
[2]Electronics, Micro-technology and Communication Research Group, Tunisia
[1] {aymen.ladhar@st.com}

Abstract
Improving diagnosis resolution becomes very important in nanometer technology. Nowadays, defects are affecting gate and transistor level. In this paper, we present a new method to volume diagnosis intra-gate defects affecting standard cell Integrated Circuits (ICs). Our method can identify the cause of failure of different intra-gate defects such as bridge, open and resistive-open defects. Our method gives accurate results since it is based on the use of physical information extracted from library cells layout. Our method can also locate intra-gate defects in presence of multiple faults. Experimental results show the efficiency of our approach to isolate injected defects on industrial designs.

1. Introduction

Fault diagnosis is the process of isolating the source of failure in a defective circuit, so that a physical failure analysis can be performed to physically examine the root cause of failure. Precise diagnosis of different defects affecting chips helps the IC manufacturers to fix the process problems and improve the yield leading to a low cost and shorter time-to-market. The existing logic diagnosis tools [1][2][3][4] can determine, by analyzing the failure's responses, the most likely locations 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 or inside the library cell associated with the identified location.

Previous works on intra-gate defects diagnosis can be put in two general categories. The first one [5][6][7] used complex transformations to perform intra-gate diagnosis. The main drawbacks of these techniques are the dependency on specific transformation for each defect model for diagnosis and diagnosing time needed if we target a volume diagnosis. This is risky since the non-modeled defects may go undiagnosed. The second category of intra-gate diagnosis techniques [8][9][10][14], also known as excitation condition based diagnosis, is based on the realistic assumption that the excitation of a defect inside a cell will be highly correlated to the logic values at the input pins of the cell. For this category, a fault dictionary, where the fault signature of all defects affecting cells library, is used. These techniques are more efficient for volume diagnosis and give more accurate results. However, the previous works that used this technique have some limitations. In fact, no solutions have been proposed in [3] [9] when the defect site is being exercised multiple times in different ways during the capture phase of test pattern. These patterns are referred in [11] as the multiple exercising conditions patterns. Other limitations of all other published works are the assumption of single defect on failing die associated on the non use of physical information to build the fault dictionary mainly for defect caused by defective contacts. In fact, the later defect disconnects in general more than one transistor from an internal connection. Only the knowledge of contact locations, net and transistors that are connected by each contact, can give a realistic and precise simulation results for the fault dictionary creation.

In this paper, we propose a new methodology that delivers precise and realistic transistor level diagnosis results for volume data, the name of the shorted lines in case of intra-gate bridging fault and the names of the disconnecting transistors in case of contact open or resistive-open defects, are presented in diagnosis results. The algorithm proposed in this work can diagnose intra-gate defects in presence of multiple exercising conditions per pattern. Our algorithm can also diagnose intra-gate defect in presence of multiple defects [12].

The rest of the paper is organized as follow. Section 2 describes how we proceed to locate intra-gate defects from diagnosis results. Section 3 explains how we proceed to extract potential intra-gate defects and how to construct the fault dictionary. In section 4 we present our algorithm to determine excitation inputs for multiple exercising condition patterns. Section 5 shows in details our methodology for volume diagnosis of intra-gate defects. Experimental results are presented and discussed in section 6.

2. Localizing potential intra-gate defect from inter-grate diagnosis results

2.1. Intra-gate defects and diagnostic counts

The stuck-at fault model is the most used fault model in diagnosis. Generally, it is followed by a ranking mechanism to see how close it is to the real defect behavior [3]. The use of these counts can be extended to locate suspect cells with intra-gate defects [7].

978-3-9810801-5-5/DATE09 © 2009 EDAA

Consider the example in figure 1, where two faults were injected. The first one on the interconnect wire between gates *C2* and *C1*, the second at transistor level inside gate *C3*. Table 1 represents the response of this circuit in presence of these two faults.

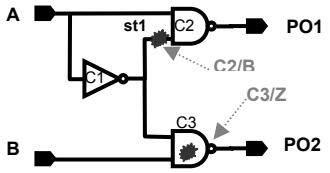

Figure 1: An intra-gate defect in presence of st fault

The inter-gate diagnosis procedure can determine two disjoint defective locations that are not interacting on any primary outputs [12]. The first defect can be modeled by a stuck-at 1 fault. However, the second one cannot be modeled by any inter-gate fault model. The simulation of stuck-at faults (st-1 and st-0), on the output of the faulty location *C3,* shows that the combination of these two fault models can explain all the failures caused by this defect. However, several simulation results don't predict the observed responses. In fact, the simulation of a st-0 on the output of *C3* drives PO2 to 0 for the input value (AB=10). However, the test's output PO2 is equal to 1. Here, we observe the miss-prediction between the simulation results and the observed test responses.

Table 1: Truth table of the defective circuit

	A	B	Tester outputs		Sim (C3/Z st-1)		Sim (C3/Z st-0)	
			PO1	PO2	PO1	PO2	PO1	PO2
0	0	0	1	1/0	1	1	1	1/0
1	0	1	1/0	0/1	1	0/1	1	0
2	1	0	1/0	1	1	1	1	1/0
3	1	1	1	1/0	1	1	1	1/0

Generally, when an intra-gate defect affects a circuit, the simulation of stuck-at fault or combination of stuck-at fault, will explain all its faulty responses, but some simulation results will not match the tester fails.

Figure 2 shows the relation between the Simulation-Fails (*SF*) obtained while simulating the circuit with an injected fault and the Tester-Fails (*TF*) from the tester. The relationship between the two sets is captured as diagnostic counts. The failing observe points which are common between *TFs* and are called Tester Fails-Simulation Fails (*TFSF*). The observe points, which only fail during simulation are called Tester Pass-Simulation Fails (*TPSF*). On the contrary, the observe points, which only fail on the tester, are called Tester Fails-Simulation Pass (*TFSP*).

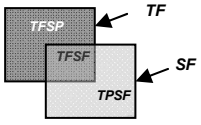

Figure 2: Relation between TF and SF

Consider the same sample in figure 1. The tester fails (1) contains four failing patterns and five failing elements. A failing pattern is defined as a pattern that detects the failures associated with all the primary outputs where the failures were observed. However, a failing element is defined as a failing pattern associated with only one primary output where the failure is observed. The per turn diagnosis applied on this tester fails (1) shows that three of the four failing patterns which constitute this TF are SLAT patterns [13]. Patterns 0 and 3 are explained by the faulty candidate *C3/Z,* pattern 2 is explained by the faulty candidate *C2/B*. However, pattern 1 is a Non SLAT failing patterns that can be explained only by the combination of the two faults *C3/Z* and *C2/B*. The simulation of a st-1 *(2)* and st-0 *(3)* on the output of the failing gate *C3* shows that the three fails caused by this defect on the primary output PO2 are explained. However the simulation of st-0 on the output of *C3* shows that {2/PO2} is a passing element belonging to TPSF category. Table 2 shows the different diagnostic counts for the considered example. As we can observe there is two disjoint cones. Consequently, candidates belonging to each one are not interacting for failing and passing patterns. So, each candidate in each cone can have its own diagnostic counts.

TF: {0/PO2, 1/PO1, 1/PO2, 2/PO1, 3/PO2} *(1)*
SF_{st1} (*C3/Z*): {1/PO2} *(2)*
SF_{st0} (*C3/Z*): {0/PO2, 2/PO2, 3/PO2} *(3)*

Table 2: Diagnostic counts

Pin name	model	#TFSF	#TFSP	#TPSF
C3/Z	St0	2	1	1
C3/Z	St1	1	2	0
C2/B	St1	2	0	0

An intra-gate defect is diagnosed if all the extracted excitation values on the inputs of the defective cell for *its failing and passing patterns* are explained by a candidate belonging to the fault dictionary.

2.2. Identification of potential intra-gate defects from stuck-at fault diagnosis results

To minimize the work space and save computing time, it is important to select the minimum number of cell candidates to be considered in the intra-gate diagnosis algorithm. From the stuck-at diagnosis results, we assign a new $TFSF_{st01}$ counts for each failing primary output. This count is the result of adding $TFSF_{st1}$ and $TFSF_{st0}$ counts. Once this step is performed, we select from each defective cone all the candidates having the biggest $TFSF_{st01}$ counts. If more than one candidate exists we select those having the two lesser TPSF counts as shown in figure 3. In fact, experiments on different cells show that:

- When a pattern-dependant defect exists the $TPSF_{st01}$ counts will be minimum on the gate's input. Since, the defect can be modeled by a stuck-at fault when the previous pattern excites the defect.
- When an intra-gate bridging fault exists and no stuck at fault on the inputs of the defective cell explains the

failures, we found that the $TPSF_{st01}$ counts will be lesser on the gate's output.

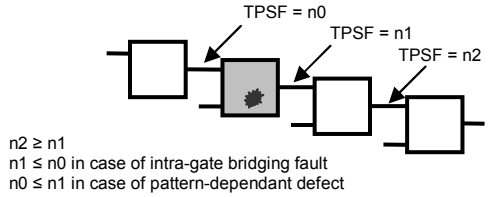

Figure 3: The TPSF counts

3. Fault dictionary

Intra-gate defects can be classified in 2 main categories upon their location on the defective cell. The first category affects neighbored lines causing bridging faults. Whereas, the second category affects contacts causing open and resistive open defects. Our diagnosis methodology is based on the use of a fault dictionary containing all the fault signatures. Figure 4 gives an overview of the proposed method to extract and simulate intra-gate defects. It begins by extracting all the neighbored lines and disconnecting transistors by defective contacts. Then, all these potential defects are simulated and their faulty behaviors are recorded in a fault dictionary.

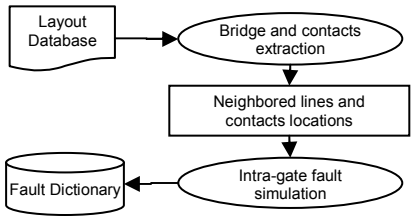

Figure 4: Fault dictionary construction

3.1. Potential intra-gate defect extraction

The method used to identify the coordinates of the neighbored lines and contact is based on the use of modified DRC rule file. The minimum metal (1), poly-silicon (2) distances and contact width (3) were changed to new DFM values. Once all the coordinates of these potential defects were identified, we determine their corresponding net names.

```
METAL_SPACING {                    (1)
external M1i < DFM_M1
}
POLY_SPACING {                     (2)
external POi < DFM_PO
}
Contact_WIDH {                     (3)
internal COi < DFM_CO
}
```

Figure 5 shows the overall view of the method used to extract potential intra-gate defect as well as the different tools used to perform this extraction. The calibre DRC is used to extract the coordinates of neighbored lines and all contacts locations. For this, we modified the DRC SVRF rule file [15]. Then, the calibre *query server* is used to determine from the (X, Y) coordinates of each potential defect the net names corresponding and shapes making up each net in the transistor netlist. Finally, we collect the net names of all the neighbored lines as well as the name of all transistors and net linking by each contact.

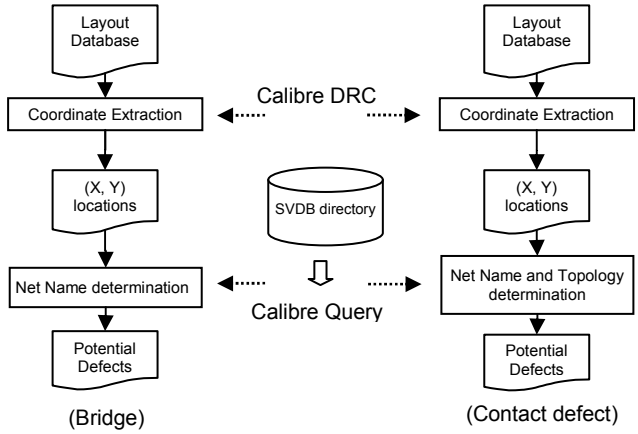

Figure 5: Potential defect extraction

3.2. Intra-gate defect simulation

Once all the potential defects are extracted from the cell's layout, we perform a *post layout simulation* on each suspect defect. All the fault signatures are then recorded on a fault dictionary.

3.2.1. Intra-gate bridging fault

Intra-gate bridging fault results from shorting two nets or more inside a cell. To simulate a bridge between two neighbored lines we suppose that they are linked by a parasitic resistance of 1Ω. The simulation of different intra-gate bridging faults in different cells shows that:

- Some intra-gate bridging faults have the same effects then a stuck-at fault on one of the primary inputs or outputs of the target gate.
- Some intra-gate bridging faults have no effect on the gate's primary outputs and the gate is kept faulty-free.
- Some intra-gate bridging faults drive some of the gate's primary outputs to the opposite value for some excitation values and keep the rest faulty free. The simulation of stuck-at fault on cell's outputs shows that the TPSF counts is different from 0.

3.2.2. Intra-gate open defect

In this work we focus on intra-gate open defects caused by defective contacts. They can be classified in two main classes upon their location on circuit.

- Source-drain opens: they are also known as stuck-open defects. The simulation of open source or drain faults shows that it can cause sequential behavior and thus require a certain sequence of patterns in order to be detected. This sequence of pattern can be composed of two patterns or more. In fact, experiments show that it may occur during test that more then one pattern put the gate's output on high impedance value. To simulate this defect we replace each contact linking metal layer to diffusion layer by a parasitic resistance of 1GΩ.

- Gate opens: they disconnect one transistor gate or more. The voltage of the floating gate depends on the neighbored lines and parasitic capacitances. In the simulation, we consider all parasitic capacitances and resistances. To simulate this defect we replace each contact linking metal layer to ploy-silicon layer by a floating node.

3.2.3. Intra-gate resistive open

A resistive-open defect is defined as an imperfect circuit connection that can be modeled as a defective resistor between the circuit nodes that should be connected. In this work, we focus in intra-gate open defects caused by contacts. Simulation results show that this defect can depend in some cases on the previous patterns applied at the gate's inputs. In the simulation we consider all the parasitic capacitances and resistances. The value of the parasitic resistance was 0.5, 1 and 5 MΩ.

4. Determining excitation conditions from patterns with multiple exercising conditions

The exercising conditions are defined as binary logic value combinations that are applied on the input pins of a library cell in the design during the capture phase of a test pattern. To diagnose intra-gate defects these exercising conditions on the input of each candidate cell are determined for both failing and observable passing patterns. However, determining the active excitation conditions values from an industrial design is not a trivial task. In fact, it may occur that the intra-gate defect is being exercised multiple times in different ways during the capture phase of a test pattern causing multiple exercising conditions [11]. This is due to various reasons like multiple capture cycles, the presence of both leading and trailing edge flops in the design etc.

The following algorithm presents our method to determine active excitation conditions from patterns with multiple exercising conditions.

Step1: Collect the exercising conditions on the inputs and outputs of each intra-gate candidate cell.

Step2: Divide the exercising conditions values belonging to *failing patterns* into two sub-categories:

- The first one SEC_F (Single Exercising Conditions) containing only excitation conditions for the failing patterns with single exercising conditions.
- The second one MEC_F (Multiple Exercising Conditions) containing excitation conditions for the failing patterns with multiple exercising conditions.

Step3: Divide the exercising conditions values belonging to *passing patterns* into two sub-categories:

- The first one SEC_P (Single Exercising Conditions) containing only excitation conditions for the passing patterns with single exercising conditions.
- The second one MEC_P (Multiple Exercising Conditions) containing excitation conditions for the passing patterns with multiple exercising conditions.

Step 4: Remove SEC_F from MEC_P and remove SEC_P from MEC_F, then repeat step 2 and 3 until no SEC_F and no SEC_P exist respectively in MEC_P and MEC_F.

Step 5: From the fault dictionary find the fault locations matching all the SEC_F and all the SEC_P

Step 6: From the list obtained in step 5 find the candidates that explain one of each MEC_P and one of each MEC_F, then rank the candidates according the number of excitation pattern used. The candidate that explains these criteria with minimum input excitation values is more likely to be the good candidate.

For sequence dependant defect, it is mandatory to introduce the previous values in the multiple exercising conditions algorithm.

Consider an example where a NAND gate has been identified as a potential intra-gate defect. The stuck-at fault simulation on the output of this gate shows that patterns P1, P2 and P3 belong to category of failing patterns, and patterns P4 and P5 are passing patterns.

Input excitation values for P1: {00} → failing pattern
Input excitation values for P2: {10, 11} → failing pattern
Input excitation values for P3: {10, 00} → failing pattern
Input excitation values for P4: {10, 00} → passing pattern
Input excitation values for P5: {10, 11} → passing pattern

So, for our example, the algorithm will produce the following results: After step 2 and 3, the algorithm will consider pattern P1 as SEC_F, so {00} is a failing excitation condition. Pattern P2, P3 are considered as MEC_F, P4 and P5 as MEC_P. In step 4 {10} is considered as a passing excitation condition and {11} as a failing excitation condition. In Step 5 and 6 we use the fault dictionary to find the fault location that explains this failure.

5. Diagnosis flow

Figure 5 shows the proposed flow of the proposed intra-gate diagnosis algorithm. The proposed method begins by sorting the test results as well as the failing patterns. The main objective of doing this step is to reduce the volume diagnosis time, since each failing pattern will be diagnosed only one time in the following steps. Our proposed method use SLAT diagnosis, i.e. each failing pattern is considered as independent diagnosis. For each failing pattern the excitation input conditions are determined for each suspect cell. Once all the data-logs are diagnosed, we select all the suspect cells identified by the volume diagnosis. Then, they are simulated with all test patterns. After this step, the diagnostic counts are computed. The input exercising conditions for observable passing patterns are determined. The multiple exercising conditions algorithm is applied to find out the active excitation input values. Finally, the fault dictionary is used to find out the intra-gate defect that matches the failures.

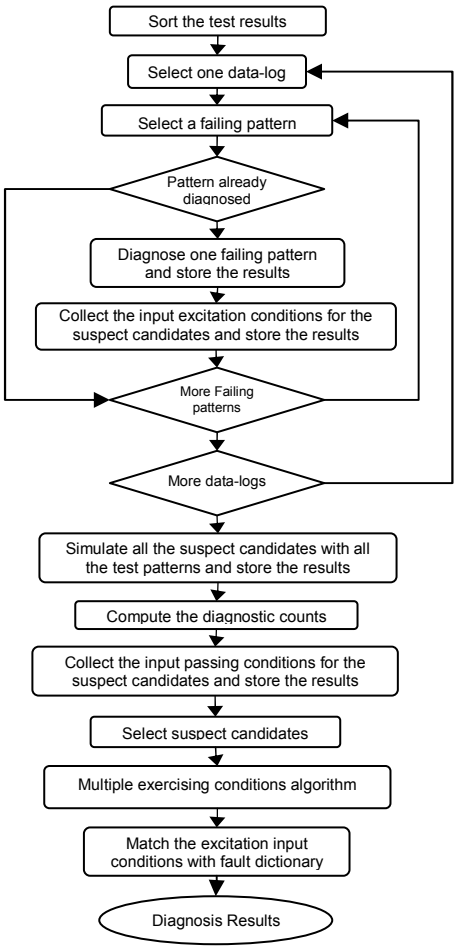

Figure 6: Complete diagnosis flow

6. Experimental results

To verify the effectiveness of the proposed intra-gate diagnosis algorithm developed in this paper, we elaborated controlled experiments in which the behavior of failing chip with an intra-gate fault was simulated. This was performed by injecting intra-gate defects in a list of library cell instances. The method used to simulate intra-gate defects without a transistor level simulator is to replace the target cells by modified ones in the library netlist. This modified cell represented a defective version of the original cell, by having changed its truth table or behavior. The modified netlist was then simulated against stuck-at test patterns, and the failures were recorded in data-log. These data-logs are then used in the volume diagnosis procedure.

Table 3: Circuits statistics

Circuit	Gate count	Gate type count	Pattern #	Scan chains #
A	40	15	14	1
B	382224	777	3131	8

Table 3 shows some characteristics of the two circuits used to verify our intra-gate diagnosis approach. The second column presents the number of gates in each circuit. The third column shows the number of gate counts in each circuit. This number is very small comparing to the gate counts.

6.1. Experiment results for intra-gate bridging faults diagnosis

Results for single intra-gate bridging fault

Table 4 shows the results for conventional stuck-at diagnosis for 6 intra-gate bridging faults injected in different sample of circuit B. Column 1 shows the suspect intra-gate cells, column 2 represents the number of failing elements on each data-log. Column 3 until 6 indicate the stuck at diagnostic counts on each candidate cell's output.

Table 4: Stuck-at faults diagnosis

Gate name	Stuck-at fault diagnosis				
	TF	$TFSF_{st0}$	$TFSF_{st1}$	$TPSF_{st0}$	$TPSF_{st1}$
F_AN2LLP	2267	1948	319	0	713
AO4ALL	854	125	729	500	0
F_MUX21NLL	1265	0	1265	0	68
F_ND4LL	237	205	32	56	159
OR5HS	331	268	63	524	649
AO7LL	524	295	229	1176	367

Table 5 shows the intra-gate diagnosis results obtained by our diagnosis flow on each suspect cell in table 4. Column 2 presents the number of suspect transistors on each candidate cell. Columns 3 until 6 show respectively the SCS_F, MCS_F, SCS_P and MCS_P counts that are used in the multiple exercising conditions algorithm. Column 7 shows the intra-gate bridging fault candidate matching the defect behavior. For example, the cell candidate F_AN2LLP has 6 transistors. A stuck-at fault on its output explains 86 failing patterns and cause 36 passing patterns. The extraction of exercising conditions on the inputs of this cell shows that 16 of the failing patterns are SEC_F however 70 are MEC_F. In the same way, 24 of the passing pattern cause SEC_P however 12 are MEC_P. Using the fault dictionary we found a bridging fault between the gate and drain of transistor M0 explaining this failure.

Table 5: Intra-gate Bridging fault diagnosis

Gate name	Trans	SEC_F	MEC_F	SEC_P	MEC_P	Defect
F_AN2LLP	6	16	70	24	12	Br G-D M0
AO4ALL	10	29	7	3	2	Br 2-7
F_MUX21NLL	12	23	52	6	23	Br 5-2
F_ND4LL	8	12	4	25	9	Br G-D M5
OR5HS	14	22	105	12	23	Br 9-12
AO7LL	6	36	85	29	8	Br G-D M5

Results for multiple-fault

Experiments where multiple faults are injected in circuit A and B are performed. On each sample we injected a stuck-at fault and an intra-gate bridging fault. Table 6 shows the per turn diagnosis results. Each class CL_i represents the equivalent candidates that explain the same failing patterns. For example, fault candidates in class *CL3* explain two failing patterns and no fault candidates in other classes can explain these patterns. The $TPSF_{st01}$ counts are equal to zero for this class; this involve that it is a stuck-at fault. From *CL1* and *CL2* of sample 1, we find a candidate cell that is belonging to these two classes; this fault candidate has 23 passing elements. The intra-gate diagnosis algorithm identifies this candidate as intra-

gate bridging faults, since its behavior matches a defective case in the fault dictionary.

Table 6: Per turn diagnosis results

Sample	Circuit	Fault Class	Per turn diagnosis			
			Failing pattern	SLAT pattern	Non explained pattern	$TPSF_{st01}$
1	A	CL1	36	28	8	23
1	A	CL2	36	6	30	0
1	A	CL3	36	2	34	0
2	A	CL1	14	9	5	0
2	A	CL2	14	5	9	58
3	B	CL1	3	1	1	0
3	B	CL2	3	2	1	986

6.2. Experiment results for intra-gate Open defect diagnosis

Table 7 shows the corresponding diagnosis results for transistor stuck-open faults. For circuit *A*, three stuck-open defects were injected on contacts linking a net to many transistors. Unlike bridging fault, excitation of transistor stuck-open fault depends on previous values. Therefore, exercising conditions were collected for previous and current vectors. In all cases, we obtained the good intra-gate candidate that explains the failures. For example, the AO22X4 cell has been identified as suspected cell for intra-gate defect. In fact, all the TF are explained and some TPSF counts exist. This cell contains 10 transistors and 17 contacts. We were able to identify the defective contact using our diagnosis approach.

Table 7: Intra-gate diagnosis results for open defect

Gate name	Trans.	Stuck-at diagnosis results			Defect
		TFSF	TFSP	TPSF	
XOR2X4	10	4	0	1	CO2
AO1FX2	8	5	0	3	CO5
AO22X4	10	9	0	2	CO4

Figure 7 represents the layout of the gate AO22X4 that has been diagnosed as defective previously. This gate is from the STMicroelectronics 65nm library. The defective contact CO4 disconnects M1, M3 transistors from M2 and M4 transistors, causing multiple stuck-open defects. Our method was able to diagnose this defect since it is based on the use of physical information.

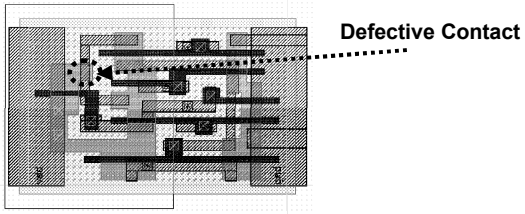

Figure 7: AO22X4's layout

6.3. Experiment results for intra-gate resistive open defect diagnosis

Table 8 shows the corresponding diagnosis results for a NAND2X4 gate with a resistive open defect. The value of the injected resistive open defect is 5 MΩ. Columns 3 to 5 show the stuck-at diagnosis counts, and column 6 shows the defective contact explaining the failures.

Table 8: Intra-gate diagnosis results for resistive open

Gate name	Trans.	Stuck-at diagnosis results			Defect
		$TFSF_{st01}$	$TFSP_{st01}$	$TPSF_{st01}$	
NAND2X4	4	4	0	3	CO1

7. Conclusion

This paper describes a new method to diagnose different types of intra-gate defects. The method is based on the use of fault dictionary. The method reports the location of intra-gate defect in the cell's layout with precision since all the potential defects are extracted from cell's layout then are simulated and recorded in a fault dictionary. So the layout analysis is not mandatory after diagnosis. The method allows also diagnosing intra-gate defects in presence of multiple exercising conditions and multiple defects. Experiments results in controlled simulated environment prove the effectiveness and accuracy of our method in isolating the injected defect.

REFERNCES

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

[2] S. Venkataraman and S. B. Drummonds, "POIROT: A Logic Fault Diagnosis Tool and Its Applications", *in Proc. of Inter. Test conf.* 2000, pp 253-262.

[3] C. Hora et al, "On Electrical Fault Diagnosis tool in full-scan circuits*", in Workshop on Defect Based Testing*, 2001, pp 17-22.

[4] W. Zou et al, "On Methods to Improve Location Based Logic Diagnosis ", *in Proc. VLSI Design, 2006*, pp 181-187.

[5] X. Fan et al, "A Novel Stuck-at Based Method fot Transistor Stuck-Open Fault Diagnosis", in *Proc. of International Test Conference* 2005, paper 16.1

[6] X. Fan et al, "A gate Level Method for Transistor-Level Bridging Fault Diagnosis*" in Proc. Of VLSI Test Symp.* 2006, pp 266-271.

[7] X. Fan et al, "Extending Gate-Level Diagnosis Tools to CMOS Intra-Gate Faults", *in Trans. In Silicon Debug and Diagnosis* 2007 pp 685-693

[8] E. Amyeen and al "Improving Precision Using Mixed Level Fault Diagnosis" *in Proc. of Inter. Test conf.* 2006 paper 22.3

[9] J. Li and E. J. McCluskey "Diagnosis of Resistive-Open and Stuck-Open Defects in Digital CMOS ICs" *in Trans. On Computer aided Design*, 2005 pp 1748-1759.

[10] R. Desineni et al "A Logic Diagnosis Methodology for Improved localization and Extraction of Accurate defect Behavior" *in Proc. of Inter. Test conf.* 2006.

[11] M. Sharma et al, "Faster Defect localization in Nanometer Technology based on Defective Cell Diagnosis" *in Proc. of Inter. Test conf.* 2007 paper 15.3

[12] Z. Wang et al "An Efficient and Effective Methodology on the Multiple Fault Diagnosis" *in Proc. of Inter. Test conf.* 2003 pp. 329-338

[13] T. Bartenstein, D. Heaberlin, L. Huisman, and D. Sliwinski, "Diagnosing combinational logic design using the single location at-a-time (slat) paradigm," in *in Proc. of Inter. Test conf.* 2001, pp. 287–296.

[14] J. C.-M. Li and E. J. McCluskey, "Diagnosis for sequence dependent chips," *in Proc. VLSI Test Symp*., Monterey, CA, 2002, pp. 187–192.

[15] Mentor graphics documentation, "Standard Verification Rule Format (SVRF) Manual", *Calibre v2007.4*