Journal of Electronic Testing (2019) 35:823–838
https://doi.org/10.1007/s10836-019-05832-w

Multi-PVT-Point Analysis and Comparison of Recent Small-Delay Defect Quality Metrics

Omar Al-Terkawi Hasib[1] · Yvon Savaria[1] · Claude Thibeault[2]

Received: 31 March 2019 / Accepted: 23 September 2019 / Published online: 26 December 2019

Abstract

The current method of designing robust digital circuits requires running analysis and simulations over multiple process-voltage-temperature (PVT) points to meet the design specifications. However, in small-delay defect (SDD) testing, the computation of the SDD test quality uses a single PVT point. This makes it less accurate to describe the test quality for chips that operate under a different point. In this paper we explore the idea of calculating the SDD quality metric over multiple PVT points using multiple SDD test quality metrics including our previously proposed metric, the weighted slack percentage (WeSPer). The results are obtained by running extensive simulations with a CMOS 28nm technology and calculating the different SDD test quality metrics under 54 different PVT points, 3 test speeds and 2 different types of test patterns for 14 benchmark circuits. The results are then analyzed and compared with respect to the test-escape window size. The comparison shows that WeSPer is the most responsive SDD test quality metric to the change in the test-escape window size. Since the simulation of 54 PVT points and the delay information extraction can be lengthy, this paper also shows two methods of estimating WeSPer across all PVT points by either predicting the results using only 3 PVT points or by considering the worst case scenario.

Keywords Small-delay defects · Delay testing · Test quality metrics · Multi-PVT-point analysis

1 Introduction

Delay testing is essentially a test of whether signal transitions can travel through a circuit within a given time frame (clock period). The goal of such test is to guarantee that the tested circuit will function correctly at the rated speed. However, selecting the proper input stimuli to cover all the fault sites in a circuit requires careful considerations. When targeting smaller delay defects, the delay of gates and their effect on the signal transition time through the circuit must be considered. Catching these small-delay defects (SDDs) in newer technologies while considering the varying nature of circuit delays with process, voltage and temperature variations (PVT variations), is a challenging task.

In the typical flow of digital circuit design, the correct behavior of implemented circuits is verified across PVT variations by simulating the design under all possible PVT points. However, when it comes to assessing the quality of an SDD test, the circuit delay information needed to calculate those metrics are typically extracted from a single PVT point [10, 17, 20, 24, 32, 38]. Furthermore, it is not clear which PVT point is the best PVT point to use for SDD test quality assessment, and to the best of our knowledge the sensitivity of SDD test quality metrics to the change of PVT simulation point was never discussed before in the literature. Thus, in this paper, we will compare the results of computing some of the common SDD test quality metrics along with our previously proposed SDD test metric, the Weighted Slack Percentage (WeSPer) [15], over an extensive 54 PVT point simulation for 14 benchmark

Responsible Editor: X. Li

✉ Omar Al-Terkawi Hasib
omar.al-terkawi-hasib@polyml.ca

Yvon Savaria
yvon.savaria@polymtl.ca

Claude Thibeault
claude.thibeault@ets.ca

[1] The Department of Electrical Engineering,
École Polytechnique de Montréal, Montreal, QC, Canada

[2] The Department of Electrical Engineering, École de
Technologie Supérieure, Montreal, Canada

circuits, 3 test clock speeds and 2 different types of test patterns.

When changing the PVT simulation point, the change in circuit delays will result in a change in the average test-escape window size for the applied delay test. The test escape window is defined as the difference between the tested SDD size to the smallest effective SDD size. The presented comparison will discuss the sensitivity of the selected metrics to the change of PVT simulation points and report the correlation of the metric value change to the change of the average test-escape window size.

Since there are many PVT points to cover in modern technologies, an interesting result that will be presented is how the SDD test quality changes with each of the PVT parameters, and whether this change can be predicted to save simulation time. Moreover, a brief discussion of which PVT point provides the worst-case for testing SDD is presented.

The rest of the paper is organized as follows. Section 2, presents a brief general review on SDD testing. Then in Section 3, the most commonly known metrics for SDD in the literature are reviewed and their pros and cons are discussed. Section 4 presents the proposed metric model and formulation. The computation flow and tools setup for extracting the results from multi-PVT point simulation is presented in Section 5. A detailed discussion on the extracted results of the metrics for all available PVT points is presented in Section 6. Finally, this paper concludes in Section 7.

2 Brief Review on SDD Testing

Standard delay testing is based on one of two delay fault models: the path delay fault model (PDF) or the transition delay fault model (TDF) [5, 23, 34, 40]. A PDF based test is an extensive search of all possible paths in a circuit for delay defects. SDDs cannot escape detection if indeed all paths are tested. However, as the circuit complexity grows, the time to apply such a test can grow exponentially, and the conditions to correctly sensitize paths become more complicated [8, 30]. On the other hand, a TDF based test applies test vectors to excite a transition (rising and falling) through each node in a circuit. It is less time consuming to apply a TDF based test than a PDF one. However, since not all paths are tested, it is possible to miss an SDD that can cause a chip to fail in the field.

To test for SDDs, most researchers have focused on building tests that enhance TDF based tests. The main idea in most of the literature is to reduce the slack window in which an SDD can escape during testing. This is either done by enhancing the automatic test pattern generator (ATPG) to sensitize longer paths (in terms of delay) [2, 3, 7, 12, 14, 19, 20, 27, 28, 31, 36, 43, 44], or by applying a faster-than-at-speed test (FAST) where the test clock is faster than the system clock [1, 6, 11, 18, 25, 26, 37, 45]. Both approaches have their challenges in terms of algorithmic complexity (in the timing aware ATPG case) or in terms of dealing with implications of using a faster-than-at-speed test clock (in the FAST case) [21].

SDD test quality metrics are used to assess the effectiveness of a delay test in catching SDDs. The quality of a classical TDF based test is assessed by computing the transition delay fault coverage (TDFC). TDFC is merely the percentage of tested nodes to total number of nodes. It is not adequate for representing the quality of SDD tests since it does not depend on the size (net added delay) of the tested delay defect. Hence, when developing new SDD test techniques, researchers tend to formulate their own forms of test quality metrics, usually used for fault dropping and pattern screening, that they report normally, along with the TDFC. This makes it difficult to compare different test methodologies with each other. However, there are some generic SDD quality metrics that are presented in the literature and sometimes used in well-known test software [10, 17, 20, 24, 32, 38]. These metrics can be partitioned into two categories: statistical metrics and non-statistical metrics. The latter are simpler and faster to compute, since they do not require any statistical information on the occurrence of delay defects in the targeted fabrication technology (a delay defect distribution). Statistical SDD test quality metrics, on the other hand, require a delay defect distribution that is used to give weight (importance) to the size of the test-escape window in each tested path. One should note that those metrics were defined for one process-voltage-temperature (PVT) point, and as it is well-known, the delay in CMOS circuits varies with PVT variations. The detection of SDDs in the presence of such variations has been a challenge that some researchers tried to solve [28, 36, 42, 44].

In [15], we reviewed some of the recent and known SDD test quality metrics under typical PVT conditions (i.e. Single PVT point) and listed the advantages and disadvantages of those metrics with respect to common SDD test practices. Namely, the use of a timing-aware ATPG and FAST. We also proposed a new non-statistical flexible metric called the weighted slack percentage (WeSPer) that targets some of the shortcomings of other metrics in the case of a FAST. The equation that defines WeSPer was formulated to be expandable depending on the needs of the test engineer using confidence level (CL) multipliers. For example, we introduced a CL multiplier that penalized the case of *over-testing*; that is testing for a delay defect that is too small to fail the circuit under normal operation.

3 SDD Test Quality Metrics Review

For completeness, this section will review and analyze some of the most significant SDD test quality metrics that can be found in the literature. This review will help in understanding the results discussed later in this paper in relation with the sensitivity of different metrics to PVT point change and their correlation to the average test-escape window. As mentioned earlier, SDD test quality metrics can be divided into two categories: statistical and non-statistical. Statistical metrics require the delay defect distribution for the target fabrication technology. This distribution is used to calculate the probability of having a delay defect of a certain size escape the applied test. Although the idea of using a delay defect distribution is compelling, it is often impractical. Such a distribution is not provided by manufacturers as part of the process design kits (PDKs) and all the current statistical metrics have used the same distribution extracted from empirical data that are a result of an extensive test of more than 70,000 chips fabricated in the 180 nm technology [22]. Thus, without an accurate delay defect distribution specific to the technology and the foundry used, no added value comes from using such distribution. On the other hand, non-statistical metrics are simpler to calculate and do not require a delay defect distribution. Obviously, non-statistical metrics do not consider the probability of occurrence of certain sizes for delay defects in a technology when evaluating the quality of the SDD test. As a consequence, for a single path in a circuit, as long as the ratio of the tested delay (tested slack) to the longest delay (smallest slack) is constant, the value of those metrics will not change regardless of the size of the test-escape window.

To minimize redundancy with later parts of the paper when explaining the formulation of SDD test quality metrics, we will define here some of the common terms used. Firstly, from here on, we will simply use the term *metric* to mean a *SDD test quality metric*. Note that all the metrics that are discussed in this paper are for TDF based tests. Under the TDF model, we will define a true path as a functional path that a transition can travel through starting from a circuit input (primary input or scan flip-flop out), passing through the targeted fault site and arriving at a circuit output (endpoint, namely, a scan flip-flop input). Moreover, in the equations of this paper, we will denote the delay of the longest true path passing through a fault site as PD_{LT}, whereas the delay of the longest activated (testable) path under a test pattern that passes through a fault site will be denoted as PD_{LA}. Also, T_{sys} represents the period of the system clock and T_{test} is the period of the test clock. Note that the algorithms for finding the longest true path delay and the activated path delay are not within the scope of this work. This work uses industrial test tools to extract such information for metric calculations.

Lastly, when considering the variations in path delay due to the change of PVT points, it is important to notice that the true and tested paths could change. Hence, a change in the metric value is to be expected. It is also important to note that the experiments in [22] did not consider voltage or temperature variations. Thus the delay defect distribution that is used in statistical metrics does not take those factors into account.

3.1 Delay Test Coverage

The *Delay Test Coverage (DTC)* is a simple non-statistical metric that, consequently, does not use a delay defect distribution function [20]. The DTC is simply calculated as the average ratio between PD_{LA} and PD_{LT} for each fault in the circuit:

$$DTC = \frac{1}{N}\sum_{f=1}^{N}\frac{PD_{LA}}{PD_{LT}} \times 100\% \tag{1}$$

Where N is the total number of faults in the circuit under test (CUT). It can be seen from Eq. 1 that the DTC does not consider the test clock frequency, the slack of the path nor the delay defect distribution. It implicitly assumes that all delay defect are equally likely to happen regardless of their size. This makes the DTC less accurate, however, it is fast and easy to calculate. Moreover, since the DTC does not use any slack measurement in the calculation, it is only accurate when the test clock period and the system clock period are equal (i.e. at-speed testing), and therefore, it cannot be used for FAST.

3.2 Statistical Delay Quality Level

The *Statistical Delay Quality Level (SDQL)* [32] is calculated based on the probability of SDD test-escapes. This probability is calculated under the *Statistical Delay Quality Model (SDQM)* based on the range of SDD values that are not covered by a test pattern set when applied with a test clock period (T_{test}). To quantify the probability of having a test-escape, a delay defect distribution is extracted from the probability of having a delay defect in a certain technology, which is then fitted into an equation of the form of $F(s) = a\,e^{-\lambda s} + b$ (Fig. 1). Moreover, two slacks are defined as follows:

$$S_{mgn} = T_{sys} - PD_{LT} \tag{2}$$

$$S_{det} = T_{test} - PD_{LA} \tag{3}$$

As discussed in [32], S_{mgn} splits the delay defect distribution into two regions. Any defect smaller than S_{mgn}

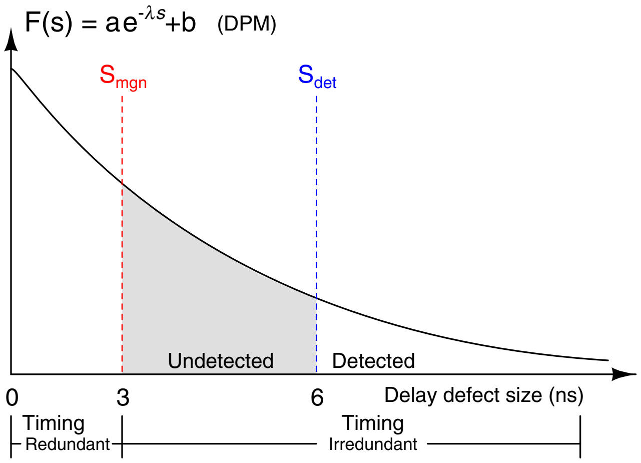


Fig. 1 Example of delay defect distribution divided into regions based on the value of S_{mgn} and S_{det}

is called a timing redundant delay defect, which is not considered as an SDD. Whereas delay defects larger than S_{mgn} are considered as SDD (timing irredundant). S_{det} splits the timing irredundant region into two subregions. Delay defects that are larger than S_{det} are detected by the test, whereas those smaller than S_{det}, but larger than S_{mgn}, are considered as undetected (test-escapes). Under those definitions, the SDQL for each fault is calculated as the area under the curve between the two slacks (as shown in Fig.1) and for the complete circuit it is calculated as:

$$SDQL = \sum_{i=1}^{N} \int_{S_{mgn\_i}}^{S_{det\_i}} F(s)\,ds \ (DPM) \tag{4}$$

Where N is the total number of faults in the CUT, $F(s)$ is the *delay defect distribution function of a delay defect of size* s and the unit is defects per million (DPM). The SDQL targets undetected timing irredundant delay defects. Thus, it does not include timing redundant delay defects into the calculation and the lower the SDQL the better. This metric has two shortcomings. First, because the SDQL is not a normalized number it gives an unfair advantage to circuits with a lower number of faults over those with a higher number of faults [10]. This also means that it is difficult to know what is good level of SDQL for a circuit without reference. Second, in a FAST, it is possible for S_{det} to become less than S_{mgn}. In that case it is not clear how to calculate this metric, however, in our computations, a perfect score (of 0) is given to this case.

3.3 Small Delay Defect Coverage

Another statistical metric that we will discuss is the *Small Delay Defect Coverage (SDDC)* [10]. The SDDC measures the ratio between the probabilities that a defect is detected by the test and the probability that a defect is timing irredundant (can cause a failure during normal operation). In reference to Fig. 1, the SDDC can be calculated as follows:

$$SDDC = \frac{1}{N}\sum_{i=1}^{N} \frac{\int_{S_{det\_i}}^{\infty} F(s)\,ds}{\int_{S_{mgn\_i}}^{\infty} F(s)\,ds} \times 100\% \tag{5}$$

Where N is the total number of faults and $F(s)$ is the *delay defect distribution function of a delay defect of size* s. In the case of faster-than-at-speed testing, if the SDDC term for any fault is > 1, SDDC is split into three parts: 1) The normal SDDC metric; 2) SDDC$_{DPM}$, which is the same as SDDC except that the score of each fault is limited to a max of 1; 3) to represent the percentage of reliability testing (testing of timing redundant faults), SDDC$_{EFR}$ is defined as the difference between SDDC and SDDC$_{DPM}$. The disadvantage of such definition is that faults that are over-tested (result > 1) are given a perfect score (of 1) in the SDDC$_{DPM}$. This is a misleading result, because over-tested faults can lead to false rejects (i.e. chips falsely identified as faulty). Moreover, although SDDC addresses many of the shortcomings of the previously mentioned metrics, it does not take into consideration process variations on the path tested, cross-talk, hazards or power supply variations, which can cause inaccuracies in fault identification and slack measurements.

3.4 Quadratic Small Delay Defect Coverage

The *Quadratic Small Delay Defect Coverage* ($SDDC^Q$) is a non-statistical metric that was presented in [10] as an alternative to the statistical SDDC when a delay defect distribution is not available. The SDDC^Q is calculated with the following equation:

$$SDDC^Q = \frac{1}{N}\sum_{i=1}^{N} \frac{(PD_{LA} + T_{sys} - T_{test})^2}{(PD_{LT})^2} \times 100 \tag{6}$$

Where N is the total number of faults. Notice that the SDDC^Q takes into account the period of the test clock and system clock that is normally used. It was shown in [10] that SDDC^Q is proportional to the delay defect size. This gives it an advantage over the DTC. However, it is difficult to interpret the significance of the SDDC^Q equation.

The results in [15] have shown that SDDC^Q is incompatible with faster-than-at-speed testing where it sometimes reports invalid values ($\gg$ 100%).

4 Review of the Proposed Metric

With the amount of uncertainty and the varying nature of delays in CMOS circuits, formulating a perfect SDD test quality metric is almost impossible. Including more information in a metric can make it more accurate, but also more complex to calculate. The current non-statistical metrics are easy to calculate with the basic delay information that is available in any PDK. However, if an accurate delay defect distribution is known, using a statistical metric to estimate the quality of an SDD test is a better choice. All the previously reported metrics used only the delay defect size or the delay of the path to assess the SDD test quality. Statistical metrics used a delay defect distribution that is not included in standard PDKs and that is not a function of voltage and temperature. Also, none of the reported metrics considered the accuracy of the delay information used in the calculation, nor the validity of the measured test results.

For example, if a FAST is applied to improve the TDF test quality, it risks over-testing the circuit. Meaning that it can fail a good circuit because it tested for SDDs smaller than the smallest SDD that can fail the circuit in normal operation. Even in a reliability testing aspect, if the FAST is applied to catch reliability defects without accounting for PVT variations on the tested paths, false detection can occur. Moreover, in real circuits, the delay of a path not only varies with environmental conditions (such as supply voltage and temperature), but can also depend on other factors (such as the chip layout and input vector [4, 19, 29]). The accuracy of the detection of a delay fault can also depend on the nature of the path. For example, hazards or reconvergence in the tested paths could mask a fault or falsely report one [19, 40]. In addition, since many of the SDD test methods that consider the variations in circuit delays target only one delay influencing factor at a time, it is difficult to know if the selection of a path based only on one factor would produce positive results when other factors are taken into account. For all those reasons, we proposed in [15] a flexible metric that is more compatible with FAST and can be adapted to different SDD test methods according to the information available at hand. In this section we will review the formulation of the proposed metric and propose a new method of making it compatible with delay defect distributions. In later sections, we will also discuss the sensitivity and predictability of this metric with PVT changes and compare it to other SDD test quality metrics. Note that, although the formulation of the metric has been previously discussed in [15], the results and comparisons across multiple PVT points are completely new and unique to this paper.

4.1 The Ideal SDD Test Model

To better understand the choices made in formulating the proposed metric, this section will state some definitions and discuss the notion of a hypothetical ideal SDD test. Under the TDF model, let us assume that:

1. The exact delay information of a path can be known.
2. The path delay does not vary as a function of time or any other factor (no process variations, cross-talk, etc.).
3. The longest true path through each fault site is known.
4. The combination of the applied pattern and test clock can test exactly the size of the smallest effective delay defect (SEDD) for each fault.

If a delay test is applied in an environment where these assumptions hold, we call it an *ideal SDD test*. Moreover, we define the SEDD as the smallest possible slack, for each fault, under normal operating conditions. In other words, the SEDD is the difference between the system clock period and the longest true path through a given fault site.

Figure 2 illustrates the concept of the ideal SDD test on an example circuit (Fig. 2a) with the delay of the annotated paths shown in Fig. 2b. The ideal test would be achieved if the marked fault is activated through the longest path that starts from I_4 and is observed at O_2 with T_{sys}. If for any reason the ATPG tool did not sensitize the longest path, the other options would be to test the marked fault through either path A or B. Since the delay of these paths are different, the ideal test can still be achieved if the test clock period is appropriately adapted to detect the SEDD. In the example of Fig. 2, this is shown by setting the test clock period to T_{idealA} or T_{idealB} respectively. Notice that if the pattern applied activated both path A and path B, and if the test was to select path B and a test clock with a period of T_{idealB}, then the result should be observed at O_1, whereas O_2 should be masked. Moreover, when applying a test pattern with a certain test clock, any endpoint that observes a negative slack should be masked.

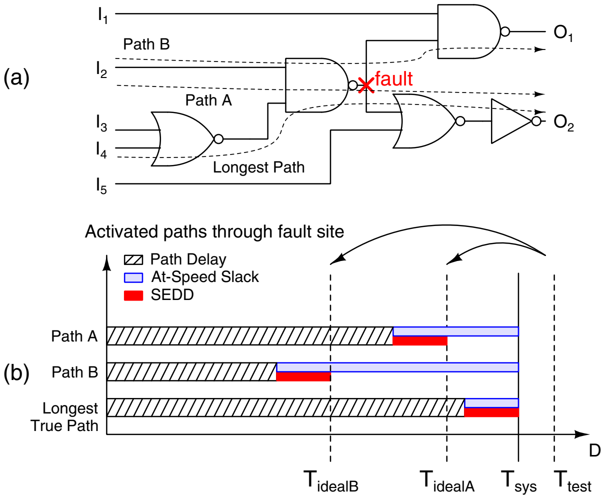


Fig. 2 Example circuit with selected paths shown in **a** along with a path delay diagram shown in (b). In **b** The path delay diagram shows the smallest effective delay defect (SEDD) size of the marked fault and slack of each path along with their ideal test clock period

Finally, the idea of the proposed metric revolves around estimating the difference between the applied delay test and the proposed hypothetical ideal SDD test model. This is done by assessing the difference between the tested slack and the SEDD, as well as representing any nonideality in a real delay test by a number (a percentage).

4.2 Weighted Slack Percentage

The proposed metric is called *Weighted Slack Percentage (WeSPer)*. As the name implies, this metric is based on path slack ratios weighted by confidence level (CL) multipliers that are defined according to the available information. WeSPer is calculated as follows:

$$WeSPer = \frac{1}{N}\sum_{i=1}^{N} f_i \, CL_i \times 100\% \tag{7a}$$

$$f_i = \frac{T_{sys} - PD_{LT_i}}{T_{test} - PD_{BA_i}} \quad \text{for} \quad T_{test} > PD_{BA_i} \tag{7b}$$

Where N is the number of faults in the CUT, T_{sys} is the system clock period, T_{test} is the test clock period and, as defined earlier, PD_{LT_i} is the longest true path through a fault site. PD_{BA_i} is the best activated path through a fault site and CL_i is the confidence level of testing through a fault site. The confidence level concept will be explained later in this section.

Basically, WeSPer is the percentage of the size of the tested defect ($T_{test} - PD_{BA_i}$) in comparison to the size of the SEDD ($T_{sys} - PD_{LT_i}$) weighted by the confidence level of the test (CL). The slack terms that are used in WeSPer are very similar to what is defined in SDDC and SDQL with one key difference; in WeSPer, the tested slack is selected based on the best activated path rather than the longest. The best activated path is the path that maximizes the term $f_i \, CL_i$. In the reviewed metrics (in Section 3), the best activated path is the longest one because it maximizes the metric. This is not necessarily true in WeSPer, since it depends on how the CL multipliers are calculated. Also, notice that f_i is defined only for positive slacks ($T_{test} > PD_{BA_i}$). Any endpoint that has a negative slack should be masked, and should not be considered as a valid observation point for that pattern.

The CL_i terms are equations that are defined by the test engineer according to the available information. This term gives a great deal of the flexibility to the proposed metric. The equation of the CL should be a weighting factor that rewards what the test engineer considers good and penalizes what he considers bad for the test. For example, in the case of hazards, whether an applied test pattern creates hazards or not at the endpoints, a CL for hazards should reflect that. When multiple confidence level multipliers are defined, CL_i is formulated as follows:

$$CL_i = \begin{cases} \prod_{k=1}^{M} C_k & \text{for} \quad M > 0 \\ 1 & \text{for} \quad M = 0 \end{cases} \tag{8}$$

Where $0 \leq C_k \leq 1$ is one type of confidence level multipliers and M is the total number of defined confidence level multipliers. If no CL multiplier is defined, CL is set to 1 and WeSPer can still be calculated. Thus, one can evaluate the SDD test quality while considering multiple factors that influence the test quality, rather than looking at disjoint quality indicators.

4.3 WeSPer for Faster-Than-At-Speed

In this section, we will describe a CL multiplier for the case of over-testing in a FAST. In this context, over-testing is defined as the case when the combination of the test pattern applied and test clock speed used test a delay defect size ($T_{test} - PD_{BA}$) that is less than the size of the SEDD ($T_{sys} - PD_{LT}$). Based on this definition, and the fact that $PD_{LT} \geq PD_{BA}$, over-testing can only occur when applying a FAST ($T_{test} < T_{sys}$).

Most metrics are not designed to measure the quality of an SDD test in the case of over-testing, and when they do, they assign a perfect score for that test case. This is the natural consequence of using faster-than-at-speed testing mainly as a reliability test, where the test is designed to catch the smallest delay defect possible. Hence, over-testing might be good as a reliability check, but if the goal is to catch regular SDDs, then it should be avoided since it tests for delay defects that cannot cause failure under normal operating conditions, and consequently, it could falsely label a good chip as defective. For that reason, a CL multiplier that penalizes over-tested faults was added to WeSPer. This CL multiplier (denoted by CL_{OT}) is defined as follows:

$$CL_{OT_i} = \begin{cases} 1 & \text{if} \quad f_i \leq 1 \\ (1/f_i)^2 & \text{if} \quad f_i > 1 \end{cases} \tag{9}$$

Where f_i is the same term defined in (7b). The term CL_{OT_i} was formulated to penalize over-testing with the same intensity as under-testing (testing SEDD with larger slacks). Moreover, to measure how much over-testing occurred when using a FAST clock, we define the *Total Over-testing Percentage (TOPer)* as follows:

$$TOPer = \frac{1}{N}\sum_{i}^{N} OPer_i \times 100\% \tag{10a}$$

$$OPer_i = \begin{matrix} 0 & \text{if} \quad f_i \leq 1 \\ 1 - (1/f_i) & \text{if} \quad f_i > 1 \end{matrix} \tag{10b}$$

Where f_i is the same term defined in (7b). Qualitatively, when calculating WeSPer with CL_{OT}, the best activated path is the path with the maximum delay that does not cause over-testing.

5 Multi PVT Point Computation Flow

To better understand the results that will be presented in the next section and the challenges that were faced to obtain those results, this section will discuss the computation flow and tools framework used in calculating the results. The reader should keep in mind that the goal is not to maximize the metrics by optimizing the test for each PVT point, but rather to look at the effects of PVT point change on the SDD test quality as reported by the selected metrics. Thus, the benchmark circuits synthesis and test patterns generation were done under an arbitrarily selected reference PVT point. By doing so, we guarantee that the same circuit structure and test patterns are used for all the PVT points, and that the only variable that affects the circuit path delays is the PVT point variation. Moreover, for each benchmark circuit, a single system clock is chosen based on the critical path of the slowest PVT point. This is automatically done by finding the critical path of each circuit under all PVT points and determining the longest one.

Figure 3 shows the computation flow diagram of the metrics for the selected benchmark circuits under all available PVT points. The first stage in the flow is to generate the Verilog netlist (using Synopsys Design Compiler [35]) and test patterns (using Mentor Tessent [13]) for all benchmark circuits for the reference PVT point. Stage 2 uses the circuits netlist and computes the worst critical path delay for each circuit under all PVT points. Using that delay, the system clock speed for each benchmark circuit is computed. Those two stages are relatively fast to process.

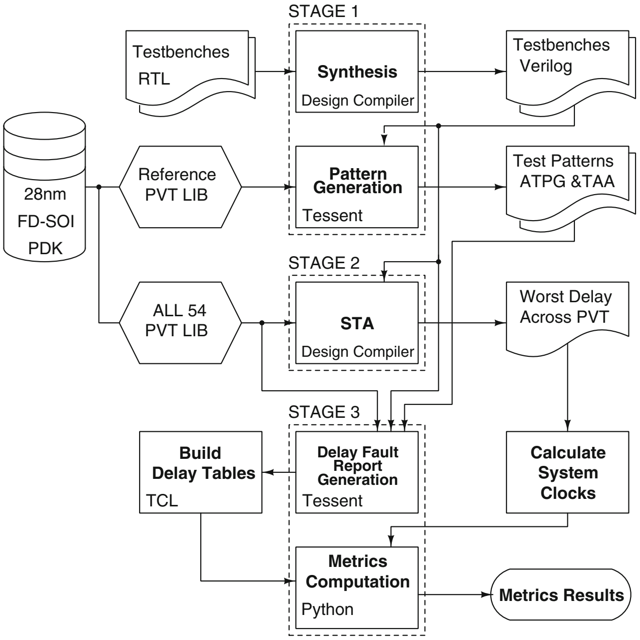


Fig. 3 The computation flow of the metrics for all 54 PVT points

Stage 3 involves a more tedious computation. In order to compute all the metrics, we need to find the true path delay, for both the longest path and the activated (tested) path, for each fault. A whole field of research is dedicated to finding the longest path for delay faults [33, 39, 41] and reviewing it is outside the scope of this work. Moreover, the proposed ideal test model of WeSPer requires finding the best activated path rather than the longest path due to the use of CL multipliers (as explained in Section 4.2). Since the calculation of WeSPer requires finding the best activated path for each fault, the extraction of such information from Tessent shell is not straightforward. By default, Tessent shell only reports the max activated path delay for each fault (for a whole set of patterns and for all possible endpoints), whereas WeSPer requires finding the activated true path delay information (for rising and falling transitions) through each fault site to every possible endpoint under each pattern. The delay information for each fault under each pattern to every possible endpoint is extracted from the tool by applying the pattern vectors one at a time while masking endpoints appropriately. The delay information is collected from Tessent shell fault reports when the timing engine is activated. These fault delay information are the ones that Tessent uses to compute the DTC. Since Tessent is not optimized to extract the delay information for each pattern and every endpoint separately, the generation of these fault reports is very time consuming. The complete flow for 54 PVT points and 14 benchmark circuits (listed in Table 1) took 3 months of run-time on an 8 cores processor. In the literature, there are other tools that are much more optimized and capable of extracting such information in a fraction of the time that we observed [33], however, they are not available to us.

From the extracted delay reports generated with Tessent shell, a TCL script builds a set of two delay tables. One is a table of the longest true path delay for each fault and the other is a table of the tested (activated) path delay for each fault under each pattern for each endpoint. Those two tables, along with the previously calculated system clocks, are used by a Python script to compute all the SDD test quality metrics that are discussed in the next section.

6 Results and Discussion

The following results were obtained from the computation flow presented in Section 5 under the 28nm FD-SOI technology from STMicroelectronics on a set of ISCAS-85, 74X-Series and ITC99 benchmark circuits. Figure 3 depicts the computation flow and indicates the tools used for each

Table 1 Benchmark circuit characteristics

Circuit	# of Faults	# of Endpoints	# of patterns ATPG	 TAA	TAA Average PD_A increase (%)	TDFC (%)
c17	34	2	7	8	4.21	100
b06	184	9	28	31	3.77	89.13
x74283	202	5	30	36	2.02	100
x74181	354	8	46	62	3.43	100
b08	532	21	81	94	2.19	96.05
c432	812	7	85	85	23.34	98.77
c499	934	32	100	124	2.80	99.14
b07	1032	44	109	116	8.89	96.80
c880	1556	26	85	99	14.78	100
c1908	2048	25	161	190	10.24	99.66
c2670	3268	140	125	175	7.73	97.71
c3540	4862	22	231	250	3.22	97.90
c5315	8028	123	144	179	11.02	99.41
c7552	10298	108	191	228	3.75	98.72

step. The version of the 28nm FD-SOI PDK used contains 54 PVT points. The reference PVT point used in the flow is selected to be at the typical process point, operating at 0.9V and 25°C. The selected reference PVT point is somewhat in the center of the PVT space of this technology. This helps to show the effects of faster and slower PVT points on the results. To compute the metrics, the system clock period is chosen to be 1.25 times the delay of the worst critical path of the benchmark circuit under the slowest PVT point. In addition, three test clock speeds are used: slow, at-speed and fast. In reference to the system clock period, the slow, at-speed and fast test clock periods are 1.1, 1 and 0.9 times the system clock period, respectively. Finally, the results are computed for two sets of patterns: conventional TDF patterns, and timing-aware TDF patterns. For brevity, these pattern sets will be referred to as as ATPG patterns and TAA (timing-aware ATPG) patterns, respectively. Table 1 shows the characteristics of the benchmark circuits used in the calculation. For each circuit, this table lists the number of faults, endpoints and patterns in each pattern set for each circuit. The sixth column in the table lists the average percentage of increase in the activated path delay (PD_A) when applying the TAA pattern set, and the last column lists the TDFC for each circuit. The TDFC value is the same for both the ATPG and TAA pattern sets.

In a previous paper [15], we focused on one PVT point and proved the validity of WeSPer with the results that were presented. In this section we will reemphasize the previously obtained conclusions with our full results of 54 PVT points simulations and discuss three more important aspects: metric sensitivity, predictability and worst-case scenario. Due to the huge number of results obtained, it is impossible to list all the results in few pages. We will present the complete set of results for one circuit, one pattern set and three test clock speed and follow with statistical information that summarizes the full set of obtained results. Moreover, we will only show the results for DTC, SDDC, SDDC$_{DPM}$ and WeSPer. This is because it was proved in [10, 15] that the SDQL is not good for comparison, since it is not normalized over the number of faults, and the SDDC^Q can have illogical results ($\gg$ 100%) when the test clock is not at-speed with the system clock. Note that all the computations of WeSPer in this work include the over-test CL multiplier (CL_{OT}).

The calculation of the SDDC and SDDC$_{DPM}$ requires a delay defect distribution, since they are statistical metrics. Unfortunately, the delay defect distribution for the 28nm technology is not available to us and adapting this distribution from other technologies is not trivial. Therefore, for the sake of generating results for the statistical metrics, we used the same delay defect distribution expressed in Eq. 11 that is used in [10, 32] as an approximation:

$$F(s) = 1.58 \times 10^{-3} \times e^{-2.1s} + 4.94 \times 10^{-6}(DPM) \quad (11)$$

Where s is the delay defect size in nanoseconds. The TDFC will be reported in the results as it serves as the theoretical upper limit for the selected metrics. Whenever any of the reported metrics exceeds the TDFC, that metric is reporting an invalid value. In addition, to better understand the presented results and compare the metrics, we introduce a comparison measure called the *mean slack difference* (MSD). The MSD is the average difference between the minimum defect size detected by the test and the SEDD. It is calculated by averaging the absolute difference between the tested slack and the minimum slack for each fault over the complete circuit. In more qualitative terms, it is the average size of the test-escape (or over-test) window for a circuit measured in nanoseconds. The closer the MSD is to zero, the closer the test is to an ideal SDD test.

6.1 Single Benchmark Analysis

Figure 4 plots the results of the c5315 benchmark circuit with an ATPG pattern set versus each PVT point under the three test speed cases (slow, at-speed, fast). The Metrics (TDFC, DTC, SDDC, SDDC$_{DPM}$ and WeSPer) are plotted with respect to the left Y-axis. This plot shows how each of the selected metrics changes with the change of PVT points. For clarity, the PVT points were ordered from slowest to fastest based on the maximum activated path delay. The plot shows that the DTC value does not change with the change of the test clock speed, and has a change of less than 1% across all PVT points. For the SDDC and SDDC$_{DPM}$, their values get better and closer to 100% when the test clock speed changes from slow to at-speed. However, when the test clock is faster than the system clock (i.e. FAST), the SDDC value is no longer valid ($>$ TDFC and $>$ 100%) and the SDDC$_{DPM}$ value saturates in most PVT points. The SDDC and SDDC$_{DPM}$ are most sensitive to PVT changes under the slow test clock (7.3% variation) and least sensitive under the fast test clock ($<$ 5% variation). On the other hand, WeSPer is the most sensitive metric to clock speed change and PVT point change. As the test clock speed increases, WeSPer results improves if no over-testing happens. This is apparent when comparing the slow PVT points (e.g. at SS/0.6V/-40°C) across the test speed changes. However, at faster PVT points, some faults start to be over-tested and the penalization of the over-tested points shows in the figure as an overall decrease in WeSPer value when moving to faster PVT points. Similar to the SDDC metric, WeSPer is most sensitive to changes when the test clock speed is slow (15.6% variation for c5315), whereas it is less sensitive when using the fast clock speed (5.5% variation for c5315) due to the penalization of over-testing.

Figure 4 also shows the value of the MSDs (labeled *Slack Diff* and *WeSPer Slack Diff*) on the right Y-axis. There are two MSD values shown: the regular slack difference that is used with other metrics by considering the longest tested path delay through a fault site, and the WeSPer slack difference that uses the delay of the best tested (activated) path instead of the longest one (as discussed in Section 4.1). The regular slack difference and the WeSPer slack difference are only different when over-testing occurs. This is because the best activated paths, in the case of a fast clock, are often not the longest ones, but rather the ones that result in a test slack closest to the SEDD size without over-testing. As a consequence, in the fast test subplot of the c5315, the MSD measured in WeSPer is lower than the regular one used by the other metrics in all PVT points.

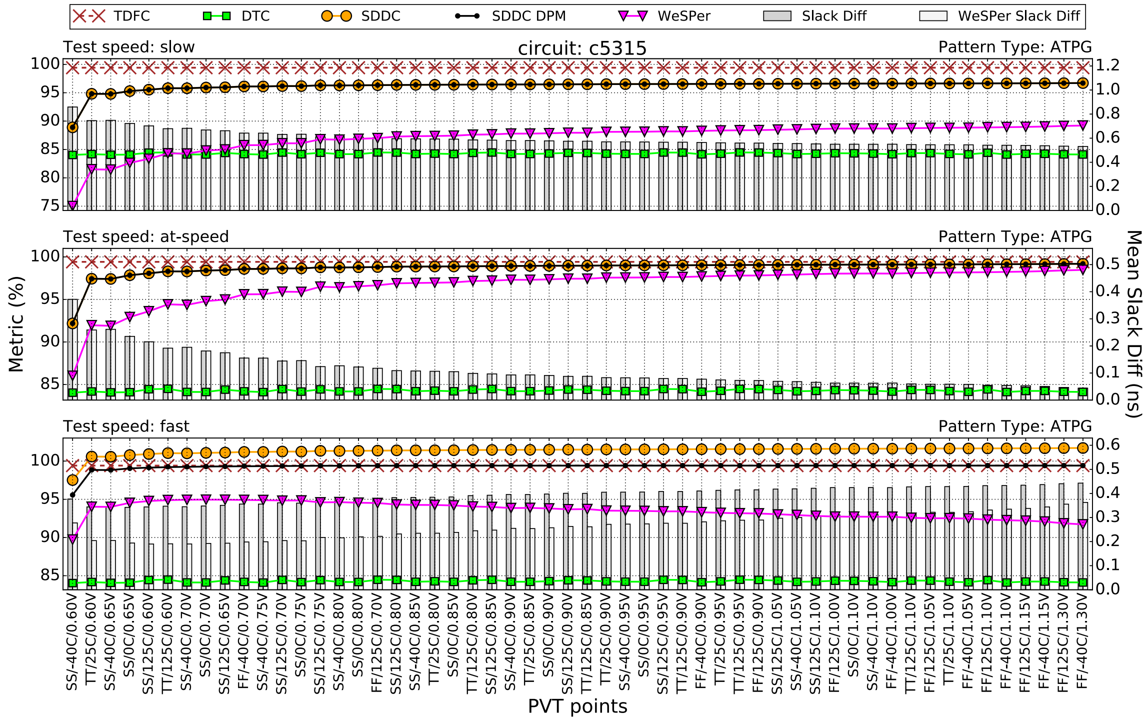


Fig. 4 Metric results for all available PVT points for the c5315. The PVT points are ordered from slowest to fastest point based on the longest activated path delay. The left Y-axis shows metric values while the right Y-axis shows the mean slack difference values. These results were obtained with a regular ATPG pattern set and are plotted for three test clock speeds: slow (top), at-speed (middle) and fast (bottom)

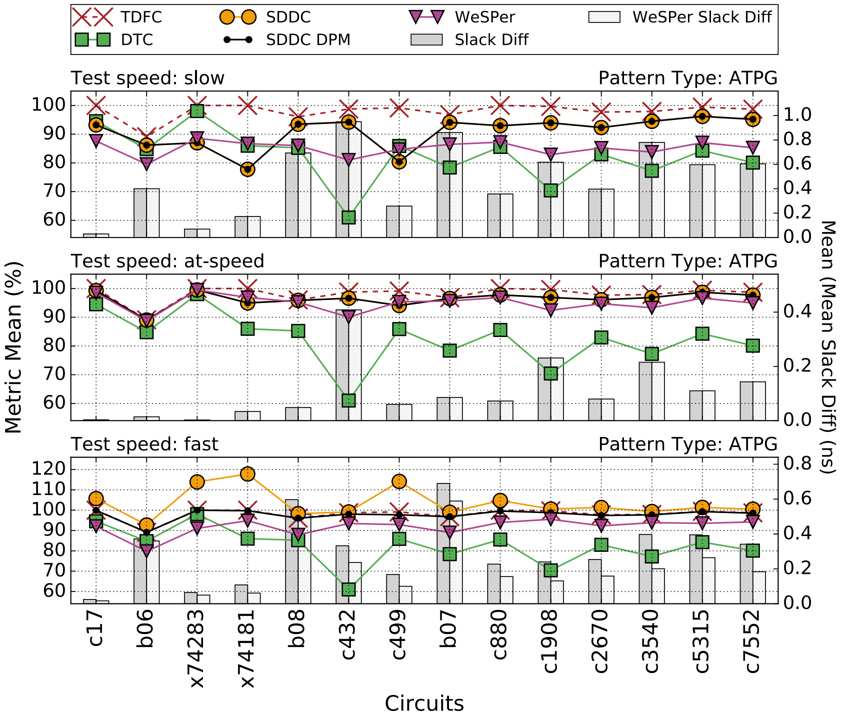


Fig. 5 Metrics results summary for a test applied on the selected benchmark circuit set using a regular ATPG pattern set for three test clock speeds: slow (top), at-speed (middle) and fast (bottom). The mean metric values across PVT points are plotted against the left Y-axis, whereas the right Y-axis plots the mean of the mean slack difference

Another interesting result can be seen by looking at the MSD results in the at-speed subplot of Fig. 4. Given that both the test clock and system clock speeds are equal and constant across PVT points in that subplot, changes in the MSD value (average test-escape window size) are mainly due to the difference between the longest path delay and the activated path delay for each fault. It can be seen here that, faster PVT points (e.g. a higher voltage point) produce smaller test-escape windows. This can be explained by the fact that faster PVT points produce tighter path delay distributions than slower ones [16].

6.2 Benchmark Circuits Results Summary

The conclusions from the results of the c5315 are also valid for the rest of the benchmark circuits. To summarize the results for all other benchmark circuits, the mean values of the results for each circuit across all PVT points will be shown. Figure 5 shows the mean of each metric and MSD results for each benchmark circuit across PVT for a conventional ATPG pattern set. In this figure, the results for all the metrics are plotted with respect to the left Y-axis, whereas the MSDs are plotted with respect to the right Y-axis. This figure shows how each metric reacts to the change of test clock speed for each of the benchmark circuits. The mean value of the DTC does not change with the test clock speed, and thus does not show the benefit of using a better test clock speed. On the other hand, all the other metrics show the advantage of using at-speed test clock over a slower test clock.

The case of faster-than-at-speed test clock is unique. The concept of penalizing over-testing was not introduced in metrics before WeSPer. Hence, in Fig. 5, SDDC and WeSPer handle that case differently. The SDDC and SDDC$_{DPM}$ values are exactly the same when no over-testing happens. However, in the case of a fast clock, the SDDC value goes above the TDFC value (the upper limit) which could be confusing or misleading. The authors of SDDC would refer to the SDDC$_{DPM}$ and SDD$_{EFR}$ in this case. It can be seen that the SDD_{DPM} saturates in all the selected circuits indicating a perfect SDD test, and the comparison of the quality of the test is left for the SDD$_{EFR}$ (results shown later in this section). In contrast, because WeSPer penalizes over-testing, it does not saturate. This makes it possible to indicate which test is better with a single metric. We should note that because, SDDC and WeSPer are built on different views on over-testing. The choice of which metric is better to use in this case depends on whether the goal of the test is to catch SDD that escape TDF based delay testing or to test for reliability defects as well.

As mentioned earlier, a TAA pattern set was also used to generate results on the benchmark circuits. Table 2 lists the percentage of improvement of using TAA pattern set results with respect to the ATPG pattern set results. The DTC is metric most affected by using a TAA pattern set. The improvement is in the DTC reaches up to 20.72%. Whereas the SDDC and WeSPer improvement is limited ($< 4\%$). When comparing this improvement to the improvement gained from using an at-speed clock versus a slow clock, it

Table 2 TAA pattern set results for metrics with respect to ATPG pattern Set result

		Slow test		At-speed test		Fast test		
Circuit	DTC	SDDC	WeSPer	SDDC	WeSPer	SDDC	SDDC_{DPM}	WeSPer
c17	4.53	0.56	1.01	0.56	1.16	0.56	0.10	0.32
b06	0.28	0.04	0.10	0.00	0.01	-0.07	0.00	0.21
x74283	1.17	0.30	0.27	0.30	0.31	0.30	0.02	0.05
x74181	4.19	1.49	0.86	1.60	0.99	1.70	0.26	0.12
b08	2.02	0.03	0.04	0.03	0.11	0.03	0.00	-0.10
c432	20.72	0.64	2.71	0.67	3.08	0.77	0.36	1.14
c499	4.21	1.30	1.00	1.57	1.16	1.85	0.52	0.28
b07	2.79	0.03	0.10	0.03	0.13	0.03	0.00	-0.10
c880	5.53	0.46	0.87	0.65	0.99	0.96	0.26	0.25
c1908	9.20	0.40	1.40	0.51	1.59	0.75	0.35	0.56
c2670	5.82	0.42	0.97	0.60	1.11	0.90	0.19	0.14
c3540	3.38	0.12	0.49	0.12	0.55	0.13	0.05	0.19
c5315	5.32	0.13	0.57	0.14	0.65	0.17	0.03	0.15
c7552	4.79	0.14	0.60	0.15	0.67	0.19	0.04	0.14

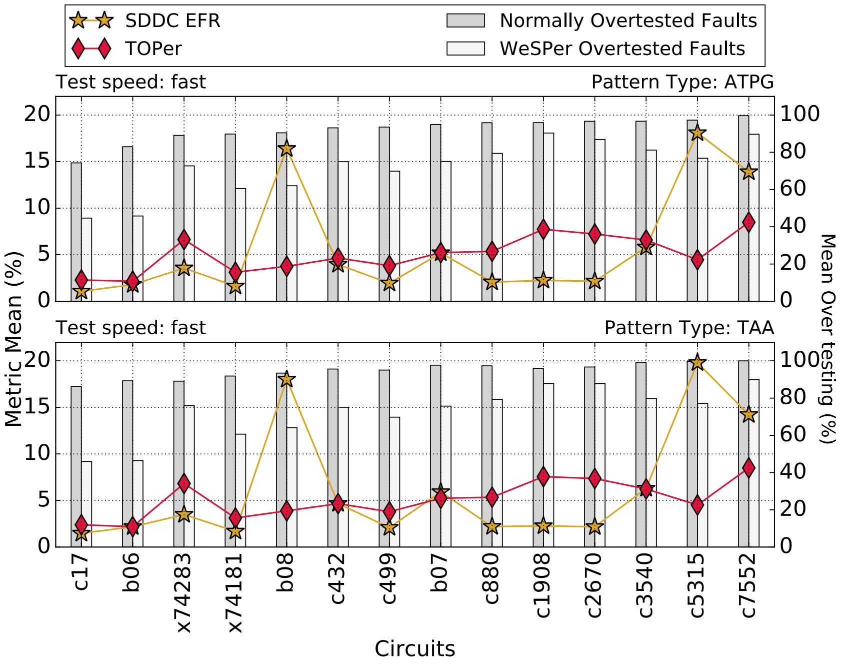
Values Represent The (%) of Improvement

suggests that using a better clock is better for the SDD test quality than using a TAA pattern set over an ATPG one.

To help analyze the over-testing affecting the results in Fig. 5 in the FAST case, Fig. 6 plots, with respect to the left Y-axis, the mean TOPer and mean SDDC_{EFR} across PVT points for each benchmark circuit and pattern set type. The right Y-axis of the figure reports the mean percentage of over-testing across all PVT points. It represents the percentage of over-tested faults to the total number of faults in a circuit. The two bars in Fig. 6 indicate the over-testing percentage when calculating other metrics and when calculating WeSPer. Remember that WeSPer looks for the best activated path and assumes that masking can be applied to over-tested endpoints when possible. This leads to having a lower over-tested fault percentage for all the benchmark circuits.

TOPer is used as a debugging metric to measure over-testing and explain the drop in WeSPer, if needed. In Fig. 6, a correlation can be seen between TOPer and the percentage of over-tested faults. In contrast, SDDC_{EFR} is

Fig. 6 Over-testing results summary. The lines show the mean value for the over-testing metrics across all PVT points with respect to the left Y-axis. The right Y-axis along with the bars show the mean percentage of over-tested faults with respect to the total number of faults for WeSPer and other metrics across all PVT points



a complementary metric to SDDC_{DPM}, the percentage of reliability testing in a test. It cannot be used as an over-testing measure as it does not correlate to the percentage of over-tested endpoint.

6.3 Metric Sensitivity

It is obvious that metrics need to be sensitive to any slight changes in the circuit that can affect the quality of the test. As explained earlier, the MSD represents the average size of the test-escape or over-test window. As seen in Fig. 4, the size of that window changes with PVT point variations, and thus, if a metric is accurate, its value should change accordingly. However, the sensitivity of a metric is considered a sign of accuracy only if it reacts to the window size change in a correct way. In this case, we expect metrics to be inversely correlated to the test-escape window size (i.e. MSD).

In Section 6.1, we briefly discussed the sensitivity of the metrics in the case of the c5315 circuit. The sensitivity of a metric is calculated as the percentage of change in the metric value across all PVT points. In Fig. 7, the sensitivity of the metrics across PVT points is shown for all benchmark circuits and test clock speeds under a conventional ATPG pattern set. Results from the TAA pattern set are very similar and are not shown here to avoid redundancy. The figures, show that the DTC exhibited the least changes (least sensitive), while WeSPer is the most sensitive in general.

To know whether the sensitivity of a metric is justified, we correlate the change in metric value to the change in the MSD value, since the MSD value reflects the average size of the test-escape window in a test. As mentioned earlier in Section 6, an ideal test would have a zero MSD. Thus, metrics should have negative correlation with the MSD in order to justify their sensitivity to PVT variations. In Fig. 8, the correlation between the metric change and MSD change across PVT points is shown. In terms of correlation to MSD change, the DTC has the worst overall correlation, while WeSPer has the best correlation. This leads to the conclusion that WeSPer is the most sensitive and most correlated to the test-escape window size among the selected metrics.

6.4 Predictability

As mentioned in Section 5, due to the involved extraction of detailed delay information for each fault in the circuit, computing WeSPer over a large set of PVT points can be time consuming and not practical, especially when a circuit design goes through several iterations. Thus, if the metric value can be predicted with acceptable accuracy from a smaller set of PVT points, rather than fully computed from the whole data set, a great amount of design time would be saved. To assess the predictability of WeSPer across the full set of PVT points, WeSPer is plotted under process, voltage and temperature variations separately. Figure 9 plots

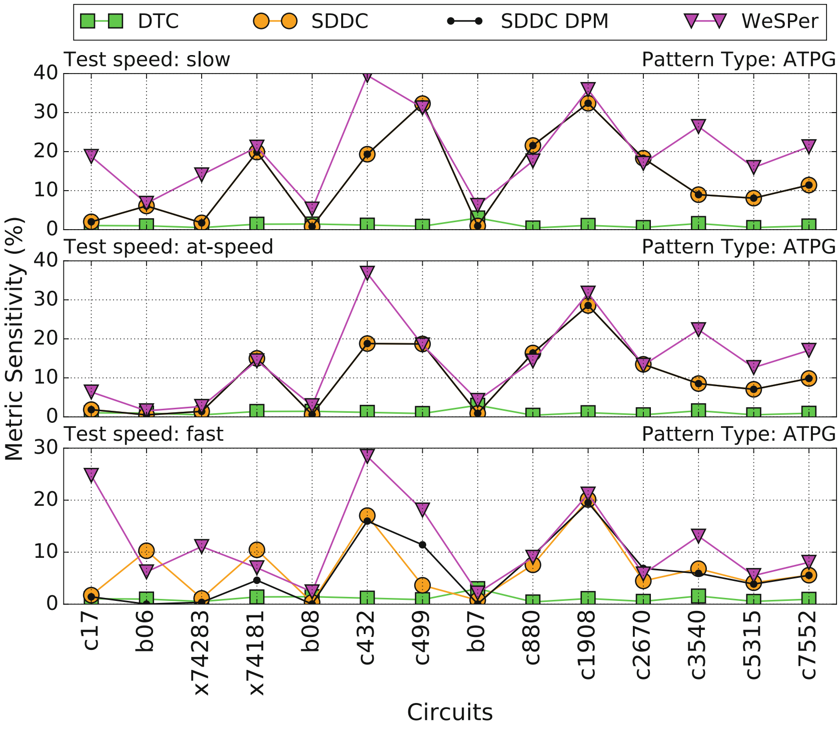


Fig. 7 The sensitivity of each metric across PVT points plotted for three test clock speeds when using the ATPG pattern set

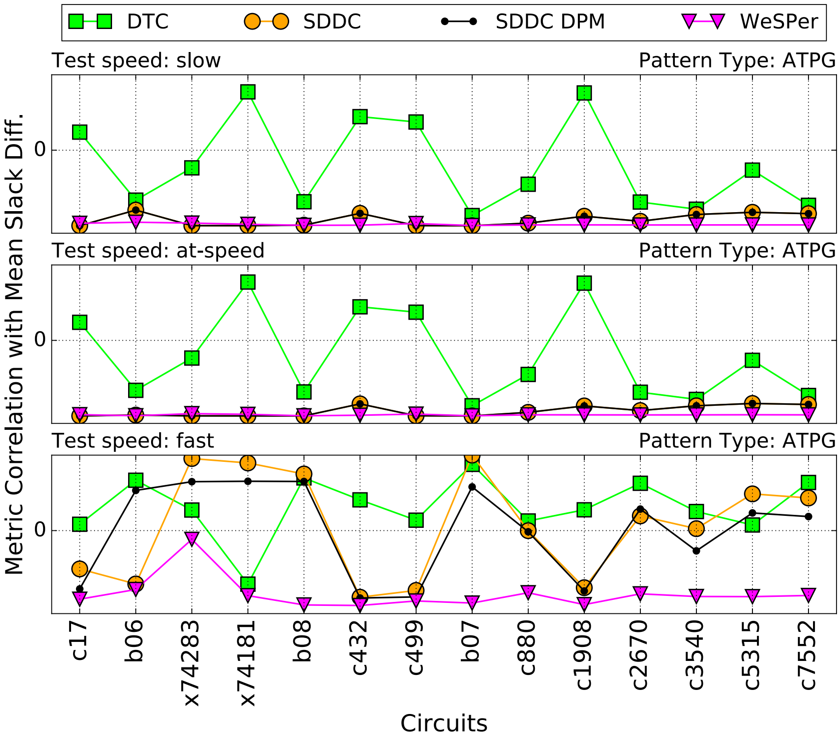


Fig. 8 Correlations between metric changes and MSD changes across PVT points for three test clock speeds: slow (top), at-speed (middle) and fast (bottom). Note that a negative correlation is expected, thus the closer the result is to -1 the better

WeSPer mean and standard deviation values against one dimension of PVT change at a time (process, temperature or voltage). For better visibility of the results, only a set of selected benchmark circuits are plotted (c432, c3540, c5315, c7552). The colored bars represents the mean value of WeSPer, whereas the smaller black bar show the standard deviation value. The plots show only the results for the conventional ATPG pattern set and slow test clock, however the following conclusion holds true for all cases.

In the case of process point variation, the mean WeSPer value has a slight linear increase of no more than 10% as the process point goes from slow to fast (Fig. 9). Also, the mean value of WeSPer is almost constant (less than 5% variation) with the change of temperature. However, under voltage variations, the mean value of WeSPer has up to 23% change. Thus, we will focus on predicting the change in the mean value of WeSPer with the change of the voltage point. It can be seen in Fig. 9 that, for all circuits, the change of the mean value of WeSPer with voltage follows the same exponential trend. In this case, the voltage curves can be predicted with 3 PVT points samples instead of 54. This saves drastically the computation time for metrics across a large number of PVT points. Figure 9 shows the prediction of the curve using three PVT points (typical point and $25°C$ and voltages $0.6V$ $0.8V$ and $0.95V$) in a dashed line. The curve fit was obtained using the following equation: $a - be^{-cV}$. The average percentage error between the fit and the mean data is less than 1.5% for all benchmark circuits. This means that we can predict WeSPer mean values with an acceptable accuracy for any circuit across all PVT variations with only 3 PVT points. This result holds true for all test speeds and pattern set types.

On the practical side, by looking at the standard deviation (black bar) in the voltage subplot in Fig. 9, it can be seen that higher voltage points have negligible variation in the WeSPer value across process and temperature changes. Thus, applying the SDD tests on circuits should be done with higher supply voltage since the test quality is consistent over all process points and temperatures.

6.5 Worst-Case PVT Point

Another conclusion that can be reached from observing the metric change with PVT variations in Fig. 4 and 9 is that the worst-case value for WeSPer is observed when testing at the slowest process point, the lowest voltage and the lowest temperature. This means that if a conservative estimation of the SDD test quality is to be made, it has to be done at that PVT point. In other words, if only one value needs to be reported as the SDD test quality, the worst case test should be reported since testing at any other PVT point will always yield a better result. Notice that this result could appear counterintuitive if one thinks that slower PVT points would imply longer path delays and thus better SDD test. The results shows that this is not the case. The test-escape window size is actually larger at those PVT points and thus the SDD test quality is worse. Also, note that the slowest PVT point happens at the lowest temperature, which maybe

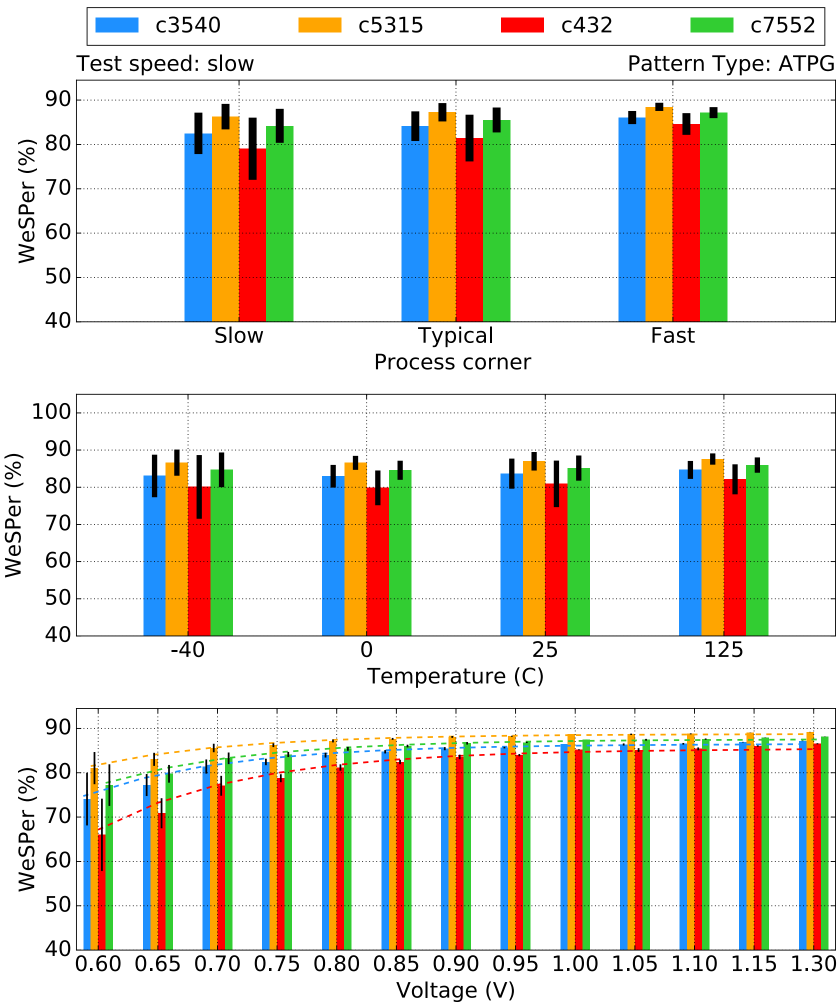


Fig. 9 WeSPer values with the change of process point, temperature and voltage for a selected set of circuits. The bars in the figure are ordered as in the legend and represent the mean value. The standard deviation value is represented by the smaller black bars. The dashed line shows that a very good curve fit can be obtained with only 3 PVT points using the equation $a - b\,exp(-c\,V)$

somewhat counterintuitive. This is due to an effect called the inverted temperature dependence [9], where at lower supply voltages, the circuits are slower at lower temperatures.

7 Conclusion

An extensive simulation campaign has been run to compute different SDD test quality metrics under 54 different PVT points, 3 test speeds and 2 different sets of test patterns for 14 benchmark circuits. By analyzing the response of the selected set of SDD test quality metrics across PVT points and different test clocks and test pattern sets, WeSPer proved to be a SDD test metric that is more sensitive and better correlated to the change in the test-escape window size than other metrics. In addition, analysis on each of the PVT parameters have shown that the worst-case SDD test quality is reported at the slowest process point, lowest voltage and lowest temperature. This can be counterintuitive, since one might think that PVT points with longer path delays should have a better SDD test quality. However, results also indicate that using higher voltages makes the SDD test quality almost independent of process and temperature variations. Thus, an SDD test is better applied in practice at the upper range of supply voltages. Finally, to save on simulation time, this work also presented a method of estimating the average SDD test quality (using WeSPer) across all PVT points using only 3 PVT points.

Acknowledgments The authors would like to thank CMC Microsystems and CMP for providing the CAD tools and access to a 28nm PDK, Octasic for providing partial financial support and scientific guidance, and the Natural Science and Engineering Research Council for providing partial funding.

References

1. Ahmed N, Tehranipoor M (2009) A novel faster-than-at-speed transition-delay test method considering ir-drop effects. IEEE Trans Comput-Aid Des Integ Circuit Syst 28(10):1573–1582. https://doi.org/10.1109/TCAD.2009.2028679

2. Ahmed N, Tehranipoor M, Jayaram V (2006) Timing-based delay test for screening small delay defects. In: Proceedings of the 43rd annual Design Automation Conference, ACM, pp 320–325
3. Bao F, Peng K, Tehranipoor M, Chakrabarty K (2013) Generation of effective 1-Detect TDF patterns for detecting small-delay defects. IEEE Trans Comput-Aided Design Integr Circuits Syst 32(10):1583–1594. https://doi.org/10.1109/TCAD.2013.2266374
4. Bao F, Tehranippor M, Chen H (2013) Worst-case critical-path delay analysis considering power-supply noise. In: Proceedings of the 22nd asian test symposium (ATS), 2013, pp 37–42. https://doi.org/10.1109/ATS.2013.17
5. Bushnell M, Agrawal V (2005) Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits (Frontiers in Electronic Testing). Springer, New York
6. Chakravarty S, Devta-Prasanna N, Gunda A, Ma J, Yang F, Guo H, Lai R, Li D (2012) Silicon evaluation of faster than at-speed transition delay tests. In: Proceedings of 2012 IEEE 30th VLSI Test Symposium (VTS), pp 80–85. https://doi.org/10.1109/VTS.2012.6231084
7. Chang CY, Liao KY, Hsu SC, Li J, Rau JC (2013) Compact test pattern selection for small delay defect. IEEE Trans Comput-Aided Design Integr Circuits Syst 32(6):971–975. https://doi.org/10.1109/TCAD.2013.2237946
8. Cheng KT, Chen HC (1996) Classification and identification of nonrobust untestable path delay faults. IEEE Trans Comput-Aided Design Integr Circuits Syst 15(8):845–853. https://doi.org/10.1109/43.511566
9. Dasdan A, Hom I (2006) Handling inverted temperature dependence in static timing analysis. ACM Trans Des Autom Electron Syst 11(2):306–324. https://doi.org/10.1145/1142155.1142158
10. Devta-Prasanna N, Goel S, Gunda A, Ward M, Krishnamurthy P (2009) Accurate measurement of small delay defect coverage of test patterns. In: 2009 Proceedings of international test conference, pp 1–10. https://doi.org/10.1109/TEST.2009.5355644
11. Fu X, Li H, Li X (2012) Testable path selection and grouping for faster than at-speed testing. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 20(2):236–247. https://doi.org/10.1109/TVLSI.2010.2099243
12. Goel SK, Devta-Prasanna N, Turakhia RP (2009) Effective and efficient test pattern generation for small delay defect. In: 2009 Proceedings of the IEEE 27th VLSI Test Symposium, IEEE, pp 111–116
13. Graphics M (2018) The tessent product suit
14. Gupta P, Hsiao MS (2004) Alaptf: a new transition fault model and the atpg algorithm. In: 2004 Proceedings of the International Test Conference, IEEE, pp 1053–1060
15. Hasib OAT, Savaria Y, Thibeault C (2016) Wesper: a flexible small delay defect quality metric. In: Proceedings of the IEEE 34th VLSI Test Symposium (VTS), pp 1–6. https://doi.org/10.1109/VTS.2016.7477266
16. Ikeda M, Ishii K, Sogabe T, Asada K (2007) Datapath delay distributions for data/instruction against pvt variations in 90nm cmos. In: 2007 Proceedings of the 14th IEEE International Conference on Electronics, Circuits and Systems, pp 154–157. https://doi.org/10.1109/ICECS.2007.4510953
17. Iyengar V, Rosen BK, Spillinger I (1988) Delay test generation. i. concepts and coverage metrics. In: 1988 International Test Conference on proceedings of the New frontiers in testing, pp 857-866. https://doi.org/10.1109/TEST.1988.207873
18. Kampmann M, Hellebrand S (2017) Design-for-fast: supporting x-tolerant compaction during faster-than-at-speed test. In: 2017 Proceedings of IEEE 20th International Symposium on Design and Diagnostics of Electronic Circuits Systems (DDECS), pp 35–41. https://doi.org/10.1109/DDECS.2017.7934564
19. Kruseman B, Majhi A, Gronthoud G, Eichenberger S (2004) On hazard-free patterns for fine-delay fault testing. In: 2004 Proceedings of the international test conference, pp 213–222. https://doi.org/10.1109/TEST.2004.1386955
20. Lin X, Tsai KH, Wang C, Kassab M, Rajski J, Kobayashi T, Klingenberg R, Sato Y, Hamada S, Aikyo T (2006) Timing-aware ATPG for high quality at-speed testing of small delay defects. In: 2006 Proceedings of the 15th asain test symposium, pp 139–146. https://doi.org/10.1109/ATS.2006.261012
21. Mahmod J, Millican S, Guin U, Agrawal V (2019) Special session: Delay fault testing - present and future. In: Proceedings of IEEE 37th VLSI Test Symposium (VTS), pp 1–10. https://doi.org/10.1109/VTS.2019.8758662
22. Mitra S, Volkerink E, McCluskey EJ, Eichenberger S (2004) Delay defect screening using process monitor structures. In: Proceedings of IEEE 22nd VLSI test symposium (VTS). IEEE Computer Society, pp 43–43
23. Mohammad TKC, Ke P (2011) Test and diagnosis for small-delay defects. Springer, New York
24. Park ES, Mercer M, Williams T (1992) The total delay fault model and statistical delay fault coverage. IEEE Trans Comput 41(6):688–698. https://doi.org/10.1109/12.144621
25. Pei S, Geng Y, Li H, Liu J, Jin S (2015) Enhanced lccg: a novel test clock generation scheme for faster-than-at-speed delay testing. In: Proceedings of the 20th Asia and South Pacific Design Automation Conference, pp 514–519. https://doi.org/10.1109/ASPDAC.2015.7059058
26. Pei S, Li H, Jin S, Liu J, Li X (2015) An on-chip frequency programmable test clock generation and application method for small delay defect detection 49:87–97. https://doi.org/10.1016/j.vlsi.2014.12.003
27. Peng K, Thibodeau J, Yilmaz M, Chakrabarty K, Tehranipoor M (2010) A novel hybrid method for sdd pattern grading and selection. In: 2010 Proceedings of the 28th VLSI Test Symposium (VTS), pp 45–50. https://doi.org/10.1109/VTS.2010.5469619
28. Peng K, Yilmaz M, Tehranipoor M, Chakrabarty K (2010) High-quality pattern selection for screening small-delay defects considering process variations and crosstalk. In: Proceedings of the design, automation & test in europe conference & exhibition (DATE), vol 2010, pp 1426–1431. https://doi.org/10.1109/DATE.2010.5457036
29. Peng K, Yilmaz M, Chakrabarty K, Tehranipoor M (2013) Crosstalk- and process variations-aware high-quality tests for small-delay defects. IEEE Trans VLSI Syst 21(6):1129–1142. https://doi.org/10.1109/TVLSI.2012.2205026
30. Pomeranz I, Reddy SM (1994) An efficient nonenumerative method to estimate the path delay fault coverage in combinational circuits. IEEE Trans Comput-Aided Des Integr Circuits Syst 13(2):240–250
31. Qiu W, Wang LC, Walker D, Reddy D, Lu X, Li Z, Shi W, Balachandran H (2004) K longest paths per gate (klpg) test generation for scan-based sequential circuits. In: 2004 Proceedings of the internationaltest conference, pp 223–231. https://doi.org/10.1109/TEST.2004.1386956
32. Sato Y, Hamada S, Maeda T, Takatori A, Nozuyama Y, Kajihara S (2005) Invisible delay quality-sdqm model lights up what could not be seen. In: 2005 IEEE Proceedings of the IEEE International Test Conference, pp 9–pp
33. Sauer M, Becker B, Polian I (2016) Phaeton: a sat-based framework for timing-aware path sensitization. IEEE Trans Comput 65(6):1869–1881. https://doi.org/10.1109/TC.2015.2458869
34. Smith GL (1985) Model for delay faults based upon paths. In: Proceedings of the international Test Conference, pp 342–351
35. Synopsys (2018) Design compiler

36. Tayade R, Abraham J (2008) Small-delay defect detection in the presence of process variations. Microelectronics J 39(8):1093–1100. https://doi.org/10.1016/j.mejo.2008.01.003, european Nano Systems (ENS) 2006
37. Tayade R, Abraham JA (2008) On-chip programmable capture for accurate path delay test and characterization. In: 2008 Proceedings of the IEEE International Test Conference, IEEE, pp 1–10
38. Tendolkar N (1985) Analysis of timing failures due to random AC defects in VLSI modules. In: 1985 Proceedings of the 22nd conference on design automation, pp 709–714. https://doi.org/10.1109/DAC.1985.1586020
39. Wagner M, Wunderlich HJ (2017) Probabilistic sensitization analysis for variation-aware path delay fault test evaluation. In: 2017 Proceedings of the 22nd IEEE European Test Symposium (ETS), pp 1–6. https://doi.org/10.1109/ETS.2017.7968226
40. Waicukauski J, Lindbloom E, Rosen BK, Iyengar V (1987) Transition fault simulation. IEEE Des Test Comput 4(2):32–38. https://doi.org/10.1109/MDT.1987.295104
41. Xiang D, Li J, Chakrabarty K, Lin X (2013) Test compaction for small-delay defects using an effective path selection scheme. ACM Trans Des Autom Electron Syst 18(3):44:1–44:23. https://doi.org/10.1145/2491477.2491488
42. Xiang D, Shen K, Bhattacharya BB, Wen X, Lin X (2016) Thermal-aware small-delay defect testing in integrated circuits for mitigating overkill. IEEE Trans Comput-Aided Des Integr Circuits Syst 35(3):499–512. https://doi.org/10.1109/TCAD.2015.2474365
43. Xu D, Li H, Ghofrani A, Cheng KT, Han Y, Li X (2014) Test-quality optimization for variable-detections of transition faults. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 22(8):1738–1749. https://doi.org/10.1109/TVLSI.2013.2278172
44. Yilmaz M, Chakrabarty K, Tehranipoor M (2010) Test-pattern selection for screening small-delay defects in very-deep submicrometer integrated circuits. IEEE Trans Comput-Aided Design Integr Circuits Syst 29(5):760–773. https://doi.org/10.1109/TCAD.2010.2043591
45. Yoneda T, Hori K, Inoue M, Fujiwara H (2011) Faster-than-at-speed test for increased test quality and in-field reliability. In: Proceedings of the 2011 IEEE International Test Conference, pp 1–9. https://doi.org/10.1109/TEST.2011.6139131

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Omar Al-Terkawi Hasib received the B.Sc. Degree in electrical engineering from Kuwait University, Kuwait, in 2006. He also received the M.A.Sc. and PhD Degrees in microelectronics from Ècole Polytechnique de Montréal, Montréal, QC, Canada in 2011 and 2019, respectively. He is currently working at Cadence design systems as an analog circuit designer. He has previously worked on analog/mixed-signal design, switch-capacitors DC-DC converters and charge pumps design, development of EDA optimization tools for analog circuits, design of analog interfaces to digital buses and small-delay defect testing for self-timed microprocessors. He has 10 years of experience working with EDA tools for analog, digital and RF circuits.

Yvon Savaria received the B.Ing. and M.Sc.A in electrical engineering from Ècole Polytechnique Montréal in 1980 and 1982 respectively. He also received the Ph.D. in electrical engineering in 1985 from McGill University. Since 1985, he has been with Polytechnique Montréal, where he is currently professor in the department of electrical engineering. He has carried work in several areas related to microelectronic circuits and microsystems such as testing, verification, validation, clocking methods, defect and fault tolerance, effects of radiation on electronics, high-speed interconnects and circuit design techniques, CAD methods, reconfigurable computing and applications of microelectronics to telecommunications, aerospace, image processing, video processing, radar signal processing, and digital signal processing acceleration. He is currently involved in several projects that relate to aircraft embedded systems, green IT, wireless sensor network, virtual network, computational efficiency and application specific architecture design. He holds 16 patents, has published 138 journal papers and 430 conference papers, and he was the thesis advisor of 160 graduate students who completed their studies. He was program co-chairman of ASAP'2006 and the general co-chair of ASAP'2007. He has been working as a consultant or was sponsored for carrying research by Bombardier, CNRC, Design Workshop, DREO, Ericsson, Genesis, Gennum, Huawei, Hyperchip, ISR, LTRIM, Miranda, MiroTech, Nortel, Octasic, PMC-Sierra, Technocap, Thales, Tundra and VXP. He is a member of the Regroupement Stratégique en Microélectronique du Québec (RESMIQ), of the Ordre des Ingénieurs du Québec (OIQ), and was a member of CMC Microsystems Board since 1999 and chairman of that board from 2008 to 2010. He was awarded in 2001 a Tier 1 Canada Research Chair (www.chairs.gc.ca) on design and architectures of advanced microelectronic systems that he held until June 2015. He also received in 2006 a Synergy Award of the Natural Sciences and Engineering Research Council of Canada.

Claude Thibeault received his Ph.D. from École Polytechnique de Montréal, Canada. He is now with the Electrical Engineering department of École de technologie supérieure, where he serves as full professor. His research interests include design and verification methodologies targeting ASICs and FPGAs, defect and fault tolerance, radiation effects, as well as IC and PCB test and diagnosis. He holds 14 US patents and has published more than 140 journal and conference papers. He has been member of different conference program committees, including VLSI Test Symposium, for which he was program chair in 2010-2012, and general chair in 2014 and 2015.