Improving Cell-Aware Test for Intra-Cell Short Defects

Dong-Zhen Lee*†, Ying-Yen Chen*, Kai-Chiang Wu‡ and Mango C.-T. Chao†

* Realtek Semiconductor Corporation, Taiwan
† Institute of Electronics, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
‡ Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

***Abstract*—Conventional fault models define their faulty behavior at the IO ports of standard cells with simple rules of fault activation and fault propagation. However, there still exist some defects inside a cell (intra-cell) that cannot be effectively detected by the test patterns of conventional fault models and hence become a source of DPPM. In order to further increase the defect coverage, many research works have been conducted to study the fault models resulting from different types of intra-cell defects, by SPICE-simulating each targeted defect with its equivalent circuit-level defect model. In this paper, we propose to improve cell-aware (CA) test methodology by concentrating on intra-cell bridging faults due to short defects inside standard cells. The faults extracted are based on examining the actual physical proximity of polygons in the layout of a cell, and are thus more realistic and reasonable than those (faults) determined by RC extraction. Experimental results on a set of industrial designs show that the proposed methodology can indeed improve the test quality of intra-cell bridging faults. On average, 0.36% and 0.47% increases in fault coverage can be obtained for 1-time-frame and 2-time-frame CA tests, respectively. In addition to short defects between two metal polygons, short defects among three metal polygons are also considered in our methodology for another 9.33% improvement in fault coverage.**

I. Introduction

Cell-aware (*CA*) test is a defect-based test methodology developed by Mentor Graphics [1] – [5], which explicitly SPICE-simulates the behavior of each targeted intra-cell defect for a standard cell and models the faulty behavior at the IO ports of the cell with a digitalized format that can be loaded into a commercial ATPG for test generation. Compared to conventional stuck-at (*SA*) or transition (*TD*) faults defining their faulty behavior at a IO port of a cell, cell-aware faults can represent a more complicated faulty behavior caused by a real defect for a cell and hence can help to cover the defects missed by the conventional fault models. The effectiveness of CA test was then validated through a series of experiments on real IC products, showing that CA tests targeting 1-time-frame (1tf) or 2-time-frame (2tf) CA faults can indeed help to capture extra defective parts escaped from the conventional test sets for SA fault, TD fault or multiple detection of SA or TD fault.

Extending the idea of CA test, some variants of CA test have been proposed to further improve the test quality. In [6], a dual-cell-aware model was proposed to cover the short defects in between two adjacent cells. In [7], design-for-manufacturing rules were utilized to extract intra-cell and inter-cell CA faults. In [8], the probability of a defect occurring was evaluated through a unified metric, which guided the test generation of CA test to cover most defects in a statistical way. In [9], small-delay 2tf CA faults were identified on a cell-type basis considering defective voltage variation and then test-generated with timing-aware ATPG that propagates a 2tf CA fault through the longest path possible. In [10], small-delay 2tf CA faults were identified on an instance basis considering the timing-analysis result of a cell and the extra delay caused by a defect. Many research works was also proposed to speed up the test generation or the fault extraction for CA tests [11] [12] [13] [14].

In Mentor Graphics' solution for CA test [1] [2], four types of intra-cell defects, including short, open, transistor stuck-on and transistor stuck-open, are considered and injected for extracting CA faults from each standard cell in the targeted libraries. For the CA faults resulting from short defects, solutions in [1] [2] considered a coupling capacitance in the SPICE netlist extracted from the layout of a targeted cell as a potential location to have a short defect. The idea behind this is that an extracted coupling capacitance between two nodes usually implies that the two corresponding metal lines are physically close and have parallel run in between. This RC-extraction-based method of finding potential locations of short defects is also used as the default setting of Mentor Graphics' tool for CA fault extraction, CellModelGen [17]. However, an extracted coupling capacitance may not be able to correspond to a physical short defect, and vice versa. Moreover, a coupling capacitance can only reflect whether two metal lines are close to each other, and hence cannot be used to identify the short defects involving three metal lines. In fact, layout analysis is a more direct way to identify locations for injecting realistic short defects, which has been suggested by many previous works regarding intra-cell short defects [15] [16].

In this paper, we develop a framework to extract CA faults associated with intra-cell short defects. Our proposed framework considers the intra-cell short defects locating between two nodes as well as those locating among three nodes. The locations for short defects are identified with a layout analysis on the targeted cells. The extracted CA faults are output into the UDFM format, which can be read in by Mentor Graphics' Tessent [18] for test-pattern generation. The experimental results based on industrial designs demonstrate that our layout-based method can identify some realistic short defects missed by RC-extraction-based method while excluding some impractical short defects identified by RC-extraction-based method. The experimental results also demonstrate that some

978-3-9819263-6-1/DATE22/©2022 EDAA

CA faults resulting from 3-metal shorts cannot be covered by the test patterns generated for the CA faults resulting from 2-metal shorts. The gain of coverage of 2tf 3-metal-short CA faults achieved by its dedicated top-off test patterns is 9.33% on average over 5 industrial designs.

II. Layout-Based CA Test Framework for Intra-Cell Short Defects

We propose to improve CA test methodology by addressing the following concerns:

- The framework based on RC extraction (RCXT) identifies potential defects by checking coupling capacitance in the extracted RC netlist. Nevertheless, identifying defects based on RC extraction lacks layout awareness and, as exemplified in Sec. II-B and Sec. II-C, is sometimes not realistic.
- The lack of layout awareness implies disregarding the actual physical proximity of polygons in the layout. The physical proximity of polygons may help determine whether a potential defect should be considered or not. Without taking advantage of the proximity information, there exist missed defects which do have to be considered in practice.

A. *Proposed layout-based flow*

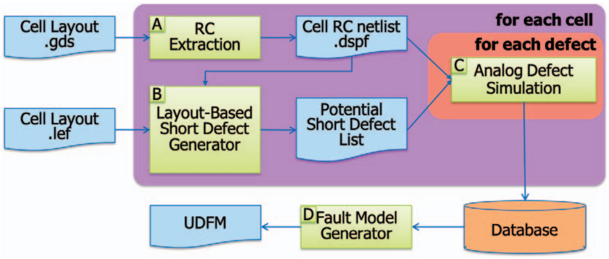


Fig. 1: Proposed flow

TABLE I: Illustration of intra-cell short defect types

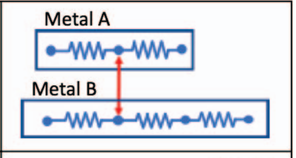
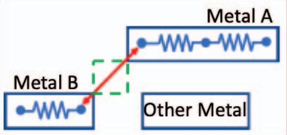
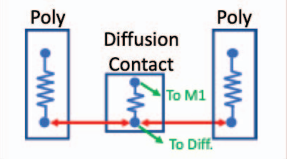
side-to-side (S2S)	
corner-to-corner (C2C)	
contact-to-poly (C2P)	

Fig. 1 depicts the overall flow of our proposed layout-based framework. Steps A and B are performed for each valid cell, while step C is performed for each potential defect.

Step A (RC Extraction): By running commercial tool Calibre [19], the parasitic resistance and capacitance of each standard cell are extracted from the cell layout and its cell schematic netlist. The coordinate of each resistance is also recorded in the obtained RC netlist so as to precisely inject potential short defects.

Step B (Layout-Based Short Defect Generator): By analyzing both the geometry of cell layout information (in .lef file) and the coordinate of each resistance (in .dspf file), potential layout-based intra-cell short defects can be found. Various types of short defects between two nodes (see Table I, denoted by red arrows) in the extracted RC netlist are introduced and will be detailed later.

Step C (Analog Defect Simulation): The potential short defects found by step B are injected into the golden RC netlist obtained by step A to create a faulty netlist. HSPICE [20] simulates the faulty netlist, and the simulation result of each potential short defect is recorded in the database.

Step D (Fault Model Generator): According to the simulation results, the 1-time-frame (1tf) and 2-time-frame (2tf) fault models are generated, and the corresponding UDFM can be produced for CA test.

Table I illustrates how our layout-based method identifies potential short defects inside a cell. Our intra-cell short defects can be classified into three types: side-to-side (S2S), corner-to-corner (C2C) and contact-to-poly (C2P), as described in the sequel:

- Side-to-Side (S2S): Given cell layout information, two parallel rectangles on the same metal layer are extracted if (*i*) there is no other rectangle between them and (*ii*) the distance between them is less than 5X minimum spacing of design rule constraints, denoted by 5DRC. According to the coordinates recorded in the .dspf file, the closest two nodes are chosen as the sites for defect injection, where an extra resistor will be added.
- Corner-to-Corner (C2C): As shown in the second row of Table I, two corner-to-corner rectangles on the same layer are extracted if (*i*) there does not exist any rectangle in the dashed green box formed by the two corners and (*ii*) the distance between them is less than 5DRC. For defect injection, an extra resistor will be added between the two nodes associated with two corners.
- Contact-to-Poly (C2P): A potential short defect may be located between a diffusion contact and a poly, and this type of short defect should also be considered. As shown in the third row of Table I, the site for defect injection on the diffusion contact is the node connecting to diffusion; the other site will be the nearest node on a particular poly. Note that there exist multiple C2P defects given a diffusion contact.

B. *Generation of two-metal-short (2MS) defects*

In this subsection, we will show real examples of short defect generation which differentiate our layout-based framework from RCXT-based methodology used in mainstream commercial tools.

Fig. 2 shows a case study of the generation of a C2C short defect, between the red and pink rectangles. Such a defect

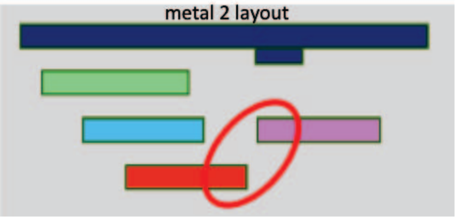


Fig. 2: Example of a corner-to-corner 2MS defect

cannot be identified by RCXT-based methodology because RCXT-based methodology relies on seeking parallel-running segments between two rectangles but in this case, the red and pink rectangles are disjoint on their running (i.e., horizontal) direction. It can be clearly noted that disregarding C2C short defects implies missing some potential defects which may really occur. In this work, based on examining the physical proximity of rectangles in the layout, C2C short defects will not be missed and the associated faults will likely be covered in the subsequent stages of ATPG and fault simulation.

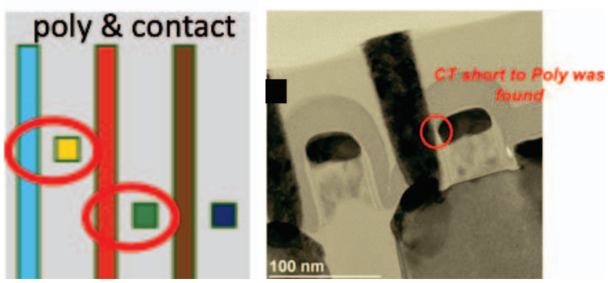


Fig. 3: Example and photomicrograph of a contact-to-poly 2MS defect

A case study of the generation of C2P short defects is shown in Fig. 3, where in the left sub-figure the squares are diffusion contacts and the vertical rectangles are polys; the photomicrograph of a C2P short defect can be found in the right sub-figure. When applying RCXT-based methodology, diffusion contacts and polys are considered to be on the different layers and cross-layer shorts are not taken into account. Therefore, C2P short defects are naturally missed by RCXT-based methodology. In this work, we set up "layer mapping" to correctly recognize same-layer rectangles according to the reality instead of the labeling, such that diffusion contacts and polys are likely to short to each other.

Fig. 4 shows a case study of the generation of S2S short defects. The light blue square (indicated by the dotted blue arrow's lower terminal) associated with signal "B2" is a piece of metal straight located between its upper via and lower via. Such a metal polygon is basically ignored during RC extraction and thus RCXT-based methodology is not able to generate the short defect indicated by the dotted blue arrow, whereas our layout-based framework is able to.

Fig. 4 also shows an example of an unreasonable short defect, as indicated by the solid red arrow. Due to the existence of parallel run between the light green (signal "B3") and purple (signal "O") rectangles, RCXT-based methodology will accordingly generate a S2S short defect despite insignificant parallel run and relatively long distance between them. In our work, the generation of such a S2S short defect is determined by whether the parallel run is longer than the distance. The defect indicated by the solid red arrow apparently does not satisfy the criterion and thus will not be generated. The rationale behind the criterion is that: before the defect between signals "B3" and "O" is generated, there would be other short defects to be generated, such as those between signals "N_5" and "B2", signals "N_5" and "B1", signals "VCC" and "N_4", etc. Given a pair of rectangles, if their distance is longer the parallel run, the S2S short defect between them should be prevailed by many other defects "in between" the pair of long-distance rectangles.

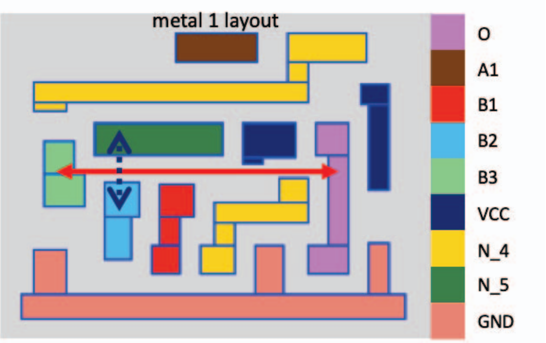


Fig. 4: Example of a side-to-side 2MS defect

C. *Considering three-metal-short (3MS) defects*

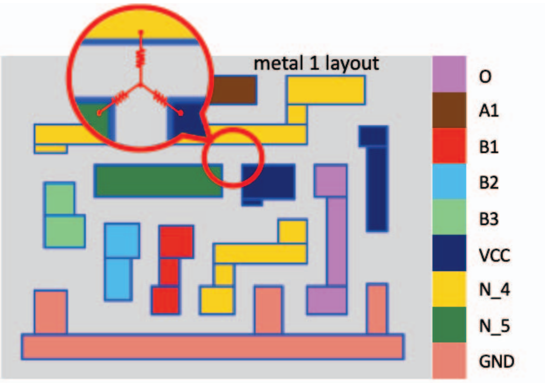
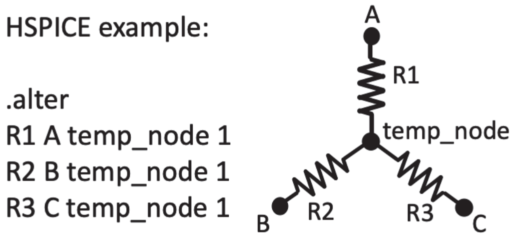


Fig. 5: Example of a 3MS defect

In addition to short defects between two different signals, short defects lying on three different signals (see Fig. 5) are also highly possible. Short defects among three signals are called three-metal-short (3MS) defects. To consider 3MS defects in our framework, we model a 3MS defect by three resistors connected as a "Y" (tripod) shape instead of "Δ" shape. The "Y" shape is more like a real defect located among the three rectangles. The central point represents the center of the defect and each resistor refers to the resistance between the defect and the corresponding rectangle. In this paper, we assume that the three resistance values are identical when modeling a specific 3MS defect. This assumption can simplify the complexity of calculating equivalent resistance values and quickly obtain an estimation.



```
HSPICE example:

.alter
R1 A temp_node 1
R2 B temp_node 1
R3 C temp_node 1
```

Fig. 6: 3MS defect injection in HSPICE

When injecting 3MS defects in HSPICE, an "alter" command is used as shown in Fig. 6. A newly added node called "temp_node" will be regarded as the center of the defect, and it will connect to three nodes, A, B and C shown in Fig. 6, corresponding to three rectangles in the layout. In the HSPICE script, the three resistors are created to model the 3MS defect and the resistance values are all assumed to 1Ω in this paper.

Finally, the proposed layout-based short defect generator, mainly comprising the generation of 2MS and 3MS defects, is summarized in Algorithm 1.

III. Experimental Results

The experiments were conducted on a set of four industrial designs. Both of our layout-based framework and RCXT-based methodology were used for comparison purpose. Table II shows the statistics of runtime for fault extraction using three different methods: RXCT-based (row 3), layout-based 2MS (row 4) and layout-based 3MS (row 5). Columns 2 to 5 show the runtimes for RC extraction, defect generation, analog simulation and fault model generation, respectively, corresponding to steps A to D as illustrated in Fig. 1. In the following subsections, detailed analysis of fault extraction and also the results of ATPG will be demonstrated.

A. Comparison of fault extraction

Table III shows the results of fault extraction of layout-based 2MS defects. For 1tf test, a total of 15,460 faults can be extracted; among which 89.0% are equivalent to the faults extracted by RCXT-based methodology and 95.2% are dominated by those. For 2tf test, among a total of 1,291 faults, 87.4% are equivalent and 89.9% are dominated. The results reveal basic competence of our framework against RCXT-based methodology, due to the large percentages of fault equivalence and fault dominance. The remaining percentages of 11.0% and 12.6% are those faults extracted by our framework but not by RCXT-based methodology. Such faults will determine the superiority of our framework, as demonstrated in the next subsection.

Table IV shows the results of fault extraction of layout-based 3MS defects. For 1tf test, a total of 8,688 faults can be extracted; among which 73.2% are equivalent to the faults extracted based on our 2MS defects and 98.5% are dominated by those. For 2tf test, among a total of 1,270 faults, 6.9% are equivalent and 15.0% are dominated. The results show that, for 1tf test, the difference in the percentage of fault equivalence between the 2MS and 3MS cases is not negligible, necessitating the consideration of 3MS defects. For 2tf test, the necessity of considering 3MS defects is much more crucial due to very low fault equivalence.

Table IV shows the detailed cell type distribution of 3MS non-equivalent faults toward the 2MS faults. One can note that more complex cells have more non-equivalent fault counts such as MUX, AO and OA. This is because complex cells have larger numbers of test conditions, and just one different test condition can cause a fault to become non-equivalent. If a design uses many complex cells, we can expect that the 3MS fault model can have a higher opportunity to detect a fault which is not detected by the 2MS fault model.

Algorithm 1: Layout-based short defect generation

```
Input: A library of standard cells
Output: A list of potential short defects
for each cell $C$ in the library do
    for each candidate pair of rectangles $(A, B)$ in the layout of $C$ do
        if there is a parallel run b/w $A$ and $B$ then
            if there exist other signal(s) b/w $A$ and $B$ then
                if S2S distance $<$ parallel run then
                    Add a S2S 2MS defect to the list;
                end
            else if S2S distance $<$ 5DRC then
                // E.g., the dotted blue arrow in Fig. 4
                Add a S2S 2MS defect to the list;
            end
        end
        if the C2C criterion is met then
            // E.g., the case in Fig. 2
            Add a C2C 2MS defect to the list;
        end
        if the C2P criterion is met then
            // E.g., the case in Fig. 3
            Add a C2P 2MS defect to the list;
        end
    end
    // Top-off 3MS defects
    for each candidate triplet of rectangles in the layout of $C$ do
        if "every" pairwise S2S distance $<$ 5DRC then
            // E.g., the case in Fig. 5
            Add a 3MS defect to the list;
        end
    end
end
```

TABLE II: Runtime for fault extraction

Method	Runtime of each step				Total (excluding RC extraction)
	RC extraction	Defect generation	Analog simulation	Fault model gen. & UDFM gen.	
RCXT	17.3 hrs	2.0 mins	30.2 hrs	8.2 mins	30.4 hrs
2MS		3.6 mins	31.7 hrs	8.3 mins	31.9 hrs
3MS		5.7 mins	18.7 hrs	5.8 mins	18.9 hrs

TABLE III: Fault extraction of layout-based 2MS defects

	# faults	# EQ faults (percentage)	# DOM faults (percentage)
1tf-2MS	15,460	13,414 (89.0%)	14,667 (95.2%)
2tf-2MS	1,291	1,144 (87.4%)	1,175 (89.9%)

TABLE IV: Fault extraction of layout-based 3MS defects

	# faults	# EQ faults (percentage)	# non-EQ faults (percentage)	# DOM faults (percentage)	# non-DOM faults (percentage)
1tf-3MS	8,688	5,473 (73.2%)	3,215 (26.8%)	8,511 (98.5%)	177 (1.5%)
2tf-3MS	1,270	87 (6.9%)	1,183 (93.1%)	190 (15.0%)	1,080 (85.0%)

B. ATPG for RCXT-based and Layout-based 2MS Faults

In this subsection, we would like to compare the ATPG pattern sets resulting from our extracted layout-based 2MS faults, denoted as *layout-based 2MS patterns* and the RCXT-based 2MS faults (the default setting of CellModelGen [17]), denoted as *RCXT-based 2MS patterns*. The following experiments are conducted based on four industrial designs implemented with the same 28nm cell library shown in the previous subsection. The biggest design contains around 1.6M instances. Both fault models are represented in the format of UDFM and their patterns are generated by the same commercial ATPG, Tessent [18], with the same ATPG setting.

Table VI first lists the comparisons on their resulting 1tf 2MS patterns. As Table VI shows, the number of 1tf layout-based 2MS patterns is higher than that of 1tf RCXT-based patterns for each design since the number of extracted 1tf layout-based 2MS faults is also more than that of extracted 1tf RCXT-based 2MS faults. Among the four designs, the average number of 1tf layout-based 2MS patterns is 188 larger, which is 5.7% more compared to the average number of 1tf RCXT-based patterns. Then we further run the fault simulation on 1tf layout-based 2MS faults for both 1tf pattern sets, and the fault coverage resulting from the 1tf layout-based 2MS patterns is always higher than that resulting from the 1tf RCXT-based 2MS patterns for each design. The average coverage difference on 1tf layout-based 2MS faults is 0.36%. This result demonstrates that some realistic 1tf 2MS defects detected by the 1tf layout-based 2MS patterns cannot be successfully detected by the 1tf RCXT 2MS patterns.

Table VII then compares the 2tf 2MS pattern sets generated by the two fault models. As Table VII shows, the number of extracted 2tf layout-based 2MS faults is always smaller than that of extracted 2tf RCXT-based 2MS faults for each design, and hence the number of 2tf layout-based 2MS patterns is also always smaller. Among the four designs, the average number of 2tf layout-based 2MS patterns is 1003 smaller, which is 21.5% less compared to the average number of 2tf RCXT-based patterns. Note that the pattern-count reduction for 2tf layout-based 2MS patterns is much significant than the pattern-count increase for 1tf layout-based 2MS patterns as shown in Table VI. Then we further run the fault simulation on 2tf layout-based 2MS faults for both 2tf pattern sets, and the fault coverage resulting from the 2tf layout-based 2MS patterns is still always higher than that resulting from the 2tf RCXT-based 2MS patterns for each design despite the fact that the number of the 2tf layout-based 2MS patterns is much smaller. The average coverage difference on 2tf layout-based 2MS faults is 0.46%. This result demonstrates that the detection condition of a 2tf layout-based 2MS fault is more complicated than a 1tf one and in turn more difficult to be activated by random 2tf patterns or patterns targeting other 2tf 2MS faults. Plus, as shown in Section II-B, the layout-based 2MS faults can represent more realistic defects and avoid impractical ones than the RCXT-based 2MS faults. As a result, the 2tf layout-based 2MS patterns can outperform the 2tf RCXT-based ones on detecting real 2tf 2MS short defects with less number of patterns in use.

TABLE V: Cell type distribution of 3MS non-equivalent faults between 3MS fault and 2MS fault

Cell type	1tf 3MS faults			2tf 3MS faults		
	# total (A)	# non-EQ (B)	(B) / (A)	# total (C)	# non-EQ (D)	(D) / (C)
AN:OR:ND:NR	1,654	495	29.9%	312	271	86.9%
XNR:XOR	977	152	15.6%	207	207	100%
MUX	2,064	758	36.7%	191	166	86.9%
HA:FA	468	325	69.4%	12	12	100%
AO:OA	3,117	1,485	47.6%	548	527	96.2%
Total	8,688	3,215	37.0%	1,270	1,183	93.1%

C. Top-off ATPG for Layout-based 3MS Faults

In practice, 2MS defects occur more often than 3MS defects. As a result, people applying ATPG patterns for 3MS faults usually want to apply ATPG patterns for 2MS faults first. In this subsection, we attempt to validate the effectiveness of 3MS patterns in condition that 2MS patterns are already applied first. The 2MS patterns used in the following experiments are the layout-based 2MS patterns shown in the previous subsection. We also apply the same commercial ATPG with the same setting to generate the top-off 3MS patterns.

Table VIII shows the result of top-off 1tf 3MS patterns generated on top of the existing 1tf 2MS patterns. Table VIII also lists the coverage of 1tf 3MS faults resulting from the 1tf 2MS patterns and the top-off 1tf 3MS patterns. As Table VIII shows, the average number of the generated top-off 1tf 3MS patterns is 1.96% of that of the 2MS patterns among the four designs. However, the top-off 1tf 3MS patterns can only increase the coverage of 1tf 3MS faults by 0.02% in average compared to the 1tf 2MS patterns. This result demonstrates that majority of the ATPG-detectable 1tf 3MS faults are already detected by 1tf 2MS patterns.

Table IX shows the result of top-off 2tf 3MS patterns generated on top of the existing 2tf 2MS patterns. Among the four designs shown in Table IX, the top-off 2tf 3MS patterns can improve the coverage of 2tf 3MS faults by 9.33% compared to the 2tf 2MS patterns. Meanwhile, the average number of top-off 2tf 3MS patterns is 20.2% of that of 2tf 2MS patterns. This result demonstrates that a significant portion of the 2tf 3MS faults cannot be properly covered by the 2tf 2MS patterns because a large portion of 2tf 3MS faults are non-equivalent to or dominated by a 2tf 2MS faults as shown in Table IV. Therefore, the detection of those 2tf 3MS

TABLE VI: ATPG patterns generated for 1tf RCXT-based 2MS faults and 1tf layout-based 2MS faults.

Design	# cell inst.	# scan cells	# Faults		# Patterns			TC (ATPG)		layout-based FC (layout-based fault sim. w/ diff. pat.)		
			RCXT	Layout	RCXT (A)	Layout (B)	(B) - (A)	RCXT	Layout	Layout (C)	RCXT (D)	(D) - (C)
D1	28K	4K	178,143	202,657	421	480	59	95.82%	95.90%	91.41%	90.92%	-0.49%
D2	45K	6K	271,941	314,637	545	694	149	96.99%	97.00%	90.90%	90.28%	-0.62%
D3	1.6M	184K	8,892,935	10,028,241	8,325	8,437	112	93.83%	93.82%	89.98%	89.79%	-0.19%
D4	1.3M	98K	13,197,027	14,966,239	3,866	4,296	430	95.30%	95.60%	91.69%	91.57%	-0.12%
Avg.	743K	73K	5,635,012	6,377,944	3,289	3,477	188	95.49%	95.58%	91.00%	90.64%	-0.36%

TABLE VII: ATPG patterns generated for 2tf RCXT-based 2MS faults and 2tf layout-based 2MS faults.

Design	# cell inst.	# scan cells	# Faults		# Patterns			TC (ATPG)		layout-based FC (layout-based fault sim. w/ diff. pat.)		
			RCXT	Layout	RCXT (A)	Layout (B)	(B) - (A)	RCXT	Layout	Layout (C)	RCXT (D)	(D) - (C)
D1	28K	4K	7,076	3,929	657	464	-193	88.90%	92.75%	92.63%	92.20%	-0.43%
D2	45K	6K	15,774	8,142	981	690	-291	88.59%	90.24%	89.80%	89.40%	-0.40%
D3	1.6M	184K	433,336	270,614	16,057	12,743	-3,314	86.02%	87.13%	86.07%	85.56%	-0.51%
D4	1.3M	98K	8,757	5,172	990	775	-215	83.08%	89.12%	89.04%	88.52%	-0.52%
Avg.	743K	73K	116,236	71,964	4,671	3,668	-1,003	86.65%	89.81%	89.39%	88.92%	-0.47%

TABLE VIII: Top-off ATPG for 1tf 3MS faults.

Design	# 3MS Faults	# Patterns		3MS FC (fsim)		
		2MS	Top-off 3MS	2MS pat.	Top-off 3MS pat.	Diff.
D1	99,390	480	4	91.76%	91.78%	+0.02%
D2	150,862	694	3	91.77%	91.79%	+0.02%
D3	4,701,817	8,437	186	89.14%	89.15%	+0.01%
D4	8,438,931	4,296	79	93.35%	93.39%	+0.04%
Avg.	3,347,750	3,477	68	91.51%	91.53%	+0.02%

TABLE IX: Top-off ATPG for 2tf 3MS faults.

Design	# 3MS Faults	# Patterns		3MS FC (fsim)		
		2MS	Top-off 3MS	2MS pat.	Top-off 3MS pat.	Diff.
D1	3,771	464	156	77.37%	89.06%	+11.69%
D2	6,547	690	296	74.43%	86.41%	+11.98%
D3	98,626	12,743	2,433	83.20%	88.75%	+5.55%
D4	2,106	775	78	81.17%	89.27%	+8.10%
Avg.	27,763	3,668	741	79.04%	88.37%	+9.33%

faults needs to rely on their own dedicated patterns. Note that the extraction of 3MS faults are currently not supported by Synopsys' or Mentor Graphics' ATPG tool, meaning that those 2tf 3MS defects can be a leak of defect coverage for current commercial ATPG pattern sets. More importantly, this average 9.33% coverage loss on 2tf 3MS faults can be effectively made up by adding 20.2% more patterns on top of the original 2tf 2MS patterns.

IV. Conclusion

In this paper, we presented a layout-based CA test methodology for intra-cell short defects. Improved types of short defects such as side-to-side (S2S), corner-to-corner (C2C) and contact-to-poly (C2P) are incorporated in our framework for better fault coverage as compared to RCXT-based methodology. Furthermore, 3-metal-short (3MS) defects are modeled on top of 2-metal-short (2MS) defects for another 9.33% improvement in 2tf fault coverage. The efficacy and advantage of our proposed framework was verified against mainstream flow using commercial tools on a set of industrial designs, and can be seamlessly integrated into the mainstream flow.

References

[1] F. Hapke *et al.*, "Defect-oriented cell-aware ATPG and fault simulation for industrial cell libraries and designs," in *Proc. of ITC*, pp. 1–10, Nov. 2009.

[2] F. Hapke *et al.*, "Defect-oriented cell-internal testing," in *Proc. of ITC*, pp. 1–10, Nov. 2010.

[3] F. Hapke *et al.*, "Cell-aware analysis for small-delay effects and production test results from different fault models," in *Proc. of ITC*, pp. 1–8, Sept. 2011.

[4] F. Hapke *et al.*, "Cell-aware production test results from a 32-nm notebook processor," in *Proc. of ITC*, pp. 1–9, Nov. 2012.

[5] F. Hapke *et al.*, "Cell-aware experiences in a high-quality automotive test suite," in *Proc. of European Test Symp.* (ETS), pp. 1–6, May 2014.

[6] Y.-H. Huang *et al.*, "Methodology of generating dual-cell-aware tests," in *Proc. of VTS*, pp. 1–6, April 2017.

[7] A. Ainha, S. Pandey, A. Singhal, A. Sanyal, and A. Schmaltz, "DFM-aware fault model and ATPG for intra-cell and inter-cell defects," in *Proc. of ITC*, pp. 1–10, Oct. 2017.

[8] F. Hapke and P. Maxwell, "Total critical area based testing," in *Proc. of ITC*, pp. 1–10, Oct. 2018.

[9] W. Howell *et al.*, "DPPM reduction methods and new defect oriented test methods applied to advanced FinFET technologies," in *Proc. of ITC*, pp. 1–10, Oct. 2018.

[10] Y.-T. Nien *et al.*, "Methodology of generating timing-slack-based cell-aware tests," in *Proc. of ITC*, pp. 1–10, Nov. 2019.

[11] C. Acero *et al.*, "Embedded deterministic test points for compact cell-aware tests," in *Proc. of ITC*, pp. 1–8, Oct. 2015.

[12] H.-W. Liu, B.-Y. Lin, and C.-W. Wu, "Layout-oriented defect set reduction for fast circuit simulation in cell-aware test," in *Proc. of Asian Test Symp.* (ATS), pp. 156–160, Nov. 2016.

[13] P.-Y. Chuang, C.-W. Wu, and H. H. Chen, "Cell-aware test generation time reduction by using switch-level ATPG," in *Proc. of Int'l Test Conf. in Asia* (ITC-Asia), pp. 27–32, Sept. 2017.

[14] Z. Gao, S. Malagi, E. J. Marinissen, J. Swenton, J. Huisken, and K. Goossens, "Defect-location identification for cell-aware test," in *Proc. of Latin American Test Symp.* (LATS), pp. 1–6, March 2019.

[15] L.-Y. Ko, S.-Y. Huang, J.-L. Chiou, and H.-C. Cheng, "Modeling and testing of intra-cell bridging defects using butterfly structure," in *Proc. of Int'l Symp. on VLSI Design, Automation and Test* (VLSI-DAT), pp. 1–4, April 2006.

[16] A. Ladhar, M. Masmoudi, and L. Bouzaida, "Extraction and simulation of potential bridging faults and open defects affecting standard cell libraries," in *Proc. of Int'l Conf. on Signals, Circuits and Systems*, pp. 1–6, Nov. 2008.

[17] *Tessent® CellModelGen Tool Reference*, Version 2017.3, Mentor Graphics Corporation.

[18] *Tessent® Scan and ATPG User's Manual*, Version 2017.3, Mentor Graphics Corporation.

[19] *Calibre® xRC™ User's Manual*, Version 2012.2, Mentor Graphics Corporation.

[20] *HSPICE® User Guide: Basic Simulation and Analysis*, Version J-2014.09, Synopsys Inc.