Defect-Oriented Test: Effectiveness in High Volume Manufacturing

Friedrich Hapke, *Member, IEEE*, Will Howell, *Member, IEEE*, Peter Maxwell, *Life Fellow, IEEE*, Edward Brazil, *Senior Member, IEEE*, Srikanth Venkataraman, *Member, IEEE*, Rudrajit Dutta, *Member, IEEE*, Andreas Glowatz, *Senior Member, IEEE*, Anja Fast, *Member, IEEE*, and Janusz Rajski, *Life Fellow, IEEE*

***Abstract*—This article describes a defect-oriented test (DOT) approach, which enables a complete physical defect-based automatic test pattern generation (ATPG) for the digital logic area of CMOS-based designs. Total critical area (TCA)-based methods are presented for the generation of needed DOT views to enable the generation of complete DOT-based patterns for detecting all cell-internal and as well all cell-external physical defects. The major aim of these new methods and patterns is to further reduce the defect rate of manufactured ICs, in addition to what is already achieved with traditional and cell-aware test (CAT) fault models. We present test results, including achieved defect rate reduction in defective parts per million (DPPM), from a large 14-nm FinFET design, including a correlation to system-level-test (SLT) fails. For a second, mature 160-nm automotive mixed-signal sensor we present high-volume production test results, again measured in DPPM, and we provide test coverage figures moving away from counting detected faults to calculating detected TCA which is reported as the chip level TCA coverage.**

***Index Terms*—Automatic test pattern generation (ATPG), bridge defects, cell-aware test (CAT), defect oriented test (DOT), defect-based test, defective parts per million (DPPM), design for testability, failure analysis, FinFET test, logic testing, open defects, test data compression, total critical area, transistor defects, transistor-level test.**

I. Introduction

In the past, many papers have been published on stuck at (SA), transition delay faults (TDFs), gate-exhaustive (GE) and timing-aware (TA) fault models. A selection of those are [1]–[15]. In 1985, a first defect-oriented test (DOT) related paper on inductive fault analysis was published in [9]. More recently, cell-aware test (CAT) was introduced for detecting all cell-internal physical defects. The CAT method and its effectiveness in reducing test escapes measured in defective parts per million (DPPM) were published in [10]–[20]. CAT diagnosis results have been published in [18] and [35], demonstrating that an electrical diagnosis pinpoints exactly the indicated physical defects inside cells. Other DOT methods for detecting defects, external to standard cells like interconnect bridge defects have been published in [21], and very recently two DOT-based papers have been published in [36] for planar technologies and in [37] for FinFET technologies. When the probability of occurrence of the defects during the production process shall be taken into account, which is not considered by traditional fault models, the calculation of critical area as published in [24]–[33] comes into place. To achieve a high outgoing product quality in general and zero DPPM especially for automotive products, it is essential to target physical defects explicitly, and as such a DOT method is required.

This article gives a complete overview and detailed information about the DOT method, including details about physical defects, the generation of the needed test views, and the generation of DOT patterns. We also provide experimental and production test results for two different technologies, including a correlation to system-level test (SLT). We present fault coverage measurements based on detected total critical area (TCA), and we provide guidance for achieving the highest product quality with lowest test costs.

In Section II, we give an overview of physical defects and how the TCA is calculated for different physical defects. In Section III, we describe how the DOT views are generated. In Section IV, we present automatic test pattern generation (ATPG), production test, and SLT results from a 14-nm FinFET design, including high volume results achieved with timing-aware CAT (TA-CAT) patterns. ATPG and high volume production test results from a 160-nm automotive design are presented in Section V. In Section VI, we present how the highest product quality can be achieved with the lowest test costs. A conclusion is given in Section VII.

II. Physical Defects

For a DOT method, it is important to understand the type of physical defects that may occur during the production process. The most important defects to be identified are bridges,

Manuscript received September 25, 2019; revised February 23, 2020 and May 18, 2020; accepted May 28, 2020. Date of publication June 10, 2020; date of current version February 19, 2021. This article was recommended by Associate Editor A. E. Gattiker. *(Corresponding author: Friedrich Hapke.)*

Friedrich Hapke, Andreas Glowatz, Anja Fast, and Janusz Rajski are with the IC EDA/Tessent Department, Mentor, A Siemens Business, 21079 Hamburg, Germany, and also with the IC EDA/Tessent Department, Mentor, A Siemens Business, Wilsonville, OR 97070 USA (e-mail: friedrich.hapke@t-online.de; andreas_glowatz@mentor.com; anja_fast@mentor.com; janusz_rajski@mentor.com).

Will Howell, Edward Brazil, Srikanth Venkataraman, and Rudrajit Dutta are with the Manufacturing and Product Engineering Department, Intel Corporation, Sunnyvale, CA 95054 USA, also with the Logic Technology Development Department, Intel Corporation, Hillsboro, OR 97124 USA, and also with Intel Corporation, W23 CX68 Leixlip, Ireland (e-mail: will.howell@intel.com; edward.brazil@intel.com; srikanth.venkataraman@intel.com; rudrajit@gmail.com).

Peter Maxwell is with the Intelligent Sensing Group, ON Semiconductor, Santa Clara, CA 95054 USA (e-mail: Peterm6389@yahoo.com).

Digital Object Identifier 10.1109/TCAD.2020.3001259

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

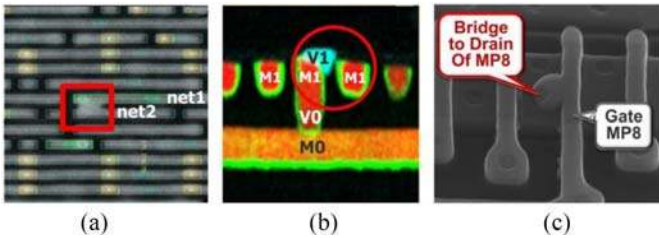

Fig. 1. Bridge defects outside and inside of standard cells.

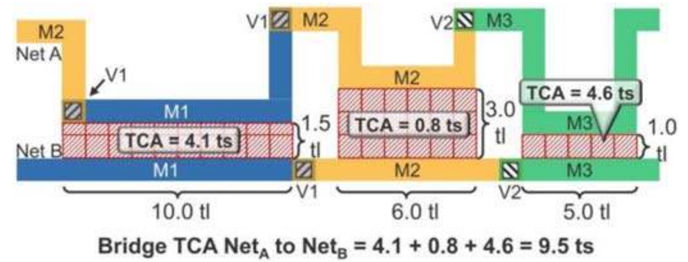

Fig. 2. TCA for a bridge from net A to net B.

opens, and transistor defects. We further distinguish between defects inside of standard cells, external to standard cells on the interconnect wiring, and cell-neighborhood defects. A further distinction can be made for bridges into hard bridges and resistive bridges, and for opens into full opens and resistive opens. For transistors, it is more difficult, but there are again hard defects which can result in hard shorts between drain and source, or between the gate and drain or source, or defects that have an effect on all terminals (including the bulk) of a transistor. But, also for transistors, there are often resistive types of defects which result in drive strength, leakage, and/or small delays introduced by these resistive transistor defects. In the following sections, we discuss in more detail the physical defects and their TCA calculations for bridges, opens, and transistor defects.

For the TCA calculations, dimensions are expressed in terms of a normalizing parameter named technology length (tl). The value of this parameter defaults to the width of metal1, which is extracted from the layout of the cell or the layout of the chip. Critical areas are expressed in units of technology squares (ts), where 1ts is the area of a square with a side length of 1tl, and for example the area of a rectangle with side lengths of 3tl and 2tl is 6ts.

A. Bridge Defects

Bridge defects are defined to be an unintended connection between two adjacent interconnect nets or two adjacent cell-internal physical objects of the same layer or of different layers. Examples for such bridge defects, proven by physical failure analysis (PFA), are shown in Fig. 1.

In Fig. 1(a), we show a cell-external interconnect bridge defect due to a side-to-side short on metal3 between net1 and net2, in Fig. 1(b), a cell-internal bridge due to a via1 (V1) bridge to two different metal1 (M1) objects, and in Fig. 1(c), a cell-internal bridge defect between the gate and drain of a planar transistor.

The probability of occurrence of bridge defects depends largely on three parameters: 1) the distance of the two adjacent nets/objects; 2) the length of the adjacency; and 3) the defect size distribution. The calculated TCA is a good measure for the probability of occurrence of the defect during the production process. Since the defect size distribution is technology dependent and difficult to obtain, we use a $1/s^3$ function, based on early work by Stapper [28]. The calculation is based on the approach in [29].

The formula for calculating the TCA for bridges is as follows:

$$\mathrm{TCA} = \int_{S_{\min}}^{S_{\max}} \frac{3 * \mathrm{dist}_{\min}^2 * (s + \mathrm{length}) * (s - \mathrm{dist})}{s^3} ds. \quad (1)$$

The definitions for the variables used in the formula shown in (1) are as follows:

1) s = spot size in [tl];
2) $s_{\min}$ = minimum spot size in [tl], that can create a defect;
3) $s_{\max}$ = maximum spot size in [tl] to be considered;
4) $\mathrm{dist}_{\min}$ = technology-dependent minimum spacing distance of nets in [tl];
5) length = length of the adjacent bridging area in [tl];
6) dist = distance of the two net segments in [tl].

More details can be found in [21]. In addition to calculating the TCA for two adjacent interconnect nets, the TCA of one interconnect net to power and to ground is calculated as presented in [36].

An example of a bridge TCA calculation is given in Fig. 2 for a cell-external bridge between net A and net B in the interconnect wiring. Both nets use wires in layer metal1 (M1), metal2 (M2), and metal3 (M3).

As can be seen in Fig. 2, there are three side2side bridge possibilities between net A and net B in layer M1, M2, and M3. The total bridge TCA for all three locations is the sum of the individual TCA values. In this example, it sums up to 9.5 ts. For calculating the three individual TCA values as shown in Fig. 2, formula (1) has been used. For the left bridge area for example with a spot size minimum of 1.5tl, and a spot size maximum of 3.5tl. The spot size minimum is always the distance, and the spot size maximum is always the distance plus 2.0tl.

For cell-internal bridges, it is important to know that there are many cell-internal interlayer bridge possibilities between different layers. Thus, it is important to know the cell-internal layer stack. A simplified layer stack for a FinFET technology is shown in Fig. 3.

The black arrows are indicating cell-internal interlayer bridge possibilities. The used layer abbreviations are as follows: DI = diffusion, PS = poly silicon, COP = contact to poly, COD = contact to diffusion, M0 = metal0, M1 = metal1, M2 = metal2, V0 = via from M1 to M0, and V1 = via from M2 to M1.

As an example, it can be seen in Fig. 3, that there is an interlayer bridge possibility from M0 to COD, and from M0 to PS. All these cell-internal interlayer bridges are considered

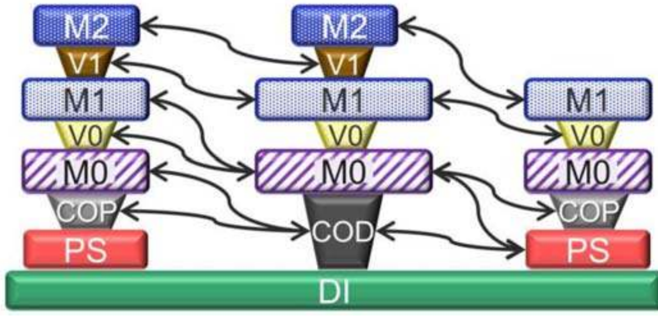

Fig. 3. Simplified FinFET layer-stack.

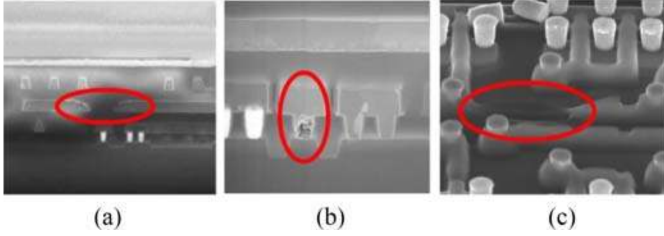

Fig. 4. Open defects outside and inside of standard cells.

for our DOT method. For each of those interlayer bridges, the TCA is calculated, and based on a predetermined TCA threshold, a decision is made if the bridge is to be considered or not.

B. Open Defects

Open defects are defined to be an unintentionally fully or partially disconnected interconnect net or via, or an unintentionally fully or partially disconnected cell-internal physical object of the same layer or a via between two different layers. Examples of open defects proven by PFA are shown in Fig. 4. In Fig. 4(a), we show a cell-external interconnect defect due to a metal2 open, in Fig. 4(b), a cell-external defect due to a missing via, and in Fig. 4(c), a cell-internal metal1 open.

The probability of occurrence of open defects depends largely on the following three parameters: 1) the width of the net segment; 2) the length of the net segment; and 3) the defect size distribution. The calculated TCA is a good measure for the probability of occurrence of defects during the production process. As explained for bridges already, since the defect size distribution is technology dependent and difficult to obtain, we use a $1/s^3$ function.

The formula for calculating the TCA for opens is as follows:

$$\mathrm{TCA} = \int_{S_{\min}}^{S_{\max}} \frac{3 * \mathrm{width}_{\min}^2 * (s + \mathrm{length}) * (s - \mathrm{width})}{s^3} ds. \tag{2}$$

The three variables used in the formula shown in (2) are as follows:

1) $\mathrm{width}_{\min}$ = technology-dependent minimum width of nets in [tl];
2) length = length of the net segment in [tl];
3) width = width of the net segment in [tl].

A layout example for the TCA calculation for open defects is given in Fig. 5.

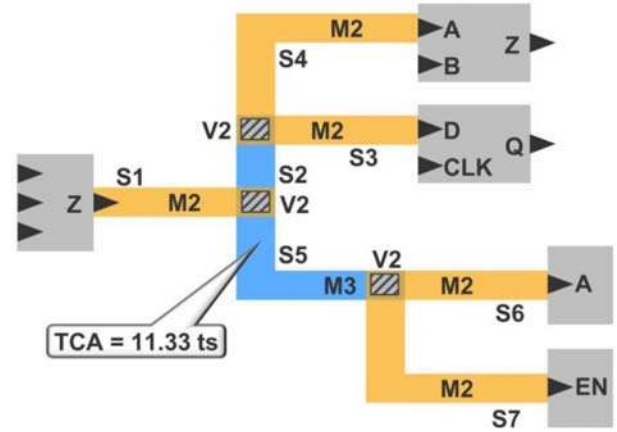

Fig. 5. TCA for each net segment.

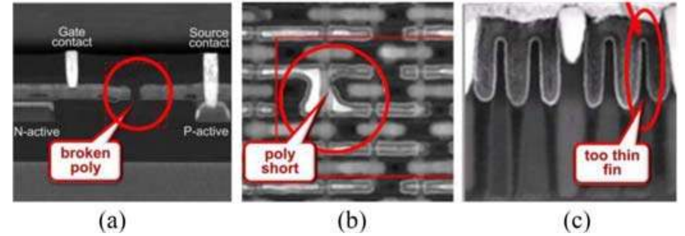

Fig. 6. PFA photographs of transistor defects.

For each net segment S1–S7 in Fig. 5, the TCA is calculated independently as described in (2). The TCA for net segment S5 is shown with a callout box in Fig. 5.

The example in Fig. 5 shows a cell-external net, connecting standard cells, but the same TCA calculation is done for cell-internal nets as well.

The TCA for vias (contacts from one layer to another layer) is calculated with the same formula as for net segments, with the addition that multiple/redundant vias on the same net segment will be considered. The TCA for multiple/redundant vias will be smaller than the TCA of a single via. The TCA of vias is allocated to the corresponding net segment, meaning that the TCA of each net segment is the sum of the net segment routing layer plus the TCA of the via(s).

C. Transistor Defects

Transistor defects are defined to be a full or partly non-functional transistor. These transistor defects result in either constantly or partly on or off defects, a drive strength defect, or a leakage defect. In case that the transistor is still switching, then the defect typically results in a small or large delay at the cell output introduced by the defective transistor.

Examples of transistor defects are shown in Fig. 6. Details on Fig. 6(a) have been published in [17]. In Fig. 6(b), we show poly patterning (short) defects, and in Fig. 6(c), a too thin fin defect.

The probability of occurrence of transistor defects depends largely on the channel length, the channel width, and the defect size distribution.

As explained for bridges and opens already, since the defect size distribution is technology dependent and difficult to obtain, we use a $1/s^3$ function.

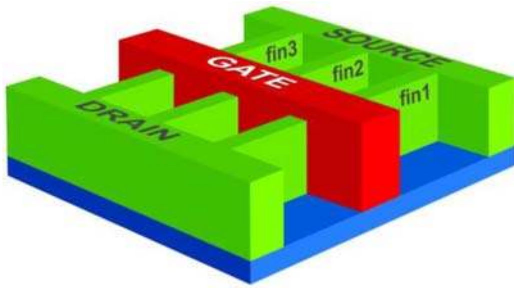

Fig. 7. FinFET transistor.

The TCA calculation formula for transistor defects is very similar to bridge and open defects and is as follows:

$$\text{TCA} = \int_{S_{\min}}^{S_{\max}} \frac{3 * \text{clength}_{\min}^2 * (s + \text{cwidth}) * (s - \text{clength})}{s^3} ds. \tag{3}$$

The three variables used in the formula shown in (3) are as follows:

1) $\text{clength}_{\min}$ = technology-dependent minimum channel length of transistors in [tl];
2) cwidth = channel width of the transistor in [tl];
3) clength = channel length of the transistor in [tl].

For FinFET transistors, the values for cwidth and clength are derived from the number of fins and from the fin dimension (fin width, fin height, and pitch).

There is a risk in FinFET technologies that small delays are introduced much more than it is the case in planar transistor technologies. This is because of the 3-D nature of a FinFET transistor as shown in Fig. 7, where each fin of the 3-D transistors can have defects on its own, which will result either in reduced drive strength, because one or more fins are not operating as they should, or in leakage current within one or more fins of the transistor.

When only one fin (or a small number of fins) produces an abnormally high leakage current, then the defect behavior at the cell output will be a small delay and the finally settled state will not reach the fault-free voltage.

When only one fin (or a small number of fins) produces an abnormally low drive strength, then the defect behavior at the cell output will only be a small delay, but the finally settled state reaches the fault-free voltage in a static test.

When all fins create a too high leakage or a too low drive strength, then the defect behavior will result in a gross delay.

In planar technologies, small delays can also be introduced for cells with a high drive strength, because the high drive strength is typically produced by multiple parallel fingers of planar transistors. But in FinFET technologies, even when a cell with the lowest drive strength is realized, each transistor will typically have multiple (parallel) fins.

D. Cell Neighborhood Defects

In order to target cell-neighborhood defects, it is necessary to perform an extraction process based on the IC layout. During this extraction process, a list of adjacent cell pairs is determined, covering the cell-neighborhood defects. A cell pair

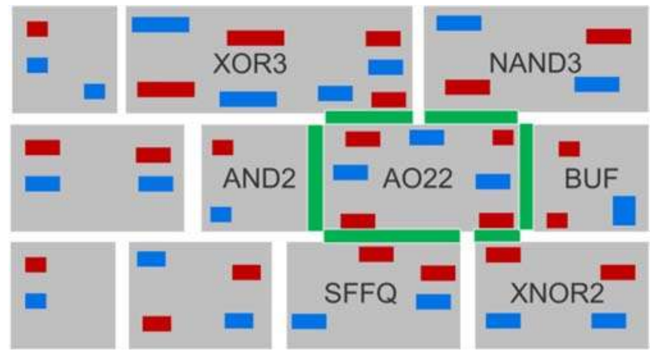

Fig. 8. Cell-neighborhood defect possibilities.

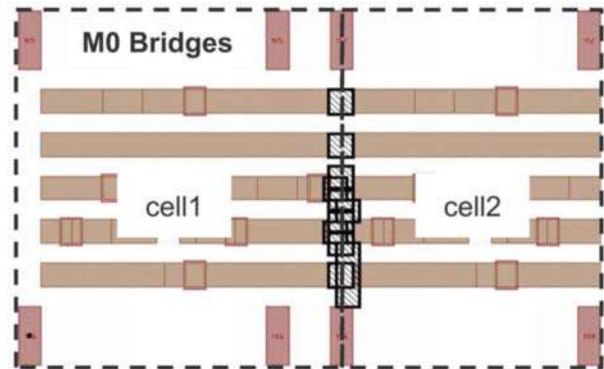

Fig. 9. Cell-neighborhood analysis—case 1.

is defined as a unique combination of two cells, considering the specific placement in the layout. This includes the distance between the cells, if a cell is flipped and/or rotated, and the position in *x*/*y* direction to each other.

A simplified IC layout displaying defect possibilities between neighboring cells is shown in Fig. 8. The possible bridging areas between the AO22 cell instance in Fig. 8 and its neighbor cells (in this example, six neighbor cells) are marked with *green rectangles*. The green marked areas need to be analyzed to identify potential bridges. In this example, six cell pair combinations need to be analyzed, considering only bridges between the neighboring cells, because open defects cannot occur in that area.

In Fig. 9, an example for cell-neighborhood defects is shown, where cell 2 is simply placed right of cell 1, i.e., cell 2 has just an offset in the *x*-direction, without an offset in *y*-direction, and it is not rotated nor flipped.

As can be seen in Fig. 9, there are 13 metal0 bridge possibilities (shown as *black rectangles*) between cell 1 and cell 2 that need to be targeted.

Another cell-neighborhood case is shown in Fig. 10. In this example, the layout tool first rotated cell 2 to direction south, and then flipped it around the vertical axis (orientation FS) such that the VDD power line, which is in metal2 and shown in the in Fig. 10, can be shared between cell 1 and cell 2. In addition, cell 2 has an offset of 108 nm in the *x*-direction, depicting a completely different situation as for case 1.

As a result of this special but very typical placement, shown in Fig. 10 there are two possible metal1 bridges (*blue shaded*

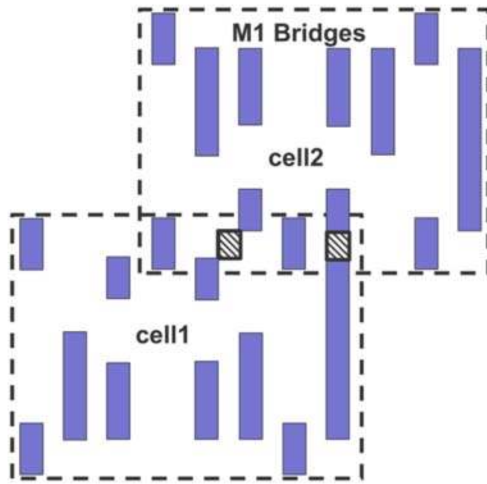

Fig. 10. Cell-neighborhood analysis—case 2.

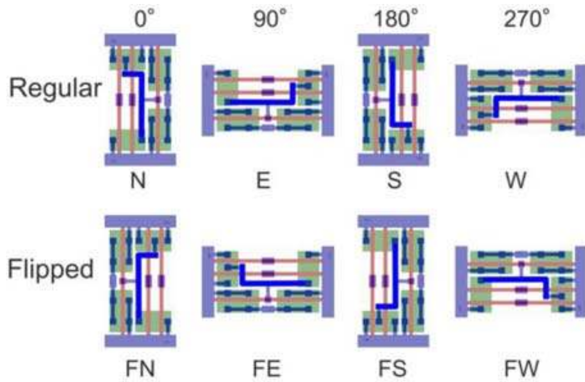

Fig. 11. Orientation variants.

rectangles) between cell 1 and cell 2 to be targeted. This example also shows how complex it is to identify potential cell-neighborhood defects as it is a clear chip layout dependent situation, which means layout tools have the freedom to place a cell in eight different orientation variants as shown in Fig. 11.

The orientation variant north “N” is the original layout of the standard cell. The flipped variants are first rotated and then flipped around the vertical axis.

For the two designs discussed in this article, cell neighbor extraction resulted in 123 000 cell pair variants for the ON design and 330 000 cell pair variants for the Intel design (for details see [36] and [37]).

The formula for calculating the TCA for cell-neighborhood defects is the same as for bridges. For details see formula (1).

III. Defect-Oriented Test View Generations

To be able to target all physical defects explicitly and accurately, dedicated test views need to be generated from the standard cell layout and from the actual layout of the chip.

The whole DOT-based defect view generation and ATPG flow are illustrated in Fig. 12.

The format that we use in the generated DOT view files is the user-defined fault model (UDFM), a format that defines fault models as needed, to specify test cubes which force an ATPG to target the specified faults explicitly.

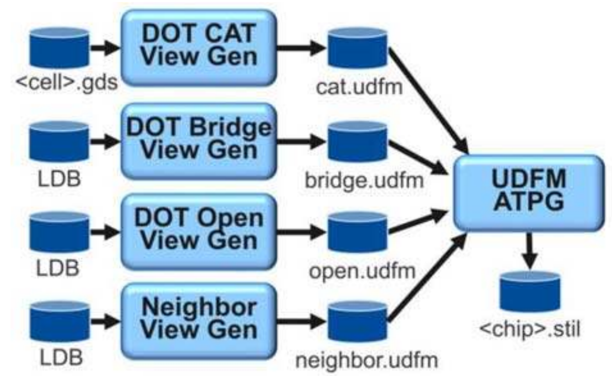

Fig. 12. DOT view generations.

The format for the layout of standard cells is the well-known GDSII format, and for the chip layout we use a layout data base (LDB) format. Using these two formats has a big advantage that no new file format is required to generate the needed DOT views.

In the following sections, we provide details for the creation of those DOT view files.

A. Cell-Internal DOT Views

The view generation for all cell-internal defects (bridges, opens, and transistor defects) has been described already in detail in various publications; (see [16], [17], [19], and [20]). This view file is well known as technology-dependent CAT view file, but we want to point out here, that for each defect a TCA is now calculated as well [as described in Section II, formula (1)–(3)] and stored in the CAT view file.

In addition to the calculated TCA for each defect, the defect delay behavior at the cell output is stored. This is important in order to target small delay defects explicitly. For this, a cell must first be analyzed for each cell-internal defect, to determine if it creates a small delay or a gross delay at the output. This analysis is carried out when creating the CAT view for each standard cell in a certain technology. In Fig. 13 we show a few defect behaviors of cell-internal defects resulting in no detection, in a small delay detection, and as well in a gross delay detection.

1) The *black* waveforms, also marked with *black dots* at the strobe time, are the fault-free waveforms. Depending on the stimuli applied to the cell inputs there are fault-free best case and worst case waveforms.
2) The *blue* rising waveforms, also marked with a *blue dot* at the strobe time, are from defects creating a small delay at the cell output.
3) The *green* waveforms, also marked with a *green dot* at the strobe time, are from defects creating a gross delay at the cell output.
4) Undetected defects will all result in a defect behavior waveform that is within the black shaded area of the fault-free waveforms.

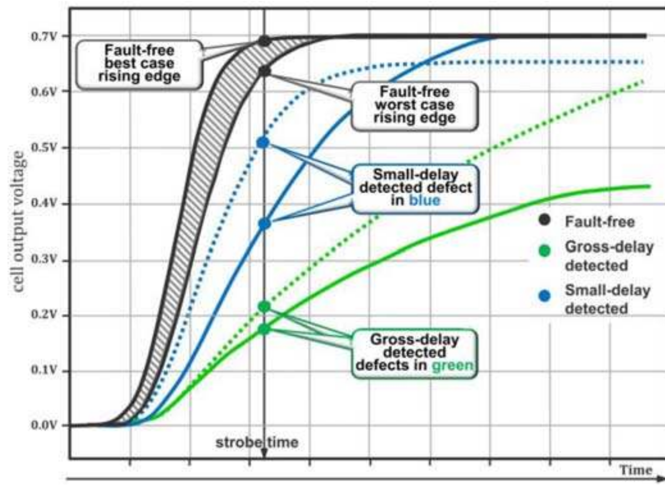

Fig. 13. Defect behavior—small and gross delays.

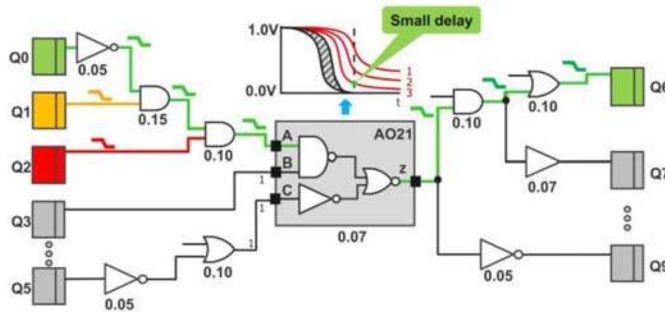

Fig. 14. TA-CAT ATPG example.

When generating the CAT view for an entire standard cell library, a view generation tool will store information for each defect about the size of the delay that is created at the cell output. A user definable threshold decides between small and gross delays. By default, a small delay is present when the defective cell output voltage changes to less than 50% at the strobe time. To be able to detect defects, which just result in small delays at the cell output, it is important that the output edge is propagated via long paths to an observation point, i.e., to a scan-flip-flop (SFF). It is also important that the edge at the cell input is created via a long path. These requirements are taken into consideration in TA-CAT.

Fig. 14 shows an example case of detecting a small delay defect being present in cell AO21. The number below each gate/cell indicates the cell delay in nanosecond. For simplicity the net delays as read from the standard delay format (SDF) file are not shown in Fig. 14, but they are clearly considered by TA ATPG. For this example, let us assume the selected defect requires as test condition a falling edge at the A input of the AO21 cell and a constant “1” state at the B and C input. Let us further assume the strobe time is 0.52 ns after the launch.

A safe detection will be reached when the input edge at the A input of cell AO21 is created at Q0 (*green* SFF on the left side of the figure) and the output edge of the AO21 cell is observed in Q6 (*green* SFF on the right side of the figure). This is the longest possible edge creation and defect observation path with a total cell delay of 0.57 ns (ignoring the net delays).

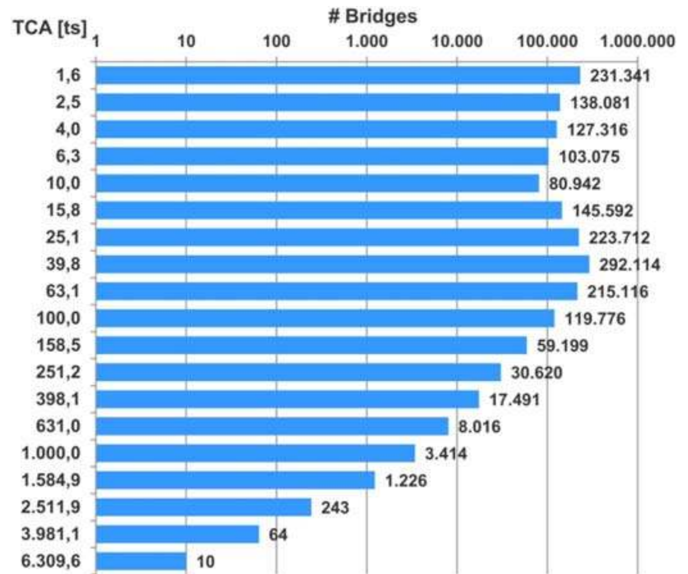

Fig. 15. Interconnect bridge distribution graph.

Creating the edge at the *orange* Q1, which would be possible as well, and observing still in Q6 will result in a total cell delay of 0.52 ns which may possibly result in a detection.

But creating the edge at the *red* Q2 and observing still in Q6 will result in a total cell delay of 0.37 ns. When assuming a small delay of 50 ps for example, an edge creation at the *red* Q2, will clearly result in NO detection of the small delay defect of 50 ps.

B. Interconnect Bridge DOT Views

For creating the bridge DOT views, the layout of the chip in LDB format is input to the extraction tool and the tool outputs the bridge DOT view as UDFM file, which can be passed on to the UDFM ATPG for generating the interconnect bridge patterns. The DOT view generation for bridges has already been published in detail in [21]. In addition to calculating the TCA for two adjacent interconnect nets, the TCA calculation of one interconnect net to power and to ground was described in [36].

As an example, Fig. 15 shows results of the bridge extraction for a chip, which has in total 9.8M extracted bridges.

Both axes in Fig. 15 have a logarithmic scale. There are ten bridges with a very large TCA of 6309 ts and many bridges with medium and small TCA. The TCA of all 9.8M interconnect bridge defects is 87.2M ts.

The bridge fault model used for ATPG is the 4-way model, which is used both for static and delay tests. The latter are to accommodate resistive bridges which may manifest themselves as delay faults. Fig. 16 illustrates the 4-way bridge model to generate delay test patterns.

For delay test patterns, the ATPG is forced to generate, propagate, and observe an edge on the victim net, while the aggressor net is forced to have a static zero and static one state.

For static test patterns, the ATPG is forced to generate a static zero state on the victim net instead of a falling edge, and a static one state on the victim net instead of a rising edge.

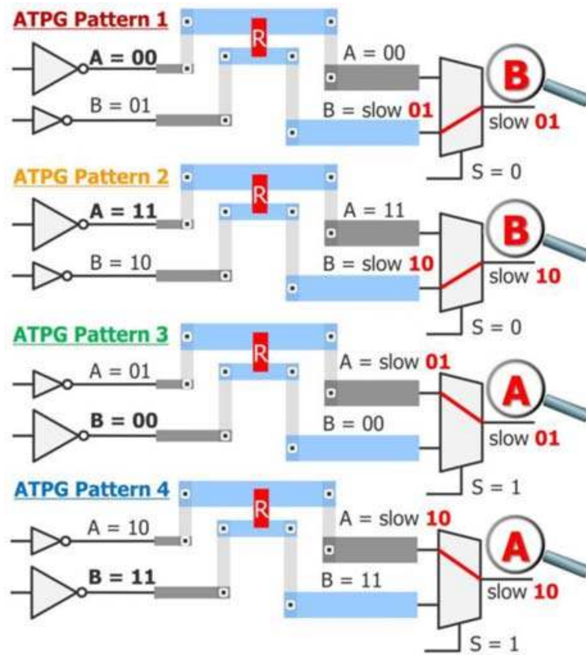

Fig. 16. Four-way bridge fault model for detecting delay defects.

The corresponding TCA value of a bridge defect is used for calculating and reporting the TCA coverage. A bridge defect reaches a 100% TCA coverage when all eight (four static and four delay) patterns are generated. Each of the four static patterns will increase the TCA coverage by 16.66%, and each of the four delay patterns will increase the TCA coverage by 8.33%. This weighting is based on results obtained in [21], which show that the vast majority of bridge defects are detected by static tests. As a result of that, we allocated 2/3 of the bridge TCA to static patterns and 1/3 of the TCA to delay patterns.

C. Interconnect Open DOT Views

The creation of the open DOT views is very similar to the creation of the interconnect bridge view, i.e., the LDB of the chip is again input to the extraction tool that outputs the open DOT view as a UDFM file. This UDFM file can be passed on again to the UDFM ATPG for generating the test patterns to detect interconnect open defects. The DOT view generation for opens has already been published in detail in [36].

As an example, Fig. 17 shows the results of the open extraction for the same chip as used for Fig. 15, which has in total 208 356 interconnect nets with 750 060 open segments.

The y-axis in Fig. 17 has a logarithmic scale, and the x-axis has a linear scale. In the example, there is a small number of open segments with a TCA of 1–10 ts. The majority of all open segments have a TCA in the range of 10–500 ts. A small number of open segments has a very large TCA in the range of 500–15 000 ts. In this example, the TCA of all 0.75M interconnect open defects is 67.7M ts.

The fault model used by the UDFM ATPG for targeting the interconnect open defects is shown in Fig. 18, which illustrates that for each interconnect net segment, the ATPG is forced to generate a zero and a one state for static patterns, and a rising and a falling edge for delay patterns.

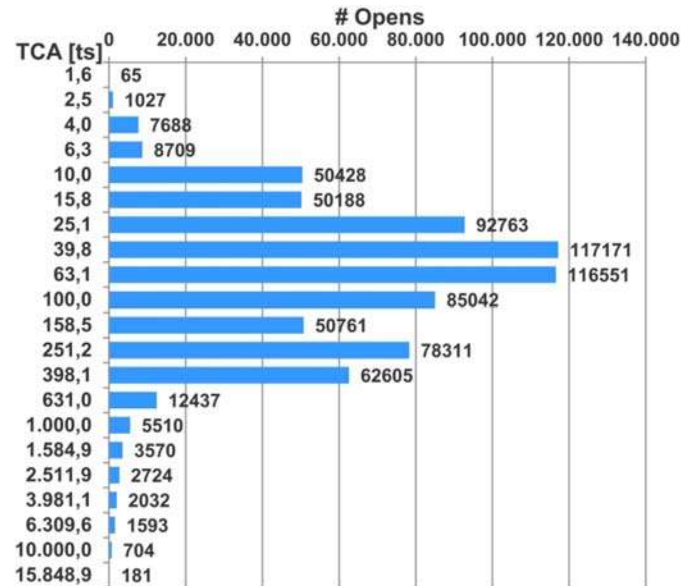

Fig. 17. Interconnect opens TCA distribution graph.

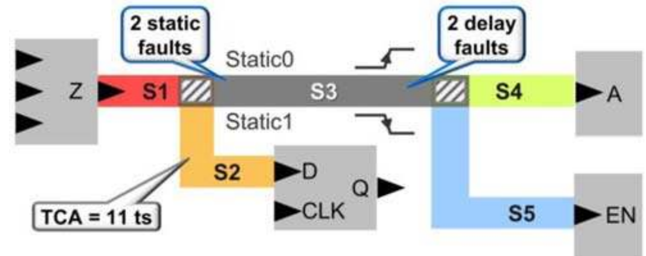

Fig. 18. Fault model for interconnect open defects.

The corresponding TCA value of each net segment is used for reporting the TCA coverage. An open segment reaches 100% TCA coverage when all four patterns are generated. Each of the two static patterns will increase the TCA coverage by 16.66%, and each of the two delay patterns will increase the TCA coverage by 33.33%. This weighting is based on well-known results obtained over decades which show, that the vast majority of open defects can only be detected with two cycle delay tests. Thus, we allocated 1/3 of the open TCA to static patterns and 2/3 of the TCA to delay patterns.

D. Cell-Neighborhood DOT Views

The first step for generating the DOT view for cell-neighborhood defects is the extraction of cell pairs that are adjacent to each other. For this, the LDB of the chip is input to the extraction tool that outputs an interface file in UDFM format, containing ranked cell pair information (from most important to least important). This is used as input to the second step, which is to merge the calculated cells pairs for creating the actual DOT view for each cell pair instance.

For the DOT view generation, two neighboring cells that have previously been extracted as a cell pair, are merged into a virtual merged cell by taking the offset in x and y direction

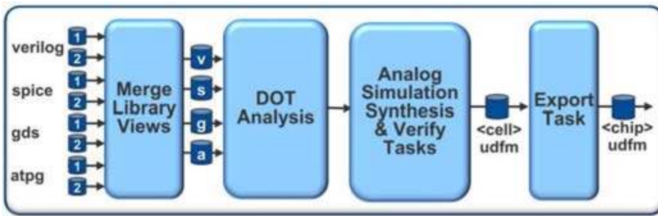

Fig. 19. Cell-neighborhood DOT view generation.

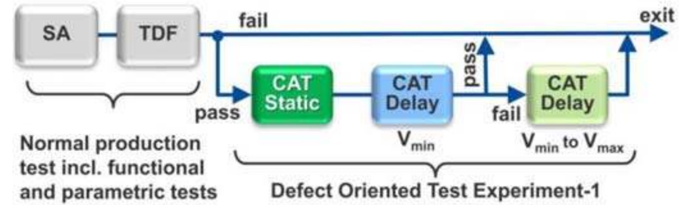

Fig. 20. Test program flow 14-nm Experiment-1.

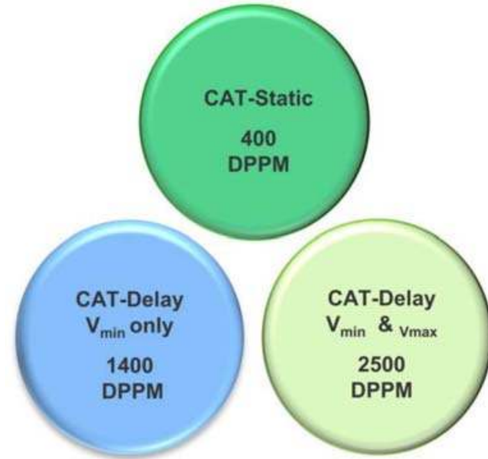

Fig. 21. Test escape rate reductions in DPPM from Experiment-1.

and also the flip and rotation information into account. The DOT view generation is illustrated in detail in Fig. 19.

After merging the cell pair into one virtual merged cell, and creating the combined Verilog, SPICE, GDS, and ATPG views, a normal DOT analysis is done similar to cell-internal defects, with the difference being that only bridges are analyzed and the area of interest is not the complete area of the two merged cells, but just a small area where the two cells are adjacent to each other. This DOT cell-neighborhood analysis is followed by analog simulations for each identified defect and by the cell-aware synthesis and verification task as already published in [35].

Each cell pair related test view is finally exported into a UDFM file by also taking the individual cell pair locations (hierarchical net names connected to the ports of each cell pair) into account. This UDFM file is input to the ATPG run as shown in Fig. 12 to generate the needed test patterns for detecting all cell-neighborhood defects in a given chip layout.

As an example, the cell-neighborhood extraction for the same chip as used for Fig. 15, and Fig. 17, which has in total 123K cell pairs to be analyzed, resulted in 0.3M detectable cell-neighborhood bridges with a TCA of 0.6M ts. Further details have been published in [36].

IV. ATPG AND TEST RESULTS 14-NM FINFET DESIGN

To judge the effectiveness of the defect oriented test method, we selected as first vehicle an Intel IP with a large area implemented in a 14-nm FinFET technology and executed various experiments as described in Sections IV-A–IV-D.

A. ATPG and Test Results—Experiment-1

In the first experiment, we still compared CAT-Static and CAT-Delay patterns with traditional SA and TDF patterns. Details from this first experiment have been published in [37], and as such the ATPG runs and ATPG results are not shown here again. But the test flow and executed tests are important to understand the complete effectiveness of all DOT methods and patterns. Fig. 20 shows the test program flow of our first DOT experiment.

As shown in Fig. 20, the three DOTs are done for all units that passed the entire normal production tests, including all functional and parametric tests. The tests are done with the same VDD voltages as used for all functional, SA, and TDF tests. Regardless of the CAT-Static test result (whether it fails or not), the CAT-Delay patterns are executed with minimum VDD ($V_{\min}$) and in case of a fail, the same part is tested again with multiple VDD from $V_{\min}$ to maximum VDD ($V_{\max}$).

The execution of the *CAT-Static* patterns (green box in. Fig. 20) on their own achieved a reduction of *400 DPPM*. The execution of the CAT-Delay patterns at $V_{\min}$ (blue box in Fig. 20) detected *3900 DPPM*. During the following $V_{\min}$ elevation recovery flow, (light green box in. Fig. 20), we retested the failing parts (3900 DPPM) again with the same CAT-Delay patterns, but in this test not just with $V_{\min}$ but also with larger VDD voltages up to $V_{\max}$. During this test, *1400 DPPM* passed with a higher VDD than $V_{\min}$, and as such these are the $V_{\min}$ *only* failing parts, and the remaining parts that did not pass with increased VDD are *2500 DPPM* that fail at both $V_{\min}$ and V_{max}.

These test results from this first experiment are shown in Fig. 21. It can be seen that the *CAT-Static* patterns on their own achieve a reduction of *400 DPPM*, compared to SA and TDF patterns and all other before executed tests, including functional tests.

But the largest reduction of *2500 DPPM* is from parts uniquely failing the *CAT-Delay* patterns at $V_{\min}$ and V_{max}.

In addition, there is this unique detection of *1400 DPPM* failing at $V_{\min}$ *only*, i.e., from parts that do not fail anymore with an elevated VDD greater than $V_{\min}$, indicating CAT-Delay patterns are more accurately assessing silicon speed distribution.

B. System-Level Test Results—Experiment-2

For this experiment, we selected 156 units from the same 14-nm FinFET IC that passed the entire traditional production test suite (which executed SA, TDF, and all functional test patterns), and only failed in SLT. Moreover, the units were

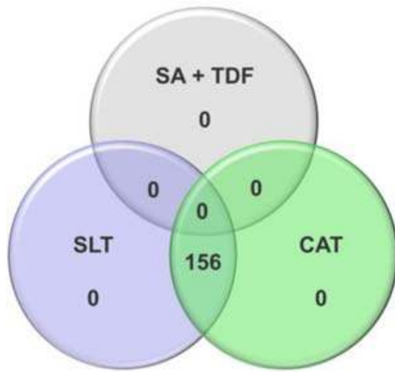

Fig. 22. System-level test results from Experiment-2.

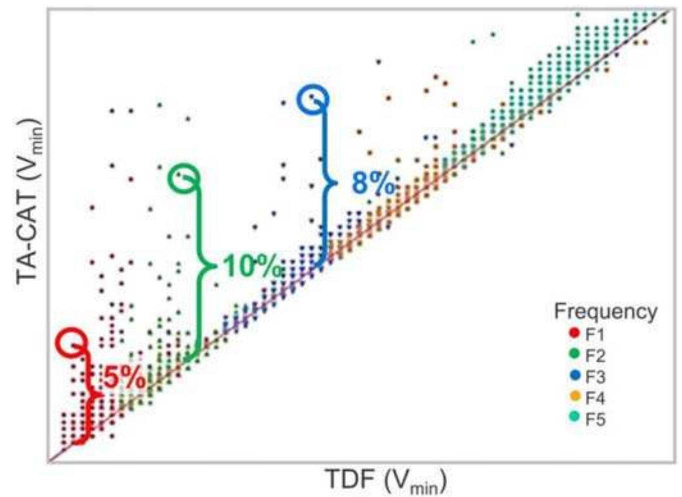

Fig. 23. $V_{\min}$ correlation TDF versus TA-CAT.

selected based on SLT failure syndrome such that they were each almost certain to be a result of a random defect in the tested part.

The test program flow to test the selected 156 SLT failures was the same as for the first experiment.

Fig. 22 shows the perfect correlation between SLT fails and CAT fails, observed for Intel's 14-nm FinFET IC.

It was expected that the SA and TDF patterns would not fail because the units had been tested with SA and TDF patterns during production testing. It was also expected that at least some units would fail CAT patterns as these units were essentially "known" SLT failures.

What was not expected was, that *all* units failed with CAT patterns. In more detail, the result was that zero units failed the CAT-Static patterns, but all 156 units failed the CAT-Delay patterns at $V_{\min}$. At nominal VDD just ten units failed with CAT-Delay patterns. The remaining 146 units were all identified to need a significant V_{min} shift to pass; an average shift of 55 mV above the specified $V_{\min}$ was observed.

C. TA-CAT Results—Experiment-3

Further analysis of selected failing 14-nm parts was done by executing three delay pattern files (the TDF, CAT-Delay, and TA-CAT) multiple times using different VDD voltages and different test frequencies. Details about the pattern generation for these three delay pattern files have been published in [37], but the results are displayed again in Fig. 23, to show the correlation between TDF and TA-CAT.

As can be seen in Fig. 23, there is a clear $V_{\min}$ test strength improvement from TA-CAT patterns versus TDF patterns. There is a bulk distribution shift in addition to improved outlier detection.

As an example, see the *red* circled case in Fig. 23, which is a part tested with frequency F1, starting to pass the TA-CAT tests with a $V_{\min}$ shift of 5% higher than required for the TDF tests.

The importance of these results is that the observed $V_{\min}$ shift is fully in line with SLT results, (see Section IV-B for details). Furthermore, this experiment has shown that TA-CAT patterns add a significant number of unique detections to the already large number of unique CAT-Delay detections in relation to traditional TDF patterns, details in [37].

The high quality improvement of 4300 DPPM during wafer test, the total match of DOT fails with SLT fails for all 156 SLT rejects, and the TA-CAT results, have convinced Intel to focus on delivering DOT patterns to upcoming products such that traditional SA/TDF patterns are no longer utilized.

D. ATPG and Test Results—Experiment-4

The results achieved with the 14-nm experiments as described in Sections IV-A–IV-C, led to the decision at Intel to no longer utilize traditional SA and TDF patterns in high volume manufacturing (HVM) tests and to base the structural HVM tests fully on CAT-Static, CAT-Delay, and TA-CAT patterns without any execution of SA and TDF patterns. A second major change is in obtaining the base $V_{\min}$ evaluation of the product no longer on TDF patterns, but on CAT-Delay patterns with the target to get a much better $V_{\min}$ correlation to the actual $V_{\min}$ of the product as achieved with SLT.

For this HVM experiment, we have chosen to do all ATPG runs from scratch, as detailed below.

The *CAT-Static* ATPG run targets all cell-internal static detectable defects. For this ATPG run, the cat.udfm file is read. The DOT view generation for this UDFM file is described in Section III-A.

The *CAT-Delay* ATPG run targets all cell-internal delay detectable defects. For this ATPG run, the cat.udfm file is read. The DOT view generation for this UDFM file is described in Section III-A.

The *TA-CAT-Delay* ATPG run targets all cell-internal small-delay detectable defects. For this ATPG run, the cat.udfm file and in addition the SDF file are read, to enable both the TA small-delay propagation via long paths, and the generation of the needed edges at the cell inputs via long paths. The DOT view generation for this UDFM file is described in Section III-A. For this TA-CAT ATPG run, we set the small-delay limit so that all defects that produce a defect impact greater than 50% are filtered out. Setting the defect impact threshold to 50%, means that in case of a defect, the cell output voltage changes to more than 50% of the supply voltage. For this design, it

TABLE I
ATPG RESULTS OF 14NM EXPERIMENT-4

Fault Model	Number Faults	Number Patterns
CAT-Static	55.1M	7,424
CAT-Delay	45.0M	37,000
TA-CAT	11.0M	46,000

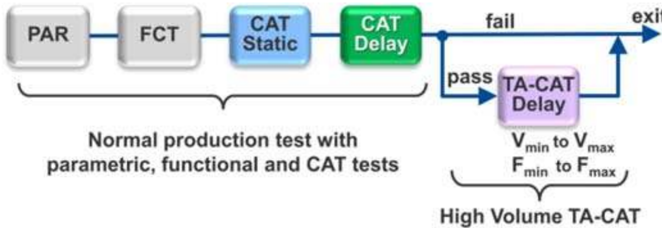

Fig. 24. Test program flow 14-nm Experiment-4.

resulted in only about 25% of all CAT-Delay defects being targeted by the TA-CAT ATPG.

As described before, traditional SA/TDF patterns are no longer utilized and test results are shown for what is achieved in addition to the achievements of CAT-Static and CAT-Delay patterns.

The results from these three pattern generations are shown in Table I. All ATPG runs have been done from scratch; no top-off run was done. This enabled us to compare the effectiveness of the different patterns and to get unique detection information from the test system.

As can be seen from Table I, the number of TA-CAT faults are just about 25% of the CAT-Delay faults.

A high test coverage (TC) of >98% was achieved with CAT-Static patterns and >85% with CAT-Delay patterns.

The test program flow of the fourth experiment as shown in Fig. 24 was applied to millions of units, produced in the 14-nm FinFET technology. It is important to note that all units tested with TA-CAT patterns had already passed all parametric (PAR), all functional (FCT), and as well the structural CAT-Static and CAT-Delay tests; any new failures are thus uniquely from TA-CAT tests.

As shown in Fig. 24, the TA-CAT Delay patterns are executed with multiple frequencies and multiple VDD voltages to calculate the $V_{\min}$ shift in relation to traditional CAT and FCT patterns.

Based upon the strength of results in Experiments 1–3, TA-CAT patterns were added directly to the HVM test program without any engineering flow testing for DPPM and $V_{\min}$. As such, TA-CAT patterns were run on well over one million units across multiple products. In addition, the TA-CAT patterns have been added to both wafer sort and final package test with unique benefits in both sockets.

As this product/process is mature, there is not a great deal of unique DPPM left to be detected. Despite this, TA-CAT patterns at wafer sort delivered a unique reduction of ~300 *DPPM*, a good portion of which was detected before only in a cold package test. This was an exciting result as it demonstrates that TA-CAT patterns can be used for advanced cold package test reduction. The result was not expected due to TA-CAT $V_{\min}$ focus, but it can be rationalized; TA-CAT stresses the tightest margin paths with small-delay cell defects and so even reverse temperature correlation for speed cannot help these parts pass as the patterns are so timing robust.

In the final package test, unique failures were below 50 DPPM. The $V_{\min}$ performance, however, was similar to results in Experiment 3 with an intrinsic ~20-mV shift in $V_{\min}$ above CAT-Delay performance and unique outliers as high as 360 mV were observed. As the population tested is in HVM, these units were sold directly, and no SLT correlation was done, which demonstrates the confidence placed in TA-CAT quality. The content has also been used to optimize our test flows by reordering what is run first and enhancing our $V_{\min}$ search sweeps for test time reductions. The HVM results of TA-CAT on these 14-nm IP have inspired TA-CAT ATPG for similar 10-nm IP.

V. ATPG AND TEST RESULTS 160-NM AUTOMOTIVE DESIGN

To evaluate the effectiveness of the complete DOT method and the feasibility of calculating the TCA for all physical defects, we applied the method to the logic area of a 160-nm automotive mixed-signal sensor.

For this design, we applied a TCA calculation for all bridges and opens on the interconnect, for all defects inside of standard cells and for cell-neighborhood defects between adjacent standard cells.

The chip layout is shown in Fig. 25. This design has ~400K digital gates and 1.4M SA faults.

A. ATPG Runs and ATPG Results

For this production test experiment, a different set of patterns has been generated than what was shown in Section IV. The focus here was to have a test pattern set that is acceptable from a test time point of view in production, but still provides an insight into aspects that were not given before. Previously, quality improvements from CAT versus SA and TDF patterns were evaluated [16]–[20]. We also already evaluated the quality improvements from interconnect bridge patterns versus CAT-Static and CAT-Delay patterns for this design in [21]; and in [35], we have shown that interconnect open defects are well covered with CAT-Delay patterns. Hence, we now did a combined CAT+Bridge+Open delay pattern generation named DOT-Delay, and as well a combined CAT+Bridge+Open static pattern generation, but we split the static pattern generation into DOT-Static1 and DOT-Static2 (see below). In addition, we separated the cell-neighborhood patterns and generated dedicated TA-CAT pattern to target all cell-internal small delays explicitly. The performed ATPG runs to generate the desired production test patterns are shown in Fig. 26.

DOT-Delay: The first ATPG run generating pattern reference *PR1*, targets CAT-Delay, interconnect bridge delay, and interconnect open delay defects. For this ATPG run, three DOT UDFM files are read: the cat.udfm, the bridge.udfm, and the open.udfm file.

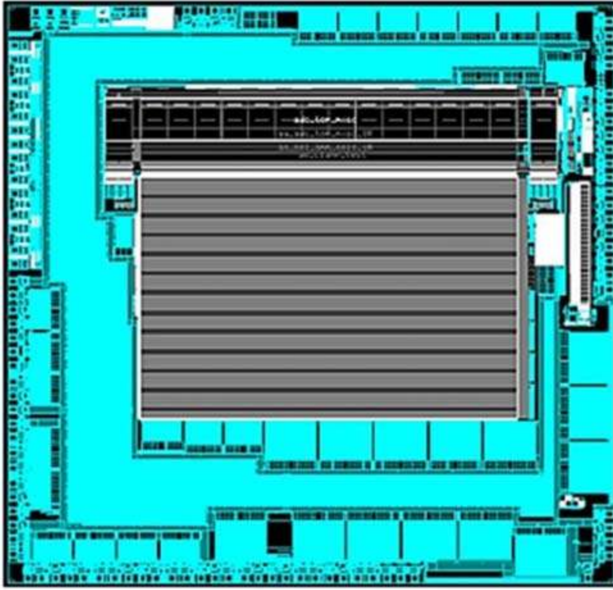

Fig. 25. 160-nm automotive design.

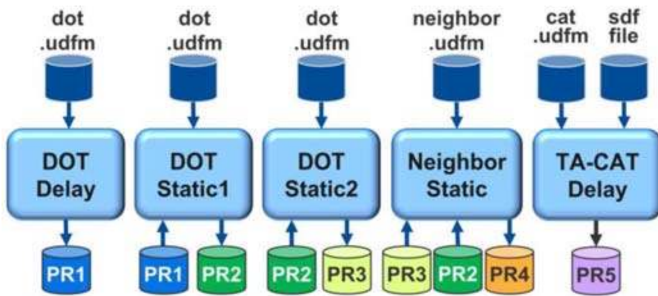

Fig. 26. Production test pattern generation runs.

DOT-Static1: The second ATPG run, generating *PR2*, is a static top-off run on PR1 patterns, targeting all CAT-Static, interconnect bridge static, and interconnect open defects that have not been marked as detected with a static fault simulation (FSIM) of PR1 patterns. For this FSIM and ATPG run, again three DOT UDFM files are read, these are the cat.udfm, the bridge.udfm, and the open.udfm file.

DOT-Static2: The third run is to challenge the common practice in industry to generate top-off static patterns after fault simulating the delay patterns using a static fault model. *PR3* is created by statically fault simulating the PR2 top-off patterns and generating patterns for the remaining undetected faults. According to common industry practice, PR3 should not be necessary because the targeted faults have already been detected by delay patterns, but we were seeking silicon proof. For this FSIM and ATPG run, the same three DOT UDFM files are read.

Neighbor Static: The fourth run generates *PR4*, targeting all cell-neighborhood defects, to evaluate their effectiveness explicitly. For this ATPG run, only the cell-neighbor UDFM file is read, and PR2 plus PR3 static patterns are fault simulated first. Neighbor-delay patterns were not generated because the results in [21] show that the vast majority of bridge defects are detected by static tests.

TABLE II
PRODUCTION PATTERN AND COVERAGE RESULTS

	#pat	#faults	TCA [ts]	FC [%]	RE [%]	TC [%]	TCA [%]	UC+AU [%]
PR5	5617	4.5M	30M	80.61	1.09	81.94	80.86	19.14
PR1	5572	25.9M	291M	81.28	2.45	83.51	82.89	17.11
PR2	2345	22.9M	214M	93.64	2.76	96.49	96.66	3.34
PR3	2377	22.9M	214M	96.20	2.76	99.13	98.61	1.39
PR4	189	22.9M	214M	96.21	2.76	99.15	98.62	1.38

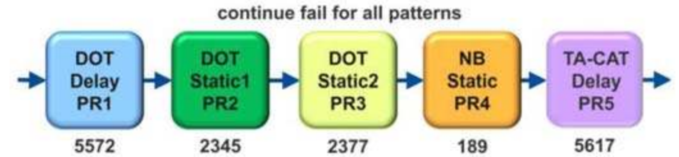

Fig. 27. Production test program flow.

TA-CAT: The fifth run is a TA-CAT pattern generation, targeting all cell-internal small delay detectable defects. This run generates *PR5*. For this run, the cat.udfm and the SDF file are read and all gross-delay detectable defects are filtered out.

The result from these five pattern generations is shown in Table II. The results of PR5 are shown in the first row of the table on purpose, because it results in the lowest TCA coverage. The TCA of the chip is the sum of the weighted delay plus static TCA, i.e., the sum of the TCA from PR1 plus PR2, which results in a total chip TCA of 505M ts. (291M ts plus 214M ts). The TCA from PR5 is a subset from the delay TCA. The defects related to PR2, PR3, and PR4 share the same TCA. More details on these TCA values can be found in [36].

As can be seen in Table II, a high static TC of 99.15% and TCA of 98.62% are achieved at the end of the PR4 generation. At the end of the PR2 generation, the TCA is about 2% lower, and it is increased to 98.61% by adding the 2377 static PR3 patterns, which are often considered redundant to the PR2 patterns.

The test results of this experiment shall provide evidence if PR3 patterns can be left out or not. The TCA coverage at the end of PR4 is just 0.01% higher than at the end of PR3, because the added TCA of the cell-neighborhood defects contributes only with 0.6M ts to the static total chip TCA of 198M ts.

B. Test Program Flow and Test Results

The five test pattern files PR1, PR2, PR3, PR4, and PR5, as explained above, have been implemented into the production test program and are all executed with continue on fail.

The test program flow is shown in Fig. 27. The numbers below the colored boxes list the number of patterns for each of the five pattern sets.

Fig. 28 shows the results in DPPM in a five category Venn diagram from testing 1 000 000 good parts.

The *uniquely* failing parts measured in DPPM are shown in Fig. 28 with *red* numbers.

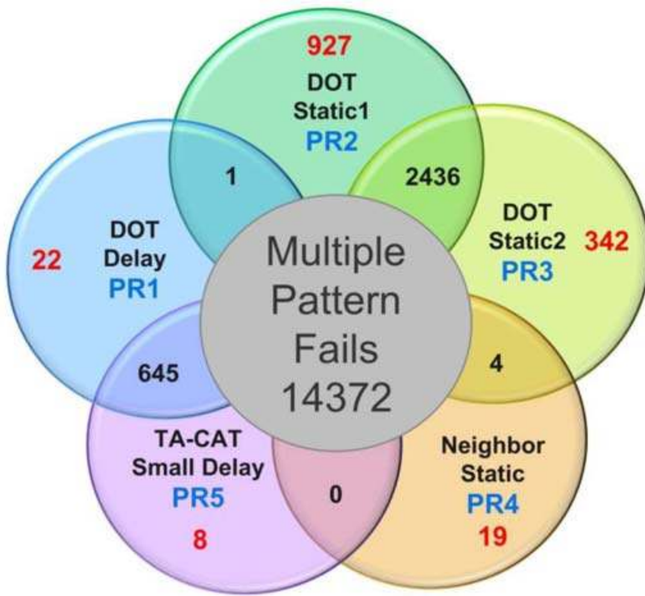

Fig. 28. Failing parts measured in DPPM.

The largest contribution to uniquely detected defects comes from static *PR2* patterns with *927 DPPM,* with a large overlap to static PR3 patterns with 2436 DPPM.

The *342 DPPM* uniquely failing parts from *PR3* is a significant number and is the proof that no compromise on static patterns should be done, and traditional industry practice on static patterns is misguided. The outgoing product quality was not affected significantly in the past, because the majority of these unique PR3 fails were also detected with MBIST patterns.

The third major contribution to unique detection comes from *PR1* patterns with *22 DPPM*. Please notice that there is a large overlap between PR1 and PR5 patterns with 645 DPPM. These are unique DOT-Delay defects that are not detected with any of the three static patterns.

Also shown in Fig. 28, there are *19 DPPM* uniquely failing the cell-neighborhood *PR4* patterns. This clearly indicates the necessity of including such tests.

Furthermore, there are *8 DPPM* uniquely failing the TA-CAT *PR5* patterns, which is again proof that small-delay defects need to be targeted explicitly with the TA-CAT ATPG.

VI. Highest Quality With Lowest Test Costs

Although top-off patterns and dedicated patterns for fault models of interest demonstrate incremental improvements from each new model, the generation of static and delay patterns can be optimized when ATPG runs are not done for each fault model in isolation, but when all UDFM files for the different fault models are read all together. Then only one static ATPG run and two delay ATPG runs need to be done as shown in Fig. 29.

Doing the ATPG runs as shown in Fig. 29, the total number of static plus delay patterns will be significantly smaller than doing ATPG runs for each fault model in isolation.

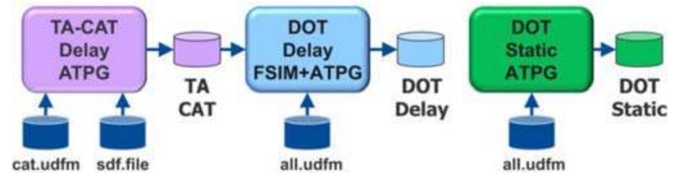

Fig. 29. Highest quality ATPG flow.

TABLE III
Chip Level Coverage Results

DOT type	#pat	#faults	TCA [ts]	FC [%]	RE [%]	TC [%]	TCA [%]	UC+AU [%]
TA-CAT	5617	4.5M	30M	80.61	1.09	81.94	80.86	19.14
Delay	3008	25.9M	291M	88.58	2.44	91.01	92.34	7.66
Static	4835	22.9M	214M	96.32	2.76	99.22	98.61	1.38

TA-CAT: This run targets all cell-internal small-delay detectable defects. The ATPG reads the SDF file and the CAT UDFM file, but filters out all gross-delay defects.

DOT-Delay: The second ATPG run is a top-off run on TA-CAT patterns, targeting all CAT-Delay, interconnect bridge delay, and interconnect open delay defects, that have not been marked as detected with a delay FSIM of TA-CAT patterns. For this FSIM and ATPG run, all four DOT view files are read: the cat.udfm, the open.udfm, the bridge.udfm, and the neighbor.udfm file.

DOT-Static: The third ATPG run is a static ATPG run from scratch, targeting all CAT-Static, interconnect bridge static, interconnect open static, and cell-neighborhood static defects. For this ATPG run, again all four DOT view files are read.

In Table III, the results from the described three chip level DOT ATPG runs are shown.

As can be seen in Table III, the number of static patterns did not change very much compared to the individual static ATPG runs as done in the production test runs shown in Fig. 26 and Table II, but the sum of TA-CAT plus DOT-Delay patterns is just 8625, whereas the sum of PR1 plus PR5 as used in the production test is 11 189 patterns. A second comparison can be done to the sum of 11 640 top-off patterns without any TA-CAT patterns as published in [36], when doing the individual fault model related delay ATPG runs.

Summing up the static and delay TCA as listed in Table III gives a total chip TCA of 505M ts, of which 211.0M ts are detected with static patterns, and 268.7M ts are detected with delay patterns, to report a combined static and delay total chip TCA coverage of 95%.

VII. Conclusion

An overview of physical defects was given, and details about the generation of DOT views have been presented to target those physical defects explicitly by an ATPG tool.

We have shown with test system results from a 14-nm FinFET chip that CAT patterns detect a huge amount of 4300 DPPM which are otherwise not detected with traditional SA, TDF, and all functional production test patterns.

We have shown on 156 SLT rejects that all parts passed traditional tests, but failed both SLT and CAT patterns. In addition, a very clear $V_{\min}$ strength improvement was achieved from TA-CAT patterns versus TDF patterns and resulted in a reduction of 300 DPPM on top of reductions achieved by CAT patterns and all other production tests. This resulted in the decision at Intel to make DOT patterns plan of record (POR), which means that each new design has to be tested with DOT patterns.

The presented results from the 160-nm automotive design show that the much more accurate TCA coverage deviates from traditional TC in both directions. In some cases, TCA coverage is higher than TC, whereas in other cases TCA coverage is lower than TC. But as the likelihood of defects is taken into account by TCA coverage, we now have a coverage figure that more realistically indicates how well all physical defects are covered, which opens up the ability to use this coverage for better estimates of quality. As a result of that, test patterns can now be optimized for the first time based on TCA and not by simply counting the number of faults.

High volume production test results from one million good automotive parts clearly demonstrate the efficacy of the DOT patterns and also show that the common practice of generating static top-off patterns, on top of delay patterns, and leaves defective parts undetected. Higher quality is therefore achieved by generating static patterns from scratch. The production test results from the automotive chip also indicate that cell-neighborhood patterns and as well TA-CAT patterns are needed to achieve zero DPPM.

Acknowledgment

The authors thank Wilfried Redemund, Maija Ryynaenen, and Juergen Schmerberg for their assistance, valuable discussion, implementations, and insight over the course of this project.

References

[1] K. C. Y. Mei, "Bridging and stuck-at faults," *IEEE Trans. Comput.*, vol. C-23, no. 7, pp. 720–727, Jul. 1974.

[2] F. J. Ferguson and T. Larrabee, "Test pattern generation for realistic bridge fault in CMOS ICs," in *Proc. IEEE Int. Test Conf. (ITC)*, 1991, pp. 492–499.

[3] P. Engelke, I. Polian, J. Schloeffel, and B. Becker, "Resistive bridging fault simulation of industrial circuits," in *Proc. Design Autom. Test Europe (DATE)*, 2008, pp. 628–633.

[4] J. Rearick and J. H. Patel, "Fast and accurate CMOS bridging fault simulation," in *Proc. IEEE Int. Test Conf. (ITC)*, 1993, p. 3.

[5] J. A. Waicukauski, E. Lindbloom, B. K. Rosen, and V. S. Iyengar, "Transition fault simulation," *IEEE Des. Test Comput.*, vol. 4, no. 2, pp. 32–38, Apr. 1987.

[6] H. Cox and J. Rajski, "Stuck-open and transition fault testing in CMOS complex gates," in *Proc. IEEE Int. Test Conf. (ITC)*, 1988, pp. 688–694.

[7] S. Chakravarty, A. Jain, N. Radhakrishnan, E. Savage, and S. Zachariah, "Experimental evaluation of scan tests for bridges," in *Proc. IEEE Int. Test Conf.*, 2002, pp. 509–518.

[8] V. Krishnaswamy, A. Ma, and P. Vishakantaiah, "A study of bridging defect probabilities on a pentium (TM) 4 CPU," in *Proc. IEEE Int. Test Conf.*, 2001, pp. 688–695.

[9] J. Shen, W. Maly, and F. Ferguson, "Inductive fault analysis of CMOS integrated circuits," *IEEE Des. Test Comput.*, vol. 2, no. 6, pp. 1181–1194, Dec. 1985.

[10] B. Lee, H. Li, L.-C. Wang, and M. Abadir, "Pattern selection for testing of deep submicron timing defects," in *Proc. IEEE Int. Test Conf.*, vol. 2, 2005, pp. 1060–1065.

[11] R. Putman and R. Gawde, "Enhanced timing-based transition delay testing for small delay defects," in *Proc. IEEE Very Large Scale Integr. Test Symp.*, 2006, pp. 336–342.

[12] S. K. Goel, K. Chakrabarty, M. Yilmaz, K. Peng, and M. Tehranipoor, "Circuit topology-based test pattern generation for small-delay defects," in *Proc. 19th IEEE Asian Test Symp. (ATS)*, 2010, pp. 307–312.

[13] X. Lin *et al.*, "Timing-aware ATPG for high quality at-speed testing of small delay defects," in *Proc. IEEE Asian Test Symp.*, Nov. 2006, p. 8.

[14] M. Yilmaz, K. Chakrabarty, and M. Tehranipoor, "Interconnect-aware and layout-oriented test-pattern selection for small-delay defects," in *Proc. IEEE Int. Test Conf.*, 2008, pp. 1–10.

[15] J. Dworak *et al.*, "Defect-oriented testing and defective-part-level prediction," *IEEE Des. Test Comput.*, vol. 18, no. 1, pp. 31–41, Jan./Feb. 2001.

[16] F. Hapke *et al.*, "Defect-oriented cell-aware ATPG and fault simulation for industrial cell libraries and designs," in *Proc. IEEE Int. Test Conf. (ITC)*, 2009, pp. 1–10.

[17] F. Hapke *et al.*, "Cell-aware production test results from a 32-nm notebook processor," in *Proc. IEEE Int. Test Conf. (ITC)*, 2012, pp. 1–9.

[18] F. Hapke, M. Keim, T. Herrmann, and M. Reese, "Improving failure analysis for cell-internal defects through cell aware technology," in *Proc. 39th ISTFA*, 2013.

[19] F. Hapke *et al.*, "Cell-aware test," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 9, pp. 1396–1409, Sep. 2014.

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

[21] P. C. Maxwell, F. Hapke, M. Ryynaenen, and P. Weseloh, "Bridge over troubled waters: Critical area based pattern generation," in *Proc. IEEE Eur. Test Symp. (ETS)*, 2017, pp. 1–6.

[22] M. Sachdev and J. Pineda de Gyvez, *Defect-Oriented Testing for Nano Metric CMOS VLSI Circuits*. 2nd ed. Dordrecht, The Netherlands: Springer, 2007.

[23] Y. Huang and C. Lu, "Methodology of generating dual-cell-aware tests," in *Proc. IEEE VLSI Test Symp. (VTS)*, 2017, pp. 1–6.

[24] P. de Gyvez and J. Jess, "On the definition of critical areas for IC photolithographic spot defects," in *Proc. 1st Eur. Test Conf.*, 1989, pp. 152–158.

[25] Z. Stamenkovic, "Algorithm for extracting integrated circuit critical areas associated with point defects," *Int. J. Electron.*, vol. 77, no. 3, pp. 369–376, 1994.

[26] F. Corsi, S. Martino, C. Marzocca, R. Tangorra, C. Baroni, and M. Buraschi, "Critical areas for finite length conductors," *Microelectron. Rel.*, vol. 32, no. 11, pp. 1539–1544, 1992.

[27] C. H. Stapper, "Modeling of defects in integrated circuit photolithographic patterns," *IBM J. Res. Develop.*, vol. 28, no. 4, pp. 461–475, Jul. 1984.

[28] C. H. Stapper, "Modeling of integrated circuit defect sensitivities," *IBM J. Res. Develop.*, vol. 27, no. 6, pp. 549–557, Nov. 1983.

[29] T. Iizuka, M. Ikeda, and K. Asada, "Exact wiring fault minimization via comprehensive layout synthesis for CMOS logic cells," in *Proc. Int. Symp. Signals Circuits Syst.*, 2004, pp. 377–380.

[30] G. A. Allan and A. J. Walton, "Efficient critical area measurements of IC layout applied to quality and reliability enhancement," *Microelectron. Rel.*, vol. 37, no. 2, pp. 1825–1833, 1997.

[31] W. Maly, "Modeling of lithography related yield losses for CAD of VLSI circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. CAD-4, no. 3, pp. 166–177, Jul. 1985.

[32] A. V. Ferris-Prabhu, "Modeling the critical area in yield forecasts," *IEEE J. Solid-State Circuits*, vol. SSC-20, no. 4, pp. 874–877, Aug. 1985.

[33] F. M. Goncalves, I. C. Teixeira, and J. P. Teiceira, "Realistic fault extraction for high-quality design and test of VLSI systems," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, 1997, pp. 29–37.

[34] B. Benware, C. Schuermyer, T. Herrmann, and M. Sharma, "Determining a failure root cause distribution from a population of layout-aware scan diagnosis results," *IEEE Des. Test Comput.*, vol. 29, no. 1, pp. 8–18, Feb. 2012.

[35] P. Maxwell, F. Hapke, and H. Tang, "Cell-aware diagnosis: Defective inmates exposed in their cells," in *Proc. 21st IEEE Eur. Test Symp. (ETS)*, 2016, pp. 1–6.

[36] F. Hapke and P. Maxwell, "Total critical area based testing," in *Proc. IEEE Int. Test Conf. (ITC)*, 2018, pp. 1–10.

[37] W. Howell *et al.*, "DPPM reduction methods and new defect oriented test methods applied to advanced FinFET technologies," in *Proc. IEEE Int. Test Conf. (ITC)*, 2018, pp. 1–10.

Friedrich Hapke (Member, IEEE) received the Diploma degree in electrical engineering from the University of Applied Sciences Hamburg, Hamburg, Germany.

He is a consultant with Mentor, A Siemens Business, Hamburg, Germany. His primary focus is in research and development of new methods and tools for supporting a complete defect-oriented test method, including cell-aware testing and cell-internal failure diagnosis. Before his retirement, he was the Director of engineering with Mentor, Hamburg, in 2018. He has authored and coauthored many publications and holds over 20 patents in the area of DFT.

Will Howell (Member, IEEE) received the M.S. degree from the University of Southern California, Los Angeles, CA, USA.

He is a Principal Engineer with Intel Corporation, Sunnyvale, CA, USA. He has 23 years of extensive experience in each of structural, functional, and system logic test solidifying his belief that Structural Defect Oriented Test methods are required to achieve quality. His primary focus is utilizing these techniques to enhance test programs for TT/DPM improvements. He presented results from Cell Aware and DOT methods on Intel designs at ITC 2018.

Peter Maxwell (Life Fellow, IEEE) received the M.Sc. degree in physics from the University of Auckland, Auckland, New Zealand, and the Ph.D. degree in electrical engineering and computer science from the Australian National University, Canberra, ACT, Australia.

He works with ON Semiconductor, Santa Clara, CA, USA, where he is responsible for test and DFT for CMOS image sensors. He was one of the first to widely publish industrial data on test effectiveness. His interests include test methodologies, DFT, and their application to yield improvement, diagnosis, and quality.

Dr. Maxwell has served on numerous IEEE conference committees, including the Program Chair of the VLSI Test Symposium and International Test Conference.

Edward Brazil (Senior Member, IEEE) received the B.Eng. degree in electronic engineering from the National University, Maynooth, Ireland. He is currently pursuing the M.Sc. degree in artificial intelligence with the University of Limerick, Limerick, Ireland.

He is a Senior DFT Engineer with Intel Corporation, Sunnyvale, CA, USA. He has over 15 years of experience in semiconductor manufacturing test and yield, design for test, and architecture. His primary focus is on test quality.

Srikanth Venkataraman (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign, Urbana, IL, USA.

He is an Architect and Strategic Planner for solutions with the confluence of Design, Manufacturing and Test, Intel Corporation, Hillsboro, OR, USA. He has built and managed groups responsible for developing and successfully deploying solution for diagnosis, debug and test used all across Intel, and adopted by the broader industry. He has over 100 journal/conference publications, patents, tutorials, and book chapters. His work has spanned the areas of Test, DFT, Yield, Fault diagnosis, Design Verification, Si Debug, DFM, and CAD.

Rudrajit Dutta (Member, IEEE) received the B.Tech. degree in electrical engineering with the Indian Institute of Technology Kharagpur, Kharagpur, India, and the Ph.D. degree from the University of Texas at Austin, Austin, TX, USA.

He is a Staff Software Engineer with Intel Corporation, Hillsboro, OR, USA. His main focus is CAD tool development for Yield Analysis.

Andreas Glowatz (Senior Member, IEEE) received the Diploma degree in electronics and computer science from the University of Applied Sciences Hamburg, Hamburg, Germany.

He is a Principal Engineer and the Head of Engineering with Mentor, A Siemens Business, Hamburg. He has over 30 years experience in Research and Development of DFT tools with the main focus on ATPG, test compression, and cell-aware test. He holds several patents and has coauthored many publications.

Anja Fast (Member, IEEE) received an apprenticeship as a hotel specialist in Hamburg, Germany. She worked as a Management Assistant and as an Independent Consultant with Mentor, Hamburg. She is a Technical Assistant with Mentor, A Siemens Business, Hamburg. She has coauthored several professional publications about cell-aware test and defect oriented test methods.

Janusz Rajski (Life Fellow, IEEE) received the Master of Science degree in electrical engineering from the Gdańsk University of Technology, Anantapur, India, and the Ph.D. degree in electrical engineering from the Poznań University of Technology, Poznań, Poland.

He is a Vice President of engineering with Mentor, A Siemens Business, Hamburg, Germany, and joined Mentor Graphics in 1995. He has published more than 240 IEEE research papers and is co-inventor of more than 100 U.S. patents.

Dr. Rajski papers won many prestigious awards, including two Donald Pederson Best Paper Awards. In 2003, he was awarded the prestigious title of "Professor of Science" by the President of Poland. In 2018, he received the Siemen's Lifetime Achievement Award for his extensive contributions to DFT.