IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 September 2023, accepted 25 September 2023, date of publication 2 October 2023,
date of current version 24 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3321323

RESEARCH ARTICLE

Optimal Test Clock Frequency Based Test Option Generation for Small Delay Defects

PRATHIBA MUTHUKRISHNAN AND SIVANANTHAM SATHASIVAM, (Senior Member, IEEE)
School of Electronics Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India

Corresponding author: Sivanantham Sathasivam (ssivanantham@vit.ac.in)

The work was supported by the Vellore Institute of Technology, Vellore, India. Prathiba Muthukrishnan receives her research fellowship from the University Grants Commission, New Delhi, India.

ABSTRACT Small delay defects (SDD) based test escapes are caused by the nature of transition delay fault (TDF) ATPG, which propagates the fault effect along the shorter path in the interest of run time. However, owing to the benefits of a lesser pattern count and complexity, TDF ATPG is the most feasible option for delay testing. Faster than at-speed testing enhances the likelihood of detecting SDD. To generate the optimal test option and prevent over or under-testing the circuit, it is necessary to select the appropriate test clock period. The recently introduced Weighted slack percentage (WeSPer) metric is used in this article to identify the best SDD test option. The method's benefit is combined with the proper selection of the optimal clock frequency to test the SDD. This article proposes an optimal test clock period selection method by considering the size of the smallest possible delay defect size that can fail the circuit during at-speed operation and the fault's propagation delay. The proposed method is applied to the set of ISCAS89 and ITC99 benchmark circuits to evaluate its effectiveness.

INDEX TERMS Faster-than-at-speed testing, fault models, nanoscale devices, small delay defects, test quality metric, VLSI testing.

I. INTRODUCTION

Modern ICs have drastically improved with the advent of deep-submicron technology. As the technology node shrinks, manufacturing defects are inevitable. However, with the advancement of the fabrication process, these defects cause timing failures rather than catastrophic failures. Any physical defect that affects the signal propagation delay of the circuit is termed a delay defect which can be classified as a large delay defect or a small delay defect (SDD). The major challenge in modern ICs is the SDDs which are difficult to detect [1]. The process variations that affect oxide thickness, interconnect length, etc., cause SDDs. In lower technology nodes, interconnect delays are more pronounced. Hence, a small variation in it can manifest as small extra delays. Also, the weak resistive opens or shorts due to airborne particles and chemicals used during the complex manufacturing process cause small delays. These resistive opens or shorts are also prone to aging effects like electromigration and can cause reliability issues. On the other hand, strong resistive opens or shorts cause large delay defects whose effect is visible on all the paths through the defect site. This can be easily detected by traditional fault models. However, SDDs escape traditional testing methods as their effect is visible only when they are propagated through long paths or when the slack is reduced. There are two popular fault models for delay defects: the path delay fault (PDF) model [2], [3] and the transition delay fault (TDF) model [4], [5]. The PDF model assumes that the delay due to defects is distributed along a path. When the cumulative delay of the path exceeds the clock period, then the defect can be detected. This model is more likely to detect small delay defects. But for large industrial circuits, the number of paths can grow exponentially, and the number of patterns also increases. Also, the PDF test set is generated by an extensive search of paths in the design that meet certain sensitization conditions. Hence, it is difficult to use the PDF model. On the contrary, the TDF model assumes that the delay defect is localized at a particular node. Hence, the number of fault sites is equal to the number of nodes in the circuit. But in this model, there are no restrictions on the

The associate editor coordinating the review of this manuscript and approving it for publication was Harikrishnan Ramiah.

path along which the fault must be propagated. Generally, the TDF-based automatic test pattern generation (ATPG) propagates the fault effect through the shortest path possible in concern of runtime. The fault has to be propagated along the longest path to detect the SDD. The timing-aware TDF ATPG tries to propagate the fault effect through the longest path at the expense of pattern count and run time [1], [6]. Even though, if the ATPG is unable to generate the pattern that propagates the fault effect through the longest, it tries to propagate through the second longest path. But in the infield operation, the longest path may be a functional path and the SDD can fail the circuit. The ALAPTF (as late as possible transition fault) model is proposed to find the robust longest path and launch a transition on it [7]. There are many paths through a gate in the circuit. Therefore, the K longest path generation algorithm selects K critical paths through a gate to launch the transition and generate the patterns [8]. The main drawback of these methods is their complex ATPG algorithm. The N-detect TDF ATPG is another alternative used for propagating each fault effect through different paths. This increases the SDD detection probability, but the overall pattern size is quite substantial [9], [10]. Several works in the literature have addressed the problem of this pattern size to generate a compact pattern set [11], [12], [13]. The ability of the patterns to propagate the fault effect through the longest path is graded and a compact pattern set is produced. However, the top-off TDF pattern required for fault coverage increases the count further. In order to improve the pattern grading technique, several factors like cross-talk, process variations, and power supply noises are taken into account to find the ability of a pattern to propagate long paths [14], [15], [16]. This method has increased the accuracy but still pattern count remains high.

Another alternative approach is faster-than-at-speed testing (FAST) [17], [18] where the frequency is tuned according to the user's necessity. This method has more control over detecting the SDDs because increasing the frequency reduces the slack of the fault. The long paths are more prone to process variations [19]. Therefore, it is better to test the shorter paths with the FAST clocks to distinguish whether the delay is due to SDD or process variation. The TDF ATPG patterns have the ability to propagate through shorter paths. So combining them with FAST can be more beneficial. But, the FAST method also poses many challenges. The generation of the faster-than-at-speed clock is addressed in [20]. The proper selection of test frequency and pattern by including the effects of IR drop is illustrated in [17]. Here, the patterns with a close delay distribution are grouped together, and the fast clock for each group is found with reference to the largest pattern delay (along with IR drop effects) of the group. This may also lead to some test escapes, as the last pattern of the group can have more slack margin. In [21], the copies of the TDF pattern set are used and each copy is tested at a particular test clock period (multiples of the system clock period). Here masking is applied to the endpoints that have a delay greater than the test clock period. In this case, the number of patterns is large and the test time is proportional to n times the application of one TDF pattern set. Here n refers to the number of test clock periods.

A TDF pattern set, a predefined set of test clock periods, and the endpoint masking are used to find the optimal pattern and test clock pair to detect the SDD more effectively [22]. Here the major problem is the test clock periods are selected based on the multiples of maximal pattern delay. Nevertheless, each pattern propagates different faults to different endpoints. Also, it does not take the smallest effective delay defect (SEDD) size (discussed in detail in later sections) into consideration. The SEDD size can be included only when the pattern-based fault-endpoint delay is known. Testing a defect less than SEDD size will result in overtesting of the circuit. A method that considers the masking concept, predefined frequencies, and quality metric to generate optimal test options (proper combination of test clock, pattern, endpoint mask) is proposed in [23]. Here, instead of pattern delay information the multiples of the system clock period are used as test clocks. The inappropriate selection of the test clock period may lead to overtesting or loss of SDD coverage. The demerit of testing at predefined frequencies is certain defects that cause small delays can be seen only at a particular frequency. In [24] the faults of a fixed size are simulated, and a hypergraph algorithm is used to find the optimal FAST clock. The graphical processing unit (GPU) based processor is required to observe the waveform of different fault sizes and the effort is high.

The test quality metric is another important factor in quality evaluation. As the FAST-based method is found optimal for SDD detection, the quality metric must also validate the same, i.e., the change in clock frequency must change the metric value. Different delay test quality metrics are discussed in detail in [25]. These metrics are broadly classified into two types: statistical and non-statistical. Small delay defect coverage (SDDC) [26] and statistical delay quality level (SDQL) [27] are the two statistical metrics introduced to validate SDD test quality. These metrics give information about the probability of test escapes of SDDs. However, the major disadvantage is the requirement of delay defect distribution, which may not be available for all technology nodes and processes.

Delay test coverage (DTC) [6], quadratic small delay defect coverage ($SDDC_q$) [26], and weighted slack percentage (WeSPer) [28] are the non-statistical metrics, that do not require any delay defect distribution information. The most common metric among them is DTC which is used in the majority of commercial tools. This metric measures the ability of the pattern to propagate the fault through the longest path. However, it lacks in validating the FAST, as it does not consider the test and system clock information. It only validates the SDD test quality when the test and system clock are equal. Further, this metric assumes that all defect sizes are equiprobably distributed contrary to the actual scenario.

$SDDC_q$ produces different values for different defect sizes but these values are > 100% when the system and test clock periods are different. Hence this cannot validate the FAST method. WeSPer metric is introduced to address all the aforementioned concerns. It includes the information on the test clock, system clock, longest activated path, and test activated path.

Based on the available works, the following conclusion has been made:

- There are several methods suggested in the literature for the selection of test clock periods based on the pattern delay information that lacks the SEDD information on each fault leading to an over or under-test of the circuit.
- The predefined multiples of system clock periods are selected as the test clock periods in the existing research works. They try to find the best test option for each fault that minimizes the slack based on the given test clock inputs. However, the quality of the test options depends completely on the test clock inputs. The existing approaches are arbitrary. Therefore, a proper framework is required to select the best clock periods.
- In WeSPer-based test optimization, redundant test clock periods may occur due to the selection of predefined test clock periods with specific interval. Also, the number of test clock periods is fixed with respect to the allowed margin.

In response to the above interpretations, this paper focuses on selecting the optimal test clock periods. The following are the prime contributions of the paper:

- A test clock period data set is generated for each benchmark circuit depending on the pattern delay and the SEDD information for each fault.
- An optimal test clock speed selector algorithm is proposed to select the most favorable test clock periods. The optimality lies in the fact that the algorithm selects the test clock periods that can maximize the WeSPer%, independent of how different circuits are constrained without a hit-or-miss approach. This is because of its data set dependency.
- The proposed algorithm
  - selects the optimal data point within the user-defined constraint on the maximum allowable number of test clock periods N.
  - avoids redundant test clock inputs to the test optimization algorithm.
  - is flexible to choose a lesser number of clock periods within the allowed constraint.
  - is sensitive to the varying N values.

The sections in this article are organized as follows. For the basic understanding of the concept behind the work, the rudiments of small delay defects and the test quality metric for evaluating them are discussed in section II. The workflow is described in III. The methodology for developing the prerequisites of this workflow is discussed in section IV. The proposed method of optimal test clock period generation and selection is discussed in section V. The optimization algorithm used for test option generation and fault grouping is discussed in sections VI and VII, respectively. The proposed method is applied to the benchmark circuits and validated in section VIII. Finally, section IX concludes the article.

II. RUDIMENTS OF SMALL DELAY DEFECT

The small extra delay due to a manufacturing defect may fail the circuit operation at the functional frequency or cause reliability issues. These delay defects can escape from traditional TDF testing. The generated test patterns may not be able to propagate the fault effect through the longest possible path, leading to SDD test escapes. Fig. 1 illustrates the effect of SDD, assuming that the flip-flops (FF) have zero delays. The X in the figure represents the fault site in the circuit. There are three paths passing through the fault site. The red dotted line represents the longest path and the other two blue dotted lines (path1 and path2) are two other paths through the fault site. Path1 is the second longest path and path2 is the shortest path through the fault site. The fault effect is captured either at the primary output or FF. The graph represents the delay of each path through the fault site. The smallest effective delay defect size that can cause the circuit to fail in at-speed operation is denoted by the blue region in the longest path delay. The SEDD size is recognized with the slack of the fault at the circuit level. Testing the delay defect of size less than SEDD leads to overtesting. If the TDF pattern cannot propagate the fault effect through the longest path, but through path1 or path2, the pattern has to be tested by overclocking the circuit and observing it at T2 and T1, respectively. Fig. 2 represents the special case of small delay due to the defect (hidden delay defect (HDD)). In this case, there is only one possible way to propagate the fault effect. The defect in the location can cause a small extra delay, which is not potential enough to fail the circuit at functional operating speed. However, this delay can grow over time due to the aging effect (electromigration) and can cause failure during infield operation and so require detection. Faster-than-at-speed testing of the path through the fault site can help to detect these SDDs. This work focuses on testing defects that can cause failure during at-speed operation and the following assumptions are made: 1) the exact path delay is known 2) the exact test clock frequency can be generated.

A. WEIGHTED SLACK PERCENTAGE METRIC

The WeSPer metric introduced to evaluate the quality of the SDD testing is given in (1).

$$\text{WeSPer} = \frac{1}{\text{NF}} \sum_{i=1}^{\text{NF}} f_i \times 100\% \tag{1}$$

$$f_i = \frac{T_{\text{sc}} - \text{PD}_{\text{LT}_i}}{T_{\text{tc}} - \text{PD}_{\text{TA}_i}} \text{CL}_i \quad for \quad T_{\text{tc}} > \text{PD}_{\text{TA}_i} \tag{2}$$

here, NF is the number of faults, T_{sc} is the at-speed functional clock period, T_{tc} is the test clock period, PD_{LT_i} and PD_{TA_i} are the longest and test activated paths through the fault site i,

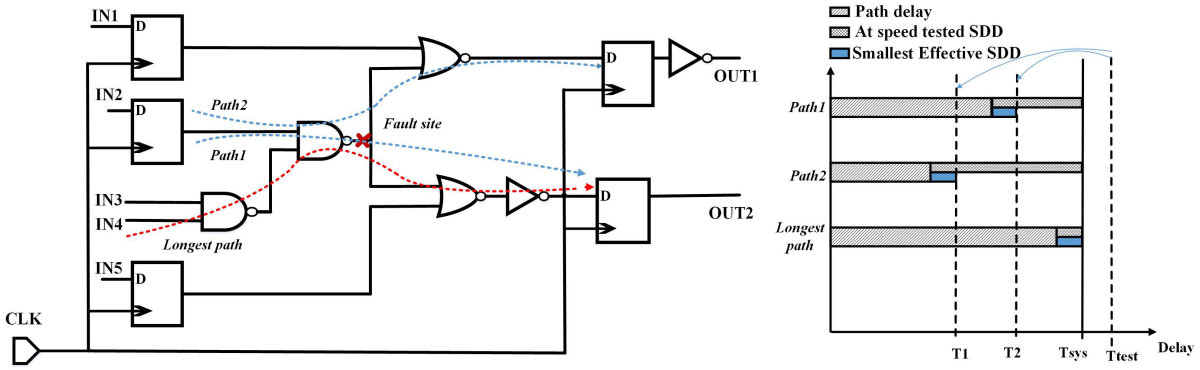


FIGURE 1. Delay of each path of the circuit through fault site X and the observation time T1 and T2 for detecting the small delay defect.

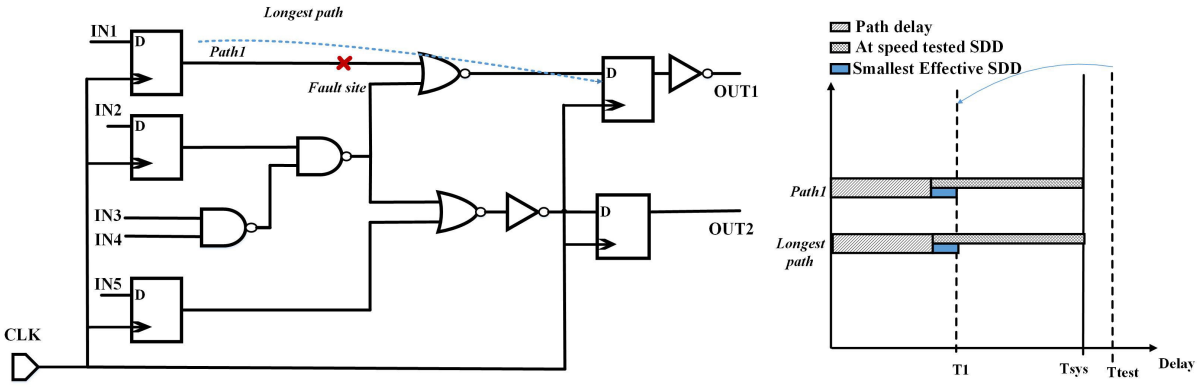


FIGURE 2. Delay of the path through the fault site X and the observation time T1 to detect the hidden delay defect.

respectively. CL_i is the confidence level (CL) factor of the fault which is introduced to include the uncertainty effect of the test. The process variation and overtesting effect can be formulated using this factor. For example, in the case of overtesting, the CL factor is formulated such that, if the chosen test clock is overtesting (testing the SDD that has a size smaller than the SEDD size) the circuit then it will minimize the WeSPer%. In this work, the size of the defect is considered to be equal to or greater than the SEDD. Hence, the CL_i is considered to be 1. This means the test is assumed to be performed in ideal conditions (i.e. no consideration of process variation effect, IR drop, overtesting, etc.). In terms of clock value, the uncertainty effect of the clock network is considered and the parameter is included in the test quality metric calculation. This is because the slack value generated by the tool [29] does not consider the clock uncertainty. Here (2) is modified as (3).

$$f_i = \frac{T_{sc} - \text{uncertainity} - \text{PD}_{\text{LT}_i}}{T_{tc} - \text{uncertainity} - \text{PD}_{\text{TA}_i}} \text{CL}_i \ \textit{for} \ \ T_{tc} > \text{PD}_{\text{TA}_i} \tag{3}$$

III. WORKFLOW

The flow diagram of work in [23] is shown in Fig. 3. There are two phases in the flow 1) the setup phase and 2) the test optimization phase. The setup phase includes TDF pattern generation, delay fault simulation, delay table creation, and test clock period selection. The test optimization phase consists of 2 substages, say, the WeSPer optimizer and fault grouping phases. The delay table and test clock periods selected from the setup stage are taken as input by the test optimization phase and it generates the optimized test option. The WeSPer% depends on the longest path slack and the best-activated path slack of the fault. In this method, the test clock periods are directly selected based on the system clock period of the circuit netlist. The test clock period is chosen as a multiple of the system clock period with an interval of 10 percent. However, the slack of the path depends on the delay through which the fault is propagated and the clock period. Hence, the choice of test clock period is a crucial factor here. The test clock chosen from the method does not always guarantee the improvement of the WeSPer%.

The workflow of the proposed method is shown in Fig. 4. This method aims to select the optimal test clock period in order to maximize the WeSPer%. It introduces the clock selection phase, which selects the optimal test clock period for each benchmark circuit despite the constraints and slack of each path. The clock selection phase consists of two subphases: 1) test clock speed data set generator for each fault and 2) optimal clock speed selector. These two subphases

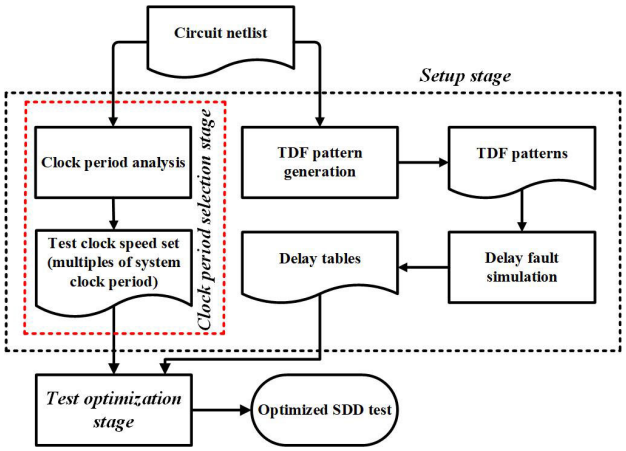


FIGURE 3. Workflow of existing method [23].

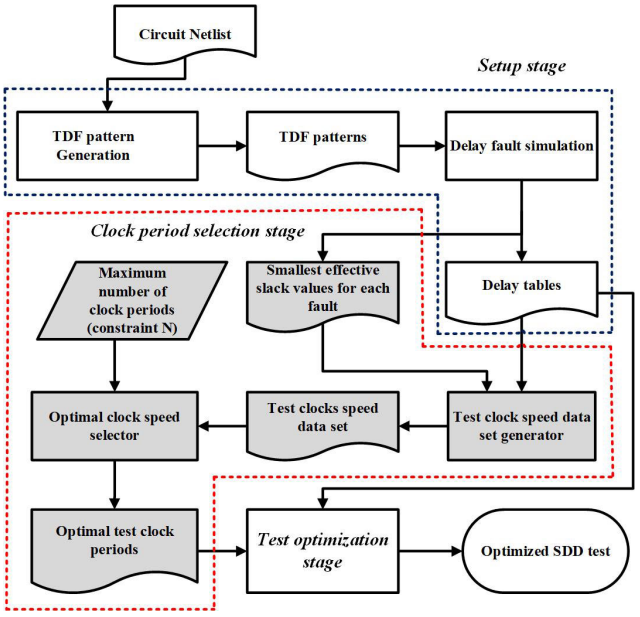


FIGURE 4. workflow of proposed method.

are explained in detail in sections V-A and V-B. The former phase uses the already available delay table and the smallest effective slack information to generate the test clock data set and the latter phase selects the optimal data points (test clock periods) from the data set based on the number of constraints on the test clock periods. The selected data is completely dependent on the circuit characteristics rather than a hit-or-miss technique as in [23]. The selected clock periods including the system clock period are given as input to the test optimization phase to generate the optimized SDD test options.

IV. DELAY TABLE CREATION

The computation of the WeSPer metric needs information on the best-activated path delays through each fault site to each endpoint. The best-activated path delay is the one,

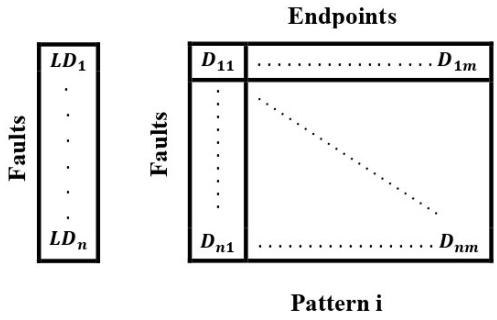


FIGURE 5. Delay table format.

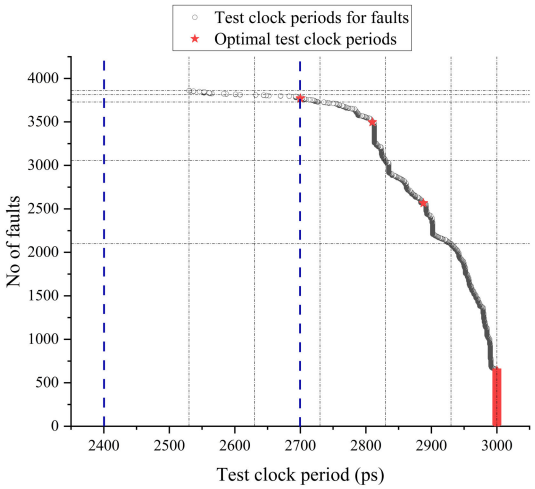


FIGURE 6. Optimal test clock period of each fault for ISCAS89 s5378 benchmark circuit.

that maximizes the WeSPer metric. If the fault effect is not propagated through the longest possible path, testing under a faster-than-at-speed clock will detect the SDD. Hence, the test clock inputs and the activated path delay will decide the WeSPer metric maximization. A commercial tool [29] is used to obtain the delay information of each fault to each endpoint under each pattern. The path delay table structure is given in Fig. 5. The left side table consists of the longest delay (LD_n) through each fault site. The right side table consists of pattern delay to each endpoint 'm'for each fault site 'n'. The procedure for delay table creation is given below.

- Read gate level netlist
- Generate the TDF pattern
- Set the timing engine ON
- Read the SDF file
- Add all the faults
- Read the single pattern format file
- Observe a single endpoint and mask the rest
- Perform the single pattern simulation
- Record the fault delay reports in the delay table
- Repeat the procedure from step 6 for each pattern

V. PROPOSED OPTIMAL TEST CLOCK PERIOD GENERATION AND SELECTION

A. GENERATION OF TEST CLOCK PERIOD SET FOR THE DESIGN

The delay table created for calculating the WeSPer metric is used to find the optimal clock frequency set. Each pattern propagates the fault effect to different endpoints through different paths. The maximum delay of the path through which the fault is propagated by the pattern is found and recorded in a set. Here, the delay information of the TDF pattern through each fault to each endpoint alone is known. Hence, there is no direct information on the paths propagated by the pattern. The algorithm 1 depicts the method for generating the test clock set. For each fault ($\phi_i \in \phi$), the algorithm finds the SEDD size with the system clock and the longest path delay through the fault site. The algorithm searches for each pattern ($P_i \in P$) that propagates the fault effect ϕ_i and records the test-activated path (PD_{TAi}) delay. The SEDD size is added to the (PD_{TAi}) and the values are recorded in τ_c. This is performed to find the set of test clocks that do not overtest the circuit. The maximum value of τ_c is chosen to avoid more overclocking and is stored separately. This is because the overclocking comes with the expenses of IR drop, increased number of masking vectors, etc. This method is repeated for all the faults. Finally, the optimal test clock period is obtained for all testable faults. Fig. 6 illustrates the effective test clock period obtained by the above method for the ISCAS89 benchmark circuit s5378.

Algorithm 1 Generating the Test Clock Period Set

```
τ ← system_clock_speed
m_a ← ∅
for each φ_i ∈ φ do
    SEDD ← T_sc − PD_LTi
    τ_c ← ∅
    for each P_i ∈ P do
        Q_i ← PD_TAi ∈ ℝ > 0
        T_c ← Q_i + SEDD
        τ_c ← τ_c ∪ T_c
    end for
    if τ_c ≠ ∅ then
        m_a ← m_a ∪ max(τ_c)
    end if
end for
```

B. SELECTION OF OPTIMAL TEST CLOCK PERIOD

The test clock speed sets are generated as discussed in section V-A. But for each fault, one test clock period is generated. All these values cannot be used for FAST in concern of the test time and run time for generating the test options. Hence, the user-defined value N (maximum number of test clock periods that can be used) is set. The number of clock periods selected should be less than or equal to N. In this work, N is considered to be 4 to compare with the work in [23]. The impact of the change in N values with respect to the WeSPer% is also performed for further analysis.

Algorithm 2 illustrates the way to select the optimal test clock periods as an input for WeSPer-based optimal test option generation (discussed in detail in the next section). The test clock set (m_a) generated from algorithm 1 must be grouped based on a range. The range is defined with the increment of 100 ps ($\delta = 0.1$) from the minimal value of the test clock set. This δ value is chosen because the pattern-based delay is in the range of a few 100's of picoseconds (observed for all benchmark circuits) and also to maintain homogeneity for comparison purposes. This value may differ based on the system clock period, design constraints, pattern delay ranges, and different technology nodes. The test clock period (m_{a_i}) $\in m_a$ is recorded into different sets based on the range they belong to. Each set is then analyzed for the data point ($f_i \in F$) that is nearest to all other data points in set F. The data point that has minimal distance (average) is added to the set (Optimal_test_clock in the algorithm). In certain cases, the number of data in the Optimal_test_clock set is more than the user-defined value, say N. In such a case, the number of data points in each range is found and the optimal test clock period for the ranges with a higher number of data is chosen.

The proposed method can be illustrated better with Fig. 6 for N = 4. In the figure, the minimum clock period is 2530 ps and the maximum is 3000 ps (system clock period). The predefined test clock periods are selected as multiples of the system clock period with a 10% interval (say 0.9T, 0.8T, 0.7T). In the figure, 0.8T (2400 ps) itself is below the limit of the optimal test clock period set values. Hence, the test clock period (0.7T) becomes redundant in the WeSPer optimization phase. Here, the utilization of the test clock periods also decreases (i.e. out of 4 different test clock periods only 3 will be seen in the final test option). The other value will be checked for generating the best test option, but it becomes redundant. The proposed test clock selection method chooses 4 different test clock periods for the s5378 circuit. The black dotted vertical lines show the ranges chosen. The range from 2930 ps to 3000 ps is not taken into consideration, as the system clock period of 3000 ps must be included as one of the test clock periods. There are 4 more ranges but only 3 clocks are required. The horizontal dotted lines show the number of faults that exist in the particular range. As the test clock period value decreases the number of faults also decreases. The group in the range from 2530 ps to 2630 ps has minimal faults in it and hence it is ignored. In the proposed method, the selected test clock period interval varies from 4 to 10% from the nominal test clock period, whereas in the predefined method, the interval varies from 10 to 30%. In the ISCAS benchmark circuit s5378, one at-speed clock and three other values (red star) are chosen. These values are given as inputs to the WeSPer-based test option generation phase, which is discussed in the next section.

The maximum pattern delay observed is 1003 ps. If this value or its multiples are considered for the test clock periods,

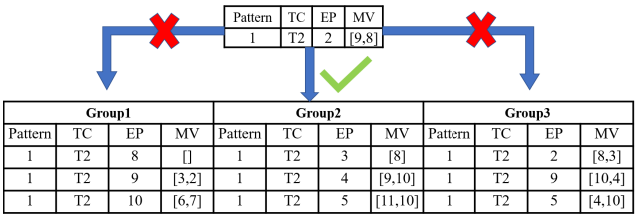
the optimizer algorithm will keep these redundant. This is because it is already observed that the 0.7T (2100ps) test clock period itself is redundant as the algorithm doesn't allow overtesting. Hence, the direct pattern delay values as considered in work [22] cannot be used in this work. Therefore the proposed method, which incorporates the SEDD size for test clock set generation, is found to be reliable for all designs constrained in different ways to perform FAST without overtesting.

Algorithm 2 Selecting Optimal Test Clock Period

```
min_val ← min(m_a), max_val ← max(m_a)
range ← ∅, Optimal_test_clock ← ∅
range ← range ∪ min_val
δ ← 0
value ← min_val
repeat
  δ ← δ + 0.1
  value ← value + δ
  range ← range ∪ value
until value <= max_value
for i = 1 to length(range) do
  F_i ← ∅
  for each m_ai ∈ m_a do
    if (m_ai >= range_i ∈ range) and (m_ai < range_i+1 ∈
    range) then
      F ← F ∪ m_ai
    end if
  end for
  Average ← ∅
  for each f_i ∈ F do
    data ← f_i
    sum ← 0, count ← 0
    for each f_i ∈ F do
      diff ← max(data, f_i) − min(data, f_i)
      sum ← sum + diff
      count ← count+1
    end for
    average ← sum/count
    Average ← Average ∪ average
  end for
  minimum_option_value ← min(Average)
  Optimal_test_clock ← Optimal_test_clock∪ (data
  when average = minimum_option_value)
end for
if length(Optimal_test_clock) >= userdefined then
  select the test clock values for which the number of data
  points between a particular range is higher
end if
```

VI. WESPER-BASED OPTIMAL TEST OPTION GENERATION

The algorithm in [23] is used to generate the optimal test options. The optimal test option for a particular fault includes the information on the test pattern, test clock, endpoint, and masking vector. The selection of the best combination of test options to detect the particular fault is the target of the algorithm. When the test clock period is less than the at-speed system clock period, there may be paths with delays exceeding the test clock period. The endpoints corresponding to these negative slack paths should be masked and the masking vector is generated for the pattern. The target of this work is to test only the small delay defects that fail the chip during at-speed operation. Hence, overtesting information is not taken into consideration.



Group1				Group2				Group3			
Pattern	TC	EP	MV	Pattern	TC	EP	MV	Pattern	TC	EP	MV
1	T2	8	[]	1	T2	3	[8]	1	T2	2	[8,3]
1	T2	9	[3,2]	1	T2	4	[9,10]	1	T2	9	[10,4]
1	T2	10	[6,7]	1	T2	5	[11,10]	1	T2	5	[4,10]

FIGURE 7. Example of fault grouping.

VII. FAULT GROUPING

The number of test options generated is proportional to the number of faults. The fault grouping is required to group all the faults that can be tested under similar conditions. The test options that have similar information i.e. pattern, test clock, and endpoint vector are grouped together into one set. The single set (group) is equivalent to the application of a single pattern of traditional TDF tests. In this work, only a single test option is generated for each fault rather than many test options. As the proposed method focuses on optimal clock period selection, validation with respect to one test option is explored. The number of pattern sets (groups) is the total number of patterns to be applied. Grouping is performed for the test option that has the same pattern and test clock. Fig. 7 represents an example (for understanding) of how the pattern set (group) looks like. The test options are represented with pattern number (Pattern), test clock period (TC), endpoint (EP), and Maskingvector (MV). There are three different groups for pattern number 1 and test clock 2 (TC2). Now, one more test option has to be grouped into the available groups. The new test option should be checked for compatibility with other groups. The compatibility will be checked with respect to the endpoint and masking vector. The $ep_i \in$ EP should not belong to $mv_{(i,j)} \in$ MV. In the example given, the observation endpoint is 2 and the masking vector contains endpoints 8 and 9. In group 1, endpoints 8, and 9 are the observation points and hence cannot be grouped. In group 3, endpoint 9 is the observation point and cannot be grouped. But, group 2 is a compatible group as endpoint 2 is not in masking vectors and 8, and 9 are not in the observation endpoint. In a similar way fault grouping is performed. If no group is compatible then a new group is formed.

TABLE 1. Benchmark characteristics.

Circuit_name	No. of fault sites	No. of flipflops
b08	734	21
b09	782	28
b11	1880	30
b13	1506	45
s298	572	14
s444	824	21
s1494	2428	6
s5378	6406	162
s9234	4820	132

TABLE 2. Transition delay fault (LOC) based ATPG characteristics.

Circuit_name	No. of Patterns	FC%	DTC%
b08	46	68.53	66.03
b09	39	85.81	81.81
b11	98	82.13	74.35
b13	59	74.5	71.19
s298	30	77.97	72.19
s444	35	78.28	70.17
s1494	151	84.02	76.01
s5378	231	83.33	75.93
s9234	189	88.15	77.94

VIII. RESULTS AND DISCUSSION

The proposed method is applied to the ISCAS89 and ITC99 benchmark circuits. The circuits are synthesized using TSMC 65 nm standard cell library using a commercial tool [30]. Table 1 summarizes the benchmark characteristics. The circuit name, number of fault sites, and the number of flip-flops are specified in columns 1, 2, and 3, respectively. Table 2 summarizes the Launch-off-capture (LOC) [5] TDF ATPG characteristics for the benchmark circuits. The number of patterns, the fault coverage (FC)%, and the delay test coverage (DTC)% are given in columns 2, 3, and 4 respectively.

Fig. 8 depicts the relative percentage increase of DTC and pattern count of timing-aware ATPG with respect to traditional TDF ATPG. Here, a maximum of 61.5% increase in pattern count for a 0.68% increase in DTC is observed. A maximum of 3.02% increase in DTC is observed for a 23.8% increase in pattern count. However, the same trend is not observed for all benchmark circuits. There is still not even a 1% increase in DTC for most of the benchmark circuits.

Table 3 shows the WeSPer% for at-speed (column 2), predefined FAST chosen at different intervals say 15%, 10%, 9%, 8%, and 7% (column 3-7), and the proposed FAST (column 8) for N = 4. The work in [23] used a 10% margin for choosing the test clock period. The results in the table show that the change in the choice of predefined test clock period intervals impacts the WeSPer%. However, the variation observed is very minimal in many cases with respect

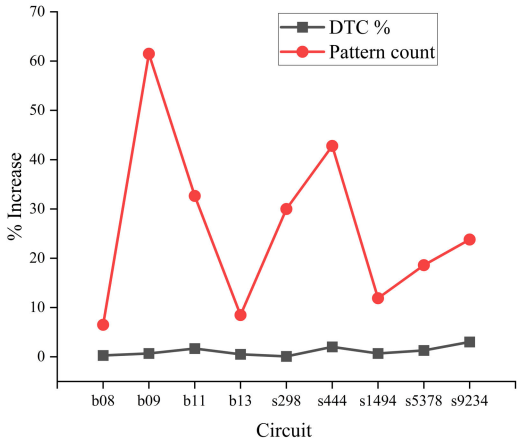


FIGURE 8. Relative percentage increase of DTC% and pattern count of timing aware patterns with respect to traditional TDF patterns.

to the different intervals of the test clock periods. These results proclaim that the optimal selection of the best test clock period is imperative to extract the maximum advantage of the test optimization phase and it should be independent of the design constraints and slack of the paths. It is observed that there is an average relative increase of 1.267% and 1.74% for a predefined FAST with respect to at-speed-based test option generation for 10% and 7% intervals, respectively. However, the proposed method shows an average relative increase of 3.213% with respect to at-speed-based test option generation. Hence it can be inferred that the selected clock periods are optimal within the constraint N.

Table 4 shows the WeSPer% for various N values of predefined FAST (column 2-5) with 10% intervals, proposed FAST (column 6-9), and No. of clock inputs that are selected by the proposed algorithm (columns 10-13). It can be clearly observed that for the predefined FAST, there is no change in the WeSPer% with respect to N. This is because the interval of 10% generates test clock periods that are beyond the limit of the range particular to each circuit. But, for the proposed method the WeSPer% varies with N. Still, the observation infers that beyond N = 4 only b11, s1494, s5378, and s9234 show an increase in quality. However, the variation is very minimal i.e. a maximum of 0.4% is observed for the b11 circuit and an average relative percentage increase of 0.07% is observed. This is because of the data set grouping of the proposed method i.e. δ value. Beyond 4 data set groups, it is observed that only a few or no data are available, hence it has a minimum contribution on the quality maximization. Also, the delay distribution of this particular technology contributes to this data grouping. The algorithm is also flexible to choose different data groups based on the technology used. The No. of clock periods selected infers that within the given constraint, the algorithm has the flexibility to select the minimal number of clock periods to achieve the maximum increase in WeSPer%. For example in the case of the b13

TABLE 3. Comparison of WeSPer% for different intervals of predefined test clock periods and the proposed FAST.

Circuit Name	at-speed WeSPer%	Predefined FAST WeSPer%					Proposed FAST WeSPer%
		15%	10% [23]	9%	8%	7%	
b08	97.78	97.78	97.96	98	98.27	98.23	99.213
b09	97.65	97.65	97.698	97.698	98.34	98.52	99.25
b11	94.096	95.81	97.51	97.45	97.49	97.57	98.582
b13	98.036	98.036	98.036	98.036	98.036	98.036	99.305
s298	96.337	96.337	96.337	96.8	96.85	97.1	98.422
s444	95.426	95.426	95.426	95.426	95.45	95.6	98.467
s1494	96.079	96.289	97.07	97.37	97.64	97.77	98.94
s5378	93.6601	94.94	96.08	96.44	96.65	97.31	98.18
s9234	92.88	95.19	97.12	97.38	97.58	97.5	98.764

TABLE 4. Comparison of WeSPer% for predefined (10% interval) and the proposed FAST for various N.

Circuit name	Predefined FAST WeSPer%				Proposed FAST WeSPer%				No. of clock period inputs (Proposed FAST)			
	N = 3	N = 4	N =5	N = 6	N = 3	N = 4	N = 5	N = 6	N = 3	N = 4	N = 5	N = 6
b08	97.96	97.96	97.96	97.96	99.13	99.213	99.213	99.213	3	4	4	4
b09	97.698	97.698	97.698	97.698	99.15	99.25	99.25	99.25	3	4	4	4
b11	97.51	97.51	97.51	97.51	97.68	98.582	98.75	98.89	3	4	5	6
b13	98.036	98.036	98.036	98.036	99.305	99.305	99.305	99.305	2	2	2	2
s298	96.337	96.337	96.337	96.337	98.422	98.422	98.422	98.422	3	3	3	3
s444	95.426	95.426	95.426	95.426	98.467	98.467	98.467	98.467	3	3	3	3
s1494	97.07	97.07	97.07	97.07	98.53	98.94	99.018	99.018	3	4	5	5
s5378	96.08	96.08	96.08	96.08	97.79	98.18	98.38	98.38	3	4	5	5
s9234	97.12	97.12	97.12	97.12	97.36	98.764	98.94	99.04	3	4	5	6

TABLE 5. Pattern group counts for predefined and proposed FAST methods.

Circuit name	Predefined FAST	Proposed FAST			
		N = 3	N = 4	N = 5	N = 6
b08	49	74	74	74	74
b09	40	53	56	56	56
b11	188	214	225	265	270
b13	59	81	81	81	81
s298	30	39	39	39	39
s444	35	74	74	74	74
s1494	215	358	405	411	411
s5378	427	562	658	762	762
s9234	377	461	545	596	637

circuit, with 2 test clock periods, the proposed method can observe a 99.305 WeSPer% irrespective of the constraint N. But, in the case of predefined FAST, only a fixed number (N) of test clocks are given as input for test optimization.

Table 5 represents the pattern group count generated by the method specified in section VII for predefined FAST and proposed FAST for various N values. The predefined FAST WeSPer% does not change with N and hence the pattern count. For the proposed FAST the change in N impact the pattern count values. By comparing table 4 and 5, it can be observed that beyond N = 4 for a small increase in quality, there is a high increase in pattern. Also for N = 3 itself, there is a noticeable average relative increase of 2.76 in WeSPer% w.r.t to at-speed test. When N = 4 there is only a 0.5 percent average relative increase in WeSPer% with the expense of 6.57% increase in pattern count. Beyond that i.e. for N > 4, there is a 13.36% increase in pattern count to have a relative WeSPer% increase of 0.01 with respect to N = 3. Hence, further analysis is performed with respect to N = 4 in order to compare with [23].

Fig. 9 shows the utilization % of the test clock period for predefined FAST and the proposed FAST methods. In the former method, the utilization percentage is less than the proposed method. The comparison is made for N = 4 (including at-speed clock) i.e. the number of test clock periods given as input to the WeSPer optimization algorithm. The number of test clock periods used by the optimization algorithm out of the total inputs given defines the utilization %. For predefined FAST, four test clock periods (T, 0.9T, 0.8T, 0.7T) are given as input and the utilization percentage is calculated accordingly. But, in the case of the proposed method, the number of clock periods is ≤N, and calculations are done accordingly. For the proposed method 100% utilization is observed. The same trend is also observed for varying N values.

Table 5 shows that the pattern group count of the predefined method is less than the proposed one. This is because the proposed method uses better test clock periods, hence the different masking vectors generated will also increase. However, the direct comparison of the two methods is not a better way, as the utilization percentage is different

TABLE 6. Overall test quality metric for all faults for N=4.

Circuit	DTC% [6]	WeSPer_atspeed% [23]	WeSPer% (Predefined) [23]	% increase	WeSPer%(Proposed)	% increase
b08	66.03	67.61	67.69	0.118	68.203	0.87
b09	81.81	84.59	84.622	0.037	85.422	0.983
b11	74.35	78.161	80.455	2.935	81.175	3.856
b13	71.19	73.584	73.584	0	74.1772	0.806
s298	72.19	75.993	75.993	0	77.119	1.481
s444	70.17	75.861	75.861	0	77.467	2.117
s1494	76.01	80.83	81.644	1.007	83.162	2.885
s5378	75.93	79.5	80.969	1.85	82.232	3.436
s9234	77.94	83.536	85.849	2.76	87.249	4.45
			Average	0.967	**Average**	2.32

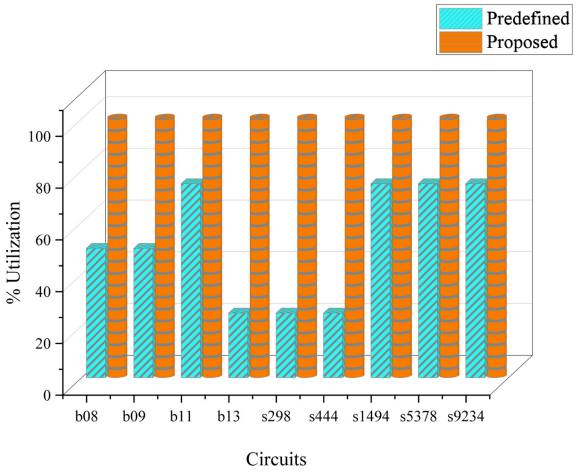


FIGURE 9. Utilization percentage comparison of predefined vs. proposed FAST method.

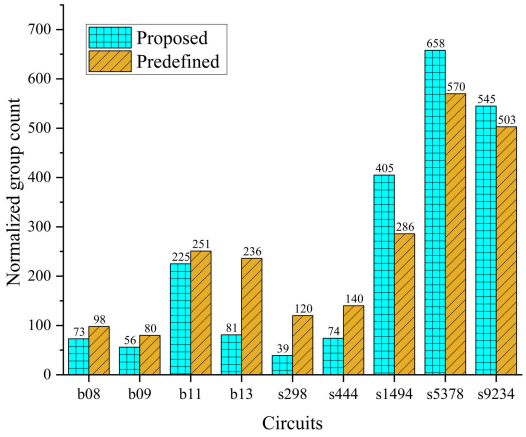


FIGURE 10. Comparison of normalized group count for the predefined and proposed method of FAST clock selection.

for both cases. In order to compare the pattern group, the predefined FAST method is normalized with respect to 100 percent utilization as shown in Fig. 10 for N = 4. The pattern group count is higher in the proposed method for three benchmark circuits s1494, s5378, and s9234. This is expected because, as the number of faults increases, the different combinations of masking vectors will be more, which in turn increases the group count. This pattern count can further be reduced by generating various test options for single faults as proposed in [23]. Still, the pattern count is less than three times the application of the TDF pattern set.

Fig. 11 shows the comparison of the relative increase in WeSPer% of timing aware (at-speed), predefined FAST, and proposed FAST-based optimal test option generation with respect to the at-speed test. It can be inferred that the proposed method has shown a higher relative percentage increase compared to other methods. A maximum of 6.335% increase is observed for the benchmark circuit s9234 and a minimum of 1.15% is observed for the b13 circuit. In the case of a predefined FAST-based WeSPer percentage, a maximum of 3.95% is observed for the benchmark circuit s9234 and a minimum of 0.004% for the b09 circuit. The proposed FAST has an average relative increase of 1.84% in WeSPer compared to the predefined FAST. In the case of three circuits

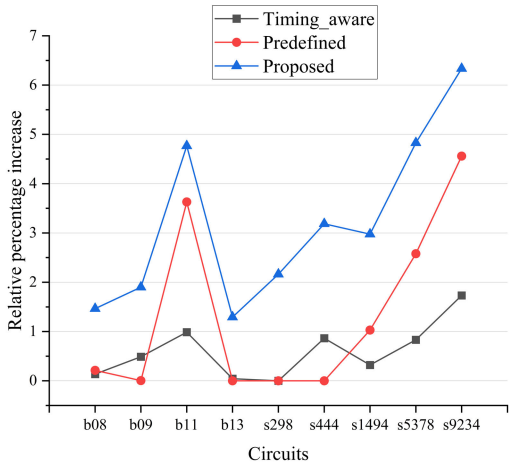


FIGURE 11. Relative percentage increase of timing-aware (at-speed), predefined and proposed FAST with respect to at-speed WeSPer%.

say b09, b13, s444 the relative increase in timing aware (at-speed) WeSPer% is higher than the predefined FAST method. The use of FAST is to detect the small delay defects that escape the timing-aware TDF tests. When a predefined multiple of test clock periods (with a 10 percent interval) is used for FAST, the actual use of the method is not harnessed properly. It is also important to avoid overtesting and false

rejection. The proposed method selects the best test clock periods and increases the SDD detection without overtesting the circuit.

Table 6 summarizes the overall test quality metric percentage for the total number of faults. The DTC%, WeSPer% for at-speed, predefined FAST, and proposed FAST are shown in columns 2,3,4, and 6, respectively for N=4. The relative percentage increase of WeSPer for predefined FAST (10% interval) and proposed FAST with respect to at-speed WeSPer is depicted in columns 5 and 7, respectively. From the reported values it is inferred that there is an average relative increase of 0.967% for the predefined method and 2.32% increase for the proposed method. When compared to the predefined method, the proposed method has an average relative increase of 1.339% in WeSPer.

IX. CONCLUSION AND FUTURE DIRECTION

The proposed method has increased the WeSPer to a maximum of 6.335% and an average relative increase of 3.213% with respect to at-speed. As compared to the predefined FAST it has an overall average relative increase of 1.339%. With the increase in WeSPer%, it can be inferred that the algorithm selects the optimal test clock data as compared to the predefined method within the constraint N. The data set grouping has also contributed to an increase in WeSPer% with a minimal number of test clock periods i.e. lesser than N for certain benchmark circuits. Also, the proposed method has 100 percent utilization in terms of test clock period inputs. Moreover, it has also reduced the arbitrary nature of most of the previous existing works. The results have also proved that the algorithm is sensitive to varying N values. As the delay distribution of different technologies varies, the sensitivity of N contributes to selecting optimal test clock periods irrespective of how the design is constrained by altering the data set grouping. The proposed method has conferred to relative increase of 2.476% in WeSPer as compared to timing-aware ATPG. Nevertheless, the increase in pattern count caused by the masking vector is a factor that must be addressed. The increase in pattern count is proportional to N. Despite the fact that the earlier work has addressed this problem, selecting the optimal pattern set, i.e. a combination of traditional TDF patterns with top-off patterns such as timing-aware, path delay fault, and N-detect ATPG-based patterns, can be advantageous. These patterns when combined with the FAST can help to reduce the pattern count. This can be a future research perspective to further optimize the test options generation and detection of SDD.

ACKNOWLEDGMENT

The authors acknowledge Siemens EDA India for providing EDA tool support to execute this work.

REFERENCES

[1] N. Ahmed, M. Tehranipoor, and V. Jayaram, "Timing-based delay test for screening small delay defects," in *Proc. 43rd Annu. Conf. Design Autom. (DAC)*, 2006, pp. 320–325.

[2] I. Pomeranz, "A metric for identifying detectable path delay faults," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 11, pp. 1734–1742, Nov. 2012.

[3] A. K. Majhi and V. D. Agrawal, "Delay fault models and coverage," in *Proc. 11th Int. Conf. VLSI Design*, Jan. 1998, pp. 364–369.

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

[5] J. Savir and S. Patil, "On broad-side delay test," in *Proc. IEEE VLSI Test Symp.*, Dec. 1994, pp. 284–290.

[6] X. Lin, "Timing-aware ATPG for high quality at-speed testing of small delay defects," in *Proc. 15th Asian Test Symp.*, Nov. 2006, pp. 139–146.

[7] P. Gupta and M. S. Hsiao, "ALAPTF: A new transition fault model and the ATPG algorithm," in *Proc. Int. Conf. Test*, Oct. 2004, pp. 1053–1060.

[8] W. Qiu, J. Wang, D. Walker, D. Reddy, X. Lu, Z. Li, W. Shi, and H. Balachandran, "K longest paths per gate (KLPG) test generation for scan-based sequential circuits," in *Proc. Int. Conf. Test*, Oct. 2004, pp. 223–231.

[9] I. Pomeranz and S. M. Reddy, "On n-detection test sets and variable n-detection test sets for transition faults," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 3, pp. 372–383, Mar. 2000.

[10] M. E. Amyeen, S. Venkataraman, A. Ojha, and S. Lee, "Evaluation of the quality of N-detect scan ATPG patterns on a processor," in *Proc. Int. Conf. Test*, Oct. 2004, pp. 669–678.

[11] K. Peng, J. Thibodeau, M. Yilmaz, K. Chakrabarty, and M. Tehranipoor, "A novel hybrid method for SDD pattern grading and selection," in *Proc. 28th VLSI Test Symp. (VTS)*, Apr. 2010, pp. 45–50.

[12] M. Yilmaz, K. Chakrabarty, and M. Tehranipoor, "Test-pattern selection for screening small-delay defects in very-deep submicrometer integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 5, pp. 760–773, May 2010.

[13] C.-Y. Chang, K.-Y. Liao, S.-C. Hsu, J. C. Li, and J.-C. Rau, "Compact test pattern selection for small delay defect," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 6, pp. 971–975, Jun. 2013.

[14] K. Peng, M. Yilmaz, K. Chakrabarty, and M. Tehranipoor, "Crosstalk- and process variations-aware high-quality tests for small-delay defects," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 6, pp. 1129–1142, Jun. 2013.

[15] K. Peng, M. Yilmaz, M. Tehranipoor, and K. Chakrabarty, "High-quality pattern selection for screening small-delay defects considering process variations and crosstalk," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2010, pp. 1426–1431.

[16] K. Peng, M. Yilmaz, K. Chakrabarty, and M. Tehranipoor, "A noise-aware hybrid method for SDD pattern grading and selection," in *Proc. 19th IEEE Asian Test Symp.*, Dec. 2010, pp. 331–336.

[17] N. Ahmed and M. Tehranipoor, "A novel Faster-than-at-speed transition-delay test method considering IR-drop effects," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 10, pp. 1573–1582, Oct. 2009.

[18] S. Chakravarty, "Silicon evaluation of faster than at-speed transition delay tests," in *Proc. IEEE 30th VLSI Test Symp. (VTS)*, Apr. 2012, pp. 80–85.

[19] R. Tayade, S. Sundereswaran, and J. Abraham, "Small-delay defect detection in the presence of process variations," in *Proc. 8th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2007, pp. 711–716.

[20] S. Pei, H. Li, S. Jin, J. Liu, and X. Li, "An on-chip frequency programmable test clock generation and application method for small delay defect detection," *Integration*, vol. 49, pp. 87–97, Mar. 2015.

[21] B. Kruseman, A. K. Majhi, G. Gronthoud, and S. Eichenberger, "On hazard-free patterns for fine-delay fault testing," in *Proc. Int. Conf. Test*, Oct. 2004, pp. 213–222.

[22] T. Yoneda, K. Hori, M. Inoue, and H. Fujiwara, "Faster-than-at-speed test for increased test quality and in-field reliability," in *Proc. IEEE Int. Test Conf.*, Sep. 2011, pp. 1–9.

[23] O. A. Hasib, Y. Savaria, and C. Thibeault, "Optimization of small-delay defects test quality by clock speed selection and proper masking based on the weighted slack percentage," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 3, pp. 764–776, Mar. 2020.

[24] M. Kampmann, M. A. Kochte, C. Liu, E. Schneider, S. Hellebrand, and H.-J. Wunderlich, "Built-in test for hidden delay faults," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 10, pp. 1956–1968, Oct. 2019.

[25] P. Muthukrishnan and S. Sathasivam, "A technical survey on delay defects in nanoscale digital VLSI circuits," *Appl. Sci.*, vol. 12, no. 18, p. 9103, Sep. 2022.

[26] N. Devta-Prasanna, S. K. Goel, A. Gunda, M. Ward, and P. Krishnamurthy, "Accurate measurement of small delay defect coverage of test patterns," in *Proc. Int. Test Conf.*, Nov. 2009, pp. 1–10.
[27] Y. Sato, S. Hamada, T. Maeda, A. Takatori, Y. Nozuyama, and S. Kajihara, "Invisible delay quality–SDQM model lights up what could not be seen," in *Proc. IEEE Int. Conf. Test*, Nov. 2005, p. 9.
[28] O. A. Hasib, Y. Savaria, and C. Thibeault, "WeSPer: A flexible small delay defect quality metric," in *Proc. IEEE 34th VLSI Test Symp. (VTS)*, Apr. 2016, pp. 1–6.
[29] (2023). *Tessent Scan and ATPG User's Manual*. Accessed: Jan. 15, 2023. [Online]. Available: https://docs.sw.siemens.com/en-US/doc/202305012.atpg_gd?audience=external
[30] (2023). *Design Compiler Manual Synopsys*. [Online]. Available: https://spdocs.synopsys.com/dow_retrieve/latest/dg/dcolh/dcug/preface/about_this_user_guide.html?sc0=x

PRATHIBA MUTHUKRISHNAN received the B.E. degree in electronics and communication engineering and the M.E. degree in applied electronics from the College of Engineering Guindy (CEG), Anna University, Chennai, India, in 2018 and 2020, respectively. She has qualified for UGC-NET, in 2019 and 2020. She received the Junior Research Fellowship by UGC, India, in December 2020. She is in the third year of research with the School of Electronics Engineering, Vellore Institute of Technology, Vellore, India, where she is also a Senior Research Fellow. Her research interests include DFT and hardware security.

SIVANANTHAM SATHASIVAM (Senior Member, IEEE) received the B.E. degree in electronics and communication engineering from the University of Madras, India, in 1997, the M.Tech. degree in VLSI design from SASTRA University, India, in 2002, and the Ph.D. degree in electrical engineering from the Vellore Institute of Technology (VIT), India, in 2014. He is currently a Professor and the Dean with the School of Electronics Engineering, VIT. He was a Leader of the VLSI and Embedded System Division, VIT, from 2007 to 2009. He has published more than 80 research articles in refereed international journals and conferences. His areas of research interests include the design for testability, reconfigurable architectures, and low-power VLSI design. He is a member of IEICE, VLSI Society of India (VSI), and the Indian Society for Technical Education (ISTE).