FF-Control Point Insertion (FF-CPI) to Overcome the Degradation of Fault Detection under Multi-Cycle Test for POST

Hanan T. Al-AWADHI†, Tomoki AONO†, Nonmembers, Senling WANG†a), Yoshinobu HIGAMI†, Members, Hiroshi TAKAHASHI†, Senior Member, Hiroyuki IWATA††, Nonmember, Yoichi MAEDA††, Member, and Jun MATSUSHIMA††, Nonmember

SUMMARY Multi-cycle Test looks promising a way to reduce the test application time of POST (Power-on Self-Test) for achieving a targeted high fault coverage specified by ISO26262 for testing automotive devices. In this paper, we first analyze the mechanism of Stuck-at Fault Detection Degradation problem in multi-cycle test. Based on the result of our analysis we propose a novel solution named FF-Control Point Insertion technique (FF-CPI) to achieve the reduction of scan-in patterns by multi-cycle test. The FF-CPI technique modifies the captured values of scan Flip-Flops (FFs) during capture operation by directly reversing the value of partial FFs causing improving the Design for Testability (DFT) to meet one constraint usually would aggravate the others [2]. For shorting the TAT of POST, many means focused on improving the test architecture involving the scan structure design or test scheduling, such as the scan chain partitioning [3], scan-shift clock reusing [4], TMS (Tri-Modal Scan) test [5] and capture-per-cycle hybrid-TPI [6]. However, these means still suffer from the problems in terms of the large hardware overhead, complex ATPG applications, and huge elapsed times for the simulation (logic & fault). Multi-cycle test that applies more than one capture clock to the circuit is proposed to test volume reduction by allowing multiple tests at each test pattern (scan-in pattern) [8]–[11]. In multi-cycle test, for each test pattern (scan-in pattern generated by an on-chip Random Pattern Generator such as a Liner Feedback Shift Register-LFSR), the test response captured at each capture cycle will be reused as test stimuli at the next capture cycles. The ability of the multi-cycle test is to reduce the number of the scan-in patterns for testing, because one root scan-in pattern generates M capture patterns as the test stimuli under the M cycles test. It thus provides more chances of stuck-at fault detection compared to the traditional scan test with single capture clock, and thus can reduce the number of scan-in patterns. In addition, it is also known that multi-cycle test has a behavior to take the CUT (Circuit Under Test) closer to its functional operation conditions that can generate functional vectors with smaller power consumption which are very helpful to low power at-speed testing for delay faults detection [12]. Multi-cycle test is thus one of the promising ways to achieve a good trade-off among the fault coverage, TAT and power consumption for POST. However, we raise two major issues that obstruct the effect of multi-cycle test to reduce the scan-in patterns for shorting the TAT of POST, which are Fault effects vanishing problem [13]–[16] and Fault Detection Degradation of capture patterns.

The first one is the Fault effects vanishing problem starting any functional operations, the time allowed for test application is very short, e.g. TAT < about 50msec.

3) The low power consumption

The consideration of power consumption during test is helpful to avoid false test (good devices fail the test) and yield loss under the delay fault model [7]. It is a difficult task to make a balanced relationship among the fault coverage, TAT and power consumption because improving the Design for Testability (DFT) to meet one constraint usually would aggravate the others [2]. For shorting the TAT of POST, many means focused on improving the test architecture involving the scan structure design or test scheduling, such as the scan chain partitioning [3], scan-shift clock reusing [4], TMS (Tri-Modal Scan) test [5] and capture-per-cycle hybrid-TPI [6]. However, these means still suffer from the problems in terms of the large hardware overhead, complex ATPG applications, and huge elapsed times for the simulation (logic & fault). Multi-cycle test that applies more than one capture clock to the circuit is proposed to test volume reduction by allowing multiple tests at each test pattern (scan-in pattern) [8]–[11]. In multi-cycle test, for each test pattern (scan-in pattern generated by an on-chip Random Pattern Generator such as a Liner Feedback Shift Register-LFSR), the test response captured at each capture cycle will be reused as test stimuli at the next capture cycles. The ability of the multi-cycle test is to reduce the number of the scan-in patterns for testing, because one root scan-in pattern generates M capture patterns as the test stimuli under the M cycles test. It thus provides more chances of stuck-at fault detection compared to the traditional scan test with single capture clock, and thus can reduce the number of scan-in patterns. In addition, it is also known that multi-cycle test has a behavior to take the CUT (Circuit Under Test) closer to its functional operation conditions that can generate functional vectors with smaller power consumption which are very helpful to low power at-speed testing for delay faults detection [12]. Multi-cycle test is thus one of the promising ways to achieve a good trade-off among the fault coverage, TAT and power consumption for POST. However, we raise two major issues that obstruct the effect of multi-cycle test to reduce the scan-in patterns for shorting the TAT of POST, which are Fault effects vanishing problem [13]–[16] and Fault Detection Degradation of capture patterns.

The first one is the Fault effects vanishing problem
that denotes the fault effects excited at some intermediate capture cycles might disappear before their effects are propagated to the final capture cycle for observation due to the expanded long propagation path that would cause fault coverage loss [13]–[16]. To address the fault effects vanishing problem, a sequential observation (SEQ-OB) technique that directly observes the values of small part of Flip-Flops (FFs) at each capture cycle has been introduced in [10] and [13].

In [14], we have proposed a method for selecting the FFs for SEQ-OB which can assist the fault coverage improvement by analyzing the circuit-structure w/o using simulation. In [15] and [16] we have proposed a DFT technique named Fault-Detection-Strengthened (FDS) method for strengthening the fault detection capability of multi-cycle test to enhance the effect of SEQ-OB technique. Also, we have developed the underlying technologies including the FDS flip-flop (FF) design and an original in-house tool named FVP-TPI (Fault-effects-Vanishing Point-TPI) to compute the most effective insertion point of FDS FFs for SEQ-OB based on the FF selection algorithm proposed in [14].

Another problem obstructs the effect of multi-cycle test is the Fault Detection Degradation (FDD) of capture patterns. In this paper, we define the FDD as the decrease of capability of capture pattern to detect more additional stuck-at faults. It is well known that multi-cycle test has the behavior to take the CUT closer to its functional operation conditions. As reported in [17], the internal state transitions of CUT will decrease and become stable at very low level as increasing the number of capture cycles that is helpful to lower power at-speed testing for delay faults (e.g.: transition delay faults) detection [12], [17], [18]. On the other hand, the state of many FFs during multiple capture cycles (responses of CUT after each capture) might consequently become constant values (e.g.: fixed at 0/1) when large number of capture cycles are applied. Since the value of FFs (capture pattern) are reused as test stimuli at the subsequent capture cycles, large number of FFs with constant values would cause the loss of randomness property of the capture patterns. We believe that this behavior obstructs to detect more additional stuck-at faults by using the capture patterns.

In [14], we have proposed a novel method named FF-Control Point Insertion (FF-CPI) technique to overcome the FDD problem of capture patterns by modifying the captured values of scan Flip-Flops (FFs) during capture operation. The method inserts FF-Control Points (FF-CP) between the output of some scan FFs and the combinational circuit to enhance the randomness of the capture patterns for detecting more additional faults by reversing the value of partial FFs or loading random vectors to the capture patterns. Also, we have proposed the methods to evaluate the FFs for determining the candidate FFs for FF-CPI that can achieve more stuck-at fault detection, by analyzing the circuit structure w/o any simulations for the purpose of shortening the development period of DFT. The Experimental results of benchmark circuits show that the proposed method can further reduce the number of scan-in patterns (at most 28.57X pattern compression) for achieving the specified target fault coverage compared to the SEQ-OB method (at most 12.5X), which is helpful to further shorten the TAT of POST.

This paper is a complement to our previous work presented in [19]. In this paper, we first overview the FF-CPI technique and the evaluation methods of FFs for FF-CPI proposed in [19]. In addition, we introduce a Multi-Criteria Decision Analysis method named TOPSIS [20] to comprehensively evaluate the metrics derived from the evaluation methods of FFs presented in [19] for ranking the FFs for FF-CPI. Moreover, since in [19], all FFs were observed by SEQ-OB technique to avoid the impact of fault effects vanishing on fault detection, which causes too large hardware overhead to a practical use. In this paper, we introduce a partial observation technique in which the limited number of FFs are used for the observation by SEQ-OB into the FF-CPI technique to achieve a good trade-off between the scan-in pattern reduction and the hardware overhead for practical use.

The main contributions of this paper are as follows.

1) A DFT technique referred to FF-CPI is proposed to address the fault detection degradation of the capture patterns under the multi-cycle test.
2) Three kinds of FF selection methods for FF-control point insertion are proposed by analyzing the circuit structure w/o any simulation that can shorten the period of DFT.
3) The Partial observation of SEQ-OB is introduced into the FF-CPI technique for achieving a good trade-off between scan-in pattern reduction and hardware overhead for practical use.
4) Experimental results of ISCAS’89 and ITC’99 benchmark circuits under the single stuck-at fault model show a significant scan-in pattern reduction with smaller hardware overhead.

The paper is organized as follows. Section 2 overviews the multi-cycle test technique and point out its problems. Section 3 introduces the FF-CPI technique to address the fault detection degradation problem. Section 4 proposes the FF selection methods for FF-Control circuit insertion. Section 5 shows the experimental results of ISCAS’89 and ITC’99 benchmark. Section 6 concludes the paper.

2. Preliminaries

In this section, we describe the basic concept of multi-cycle test and its fault detection degradation problem.

2.1 Multi-Cycle Testing

Figure 1.a shows the clock design of broad-side for a traditional scan test for stuck-at fault testing. In one test session composed of scan-shift operation and capture operation, a pseudo-random pattern (scan-in pattern) generated by LFSR is serially shifted into the scan chain and the previous test
response is shifted out from the scan chain during scan operation (Scan Enable: SE = 1), then, in capture operation (SE = 0), the scan-in pattern is applied to the CUT in parallel and the corresponding responses of the CUT are captured into the FFs at the capture clock. The traditional scan test requires large number of scan-shift clock to serially load the scan-in pattern (test response) into (out) the scan chain which depends on the length of the longest scan chain, and only one capture clock is applied for stuck-at fault detection. Suppose the length of the longest scan chain is \( L \), each test session requires \( L + 1 \) clock cycles, and only one clock (capture clock) is used for testing. To achieve a target fault coverage, LBIST usually requires large number of pseudo-random patterns (scan-in patterns) that cause great amount of test application time.

In contrast to the traditional scan test, the multi-cycle test applies more than one capture clocks during the capture operation in each test session (see Fig. 1.b). In the first capture cycle of the test session, the test response of the root scan-in pattern is captured into the scan chain, and then it will be applied to the CUT in parallel as new test stimuli of the subsequent capture operation. Thus, for \( M \)-cycle test with the length of \( L \) scan chain, each test session requires \( L + M \) clock cycles out of which \( M \) clocks are used for fault testing. In other words, \( M \) tests are available in one test session with one root scan-in pattern. It should be noted that \( L \) is general much larger than \( M \) in a large-scale circuit (e.g.: \( L = 500, M = 10 \)). Compared to the traditional scan test, the multi-cycle test could provide more chances to detect additional faults in the expanded capture cycles which are possibly missed by the root scan-in pattern. It has promising potential to reduce the number of scan-in patterns for achieving a target fault coverage (e.g.: 90% for ASIL D). The fewer the scan-in patterns can cause the less the shift operation (\( L \) clocks). As a result, it can shorten the TAT of POST in turn.

2.2 Problems of Multi-Cycle Testing for Scan-In Pattern Reduction

There are two major problems obstruct the effect of multi-cycle test to reduce the scan-in patterns. 1) fault effects vanishing; 2) fault detection degradation of capture patterns. In this subsection, we explain the angle of the problems by using the examples.

**Fault effects vanishing problem** denotes that the fault effects excited at an intermediate capture cycle might disappear before it is propagated to the last capture cycle due to the expanded long propagation path under time-frame expansion combinational circuit. To address the fault effects vanishing problem, in [14]–[16], we have proposed the DFT techniques refers to SEQ-OB with FDS-FF design to directly observe and keep the faulty values propagated to FFs at multiple capture clocks so as to gather the faulty values before vanishing. Figure 2 shows the DFT for SEQ-OB and the design of FDS_FF. In Fig. 2, we replace the normal scan flip-flop (FF2) with a FDS_FF which is constructed by inserting an XOR gate at the capture input of a normal scan-FF. In the capture operation, the faulty value propagated to the input of a FDS_FF will be XORed with the value of its adjacent FF, and the output value of the XOR gate will be stored in the scan-FF. Consequently, all the faulty values that are propagated to the position where a FDS_FF is inserted can be observed, and the fault effects vanishing issue in the multi-cycle test can be mitigated.

**Fault detection degradation problem** denotes the capability of capture patterns to detect more additional stuck-at faults degrades as a large number of capture cycles are applied. It is known that multi-cycle test has a behavior to take the CUT closer to its functional operation conditions with less internal transitions of CUT. This behavior is helpful to low power at-speed testing for detecting the delay faults [12], [17]. However, as the internal transitions of CUT decrease, it would consequently cause the state of many FFs become constant (e.g.: fixed at 0 or 1) at the following capture cycles. In other words, many bits of capture patterns would never change at the subsequent capture cycles and the sequence of capture patterns would lose the randomness property. We believe that this behavior obstructs the capture patterns to detect more additional stuck-at faults. For illustration, we performed a preliminary experiment on a small circuit to explore the mechanism of FDD problem as follows.

2.3 Analysis of FDD Problem

We defined the \( TpCi,j \) (Transitions per Cycle, 0 < i <= total number of FFs, 0 < j <= total number of capture cycles) to evaluate how many times the state of a FF (i) changed at the jth capture cycle during a complete test. For a complete
test with \( k \) scan-in patterns, the \( TP_{C,i,j} \) can be calculated by following formula.

\[
TP_{C,i,j} = \sum_{k=1}^{\text{Number of Test}} v_{i,j-1,k} \oplus v_{i,j,k}
\]

Where, \( v_{i,j,k} \) denotes the value of \( FF_i \) at the \( j \)th capture cycle when the \( k \)th scan-in pattern is applied.

Figure 3 shows an example to calculate the \( TP_C \) for \( FF_1 \) under 4 capture cycles test with 3 scan-in patterns. In the example, \( v_{1,0,k} \) denotes the initial state of \( FF_1 \) of the \( k \)th test which comes from the root scan-in pattern, \( v_{1,j,k} \) denotes the state of \( FF_1 \) at the \( j \)th capture cycle in the \( k \)th test session which are the responses of \( CUT \) after the \( j \)th capture cycle. For each test, we compare the state of \( FF_1 \) at each capture cycle with its state at the previous cycle to detect a transition of \( FF_1 \) at the cycles. When \( k (k = 3) \) test sessions are applied, we take the sum of transition at the same capture cycle as the \( TP_C \) of \( FF_1 \). The value of \( TP_C \) in the example represents that the state of \( FF_1 \) changed in three test sessions at the first capture cycle, two test sessions at the second capture cycle, and one test session at the third and the fourth capture cycle during the complete test.

To illustrate the mechanism of FDD problem, the preliminary experiments is performed on a small benchmark circuit s298 to evaluate 1) the amount of FFs that would be held at constant (e.g.: fixed at 0 or 1) as increase the capture cycles. In other words, many bits of capture patterns would never change at the subsequent capture cycles and the sequence of capture patterns would lose the randomness property. We believe that this behavior would obstruct the capture patterns to detect the additional stuck-at faults which are possibly missed by their root scan-in patterns, as follows.

1) **Evaluation of \( TP_C \) of FFs at each capture cycle**

First, we execute a logic simulation with 50 scan-in patterns under 10 capture cycles test to calculate the \( TP_C \) for each FF. Figure 4 shows the \( TP_C \) of each FF at 10 capture cycles. The vertical axis denotes the total \( TP_C \) in 50 test sessions (with 50 scan-in patterns), the horizontal axis denotes the capture cycle number, and the lines denote the \( TP_C \) of each FF at the corresponding capture cycle. It can be observed that the number of test sessions in which the state transition occurred in each FF decreases as applying more capture cycles. For \( FF_4, FF_6, FF_7, FF_8, FF_9, \) and \( FF_12 \), state transitions occur at the first capture cycle in many test sessions (16–27), then their states become constant value in most test sessions after the second capture cycle.

The observation in Fig. 4 confirmed that the multi-cycle test can cause the state of many FFs become constant (e.g.: fixed at 0 or 1) to increase the capture cycles. In other words, many bits of capture patterns would never change at the subsequent capture cycles and the sequence of capture patterns would lose the randomness property. We believe that this behavior would obstruct the capture patterns to detect the additional stuck-at faults which are possibly missed by their root scan-in patterns, as follows.

2) **Evaluation the additional fault detection at each capture cycle**

The fault-dropping simulation (one-detection drop) with 10 capture cycles using 50 scan-in patterns is executed on s298 to evaluate the number of additional stuck-at faults detected at each capture cycle. In the fault simulation, we observe the value of all FFs at each capture cycle by SEQ-OB technique, and check the fault detection at each capture cycle. Once a fault is detected at a certain capture cycle, it will be accounted inclusive of the additional faults of the capture cycle, and be dropped (eliminated) from the fault list. All faults (308 stuck-at faults) of the combinational circuit are simulated in this experiment.

Figure 5 shows the result, where the line denotes the average \( TP_C \) of all FFs at each capture cycle given in Fig. 4 which is referred to the right vertical axis, and the columns denote the total number of the additional stuck-at faults detected at each capture cycle referred to the left vertical axis. In where, the column of cycle 1 shows the result of the scan-in patterns (50 scan-in patterns), and the column of cycle 2 shows the result achieved by the capture responses of the scan-in patterns, and so on forth, the columns of cycle 3–10 shows the results achieved by the responses of the capture patterns in cycle 2–9, respectively. It should be noted that the additional faults of a capture cycle denote the faults that are newly detected at the current capture cycle (by scan-in
pattern or capture pattern) which are also detectable by other scan-in patterns or capture patterns in the following test. Compared to the results of cycle 1 where 154 stuck-at faults are detected by 50 scan-in patterns in total, it can be observed that the number of additional faults detected by the capture patterns (2–10 cycle) decrease as increasing the capture cycles, and the number of state transitions of FFs show a decreasing trend similar with the number of additional faults.

From the above observation, we insist that the state of many FFs will be held at constant (e.g.: fixed at 0 or 1) in the subsequent capture cycles that would cause the sequence of capture patterns at each test session would lose the randomness property. Therefore, it makes the capture patterns difficult to detect more additional stuck-at faults. On the other hand, since multi-cycle test enables many tests to be executed in the capture operation after a scan-in pattern is serially shifted in the scan chain. The operation for a scan-in pattern is very time consuming. If the more additional faults can be detected by capture patterns, the fewer the scan-in patterns to achieve a target fault coverage. Therefore, the multi-cycle test can cause the less the shift operation to shorten the TAT. To improve the effect of multi-cycle test to reduce the scan-in patterns for shorting the TAT of POST, it is necessary to develop a DFT technique to address the FDD problem.

3. FF-Control Point Insertion for Multi-Cycle Test

In this section, we introduce the basic idea to overcome the Fault Detection Degradation problem to enhance the stuck-at fault detection and scan-in pattern reduction for multi-cycle test [19].

As discussed above, the FDD problem of multi-cycle test arises from the loss of the randomness property of capture patterns because of the state of many FFs would be held at constant (e.g.: fixed at 0 or 1) as increasing the number of capture cycles. Therefore, a key to address the FDD problem should be improving the randomness property for capture patterns. An idea is modifying the value of some FFs as to avoid constant states during the capture cycles. For illustration, we performed an additional experiment on s298 circuit as follows.

Example of our analysis

To verify the effect of the idea of modifying the value of FFs in capture cycles, we choose a certain FF of s298 circuit as the target and reverse its value in force during the capture cycle test. We check that 1) how much impact modifying the value of a FF would have on the TpC and 2) whether more additional faults could be detected by the capture patterns.

1) The impact on TpC when modifying the value of FF

As we show the result of TpC given in Fig. 4, the state transition of FF6 occurred in 16 tests at the first capture cycle, then its state will be held at constant value at the following capture cycles in almost all tests. It is because that FF6 is one of the 2-bit mode register of the status logic in the traffic light controller which is directly controlled by a primary input (PI). In multiple cycle test, the values of PIs are fixed during capture cycles, FF6 is therefore easy to be held at a constant value. In here, we chose FF6 as a target, and perform the experiments with logic-simulation as well as that of Fig. 4 by reversing the captured value of FF6 during capture operation from the third capture cycle. Figure 6 shows the TpC of each FF at 10 capture cycles by reversing capture value of FF6.

From the above observation, we insist that the state of many FFs will be held at constant (e.g.: fixed at 0 or 1) in the following test. Compared to the result of cycle 1 where 154 stuck-at faults are detected by 50 scan-in patterns in total, it can be observed that the number of additional faults detected by the capture patterns (2–10 cycle) decrease as increasing the capture cycles, and the number of state transitions of FFs show a decreasing trend similar with the number of additional faults.

In Fig. 7, we try to show the impact on the additional stuck-at fault detection of capture patterns as reversing the capture value of FF6. The dotted columns show the total faults. The dotted columns show the total
number of additional stuck-at faults detected at the corresponding capture cycle and the dotted line shows the average $T_{PC}$ of all FFs at each capture cycle, when reversing the capture value of FF6, respectively. The blue columns and blue lines are the original results given in Fig. 5 for comparison.

The results of the dotted line show that reversing the value of FF6 during capture cycles causes significant state transitions on the FFs. Consequently, the total number of additional stuck-at faults (the dotted columns) detected at the capture cycles (cycle 3, 5, 8 and 9) start from the third capture are increased. However, the total number of additional faults detected at the first and the second capture cycle show decreases compared to the common multi-cycle test (non-reversing in FF6). The reason is due to the improvement of the stuck-at fault detection capability of capture patterns by reversing the value of FF6.

To make an understanding of the basic idea, Table 1 gives the number of additional faults detected at each capture cycle with/without reversing the value of FF6 when the first three scan-in patterns are applied. In a common multi-cycle test (non-reversing in FF6), when the first scan-in pattern is applied to the 10-cycle test, 61 faults are detected in total and of which 48 and 13 faults are additionally detected at the first (by the scan-in pattern SP1) and the second cycle (by the responses of the SP1), respectively. In the other capture cycles, there are not any additional faults that can be detected. On the other hand, when reversing the value of FF6 in force, 80 faults are detected in total and of which 17 and 2 faults are detected in addition at the third and fourth capture cycle, respectively. Applying the second scan-in pattern w/o reversing FF6 will detect 13 faults (12 at cycle 1 by SP2, 1 at cycle 2) in addition, however, the additional faults detected at the first cycle by SP2 decreased (from 12 to 11) compared to that of w/o reversing FF6. Same results can be observed in the third scan-in pattern SP3, where the number of additional faults detected at the first capture cycle (by SP3) decreased (from 46 to 33) when reversing the value of FF6, and the number of additional faults from the third capture cycle increased significantly. It can be explained by reversing the value of FF6 makes the capture patterns available to detect the faults in advance which would be detected by the following additional scan-in patterns.

This observation provides that the modification of the value of some FFs during the capture operation is available to improve the additional faults detection of capture patterns, and to reduce the number of scan-in patterns for achieving a target fault coverage. As the results shown in Table 1, to achieve 25% fault coverage, the common multi-cycle test (w/o modifying the value of FF6) requires three scan-in patterns, while only two scan-in patterns would be enough by modifying the value of FF6. In the experiment of Fig. 7, the multi-cycle test w/o reversing the value of FF6 requires 38 scan-in patterns to reach 90% stuck-at fault coverage, which is reduced to 26 scan-in patterns just by reversing the value of FF6.

### Table 1

| Capture cycle | Scan-in Pattern | Reverse FF | Accumulated | Faults | Faults | Faults |
|---------------|----------------|------------|-------------|--------|--------|--------|
| 1             | NO             | 48         | 61          | 0      | 48     | 0      |
| 2             | NO             | 15         | 51          | 0      | 33     | 0      |
| 3             | NO             | 33         | 33          | 0      | 1      | 0      |

![Fig. 8](image-url)

**Fig. 8** FF-Reversing for capture pattern control [19]

### FF-Control Point Insertion technique

Based on our analysis, we propose FF-Control Point Insertion technique (FF-CPI) to enhance the randomness property for capture patterns by inserting FF-Control Points (FF-CP) between the output of certain scan FFs and the combinational circuit to modify the captured values of FFs before they are applied to the following capture cycle. In [19], we proposed two kinds of FF-CP circuit named FF-Reversing and Random-Load to modify the value of FFs during capture operation which are described as follows.

**FF-Reversing CPI**

One FF-CP circuit is to reverse the captured value of the FFs per cycle by inserting a value reversion (bit-flipping) circuit at the scan-FF, we call it the FF-Reversing.

Figure 8 shows the basic design concept of FF-Reversing control. In the capture mode, the present-state ($T_i$: the pattern applied at the current capture cycle) and the next-state ($R_i$: capture response at the current capture cycle) of FF are checked whether there is a transition occurs at the current capture cycle or not. If not, the FF-Reversing control circuit will apply the inverted value of $R_i$ to the CUT as $T_{i+1}$, otherwise apply the current capture response $R_i$ to the CUT for the next capture cycle. An external control signal “CAP_LOAD” is used to enable the FF-Reversing when it is set to 1, otherwise, the normal capture operation will be performed. This method can reverse the value of control FF at capture cycles automatically and easy to control, however, it only modifies the capture patterns with inverted value that cannot directly improve the randomness for capture patterns which might have less effect to improve the additional stuck-at faults detection.

**Random-Load CPI**

Another method to modify the value of FFs during capture operation is to directly load the pseudo-random vectors to the capture patterns, we call it the Random-Load.

Figure 9 shows the structure of the method (details are
masked due to the patent concern). A Random-Load Circuit is inserted between the output of FF and the combinational logics, and a “CAP_LOAD” signal controls the circuit to select either the value of the FF or the pseudo-random vectors will be applied to the combinational logics. The pseudo-random vectors can be fed by either the stored data in a memory or generated by an on-chip test pattern generator (TPG). This method can directly improve the randomness of capture patterns, however, would cause large area overhead and complicate the timing design of capture operation.

4. FFs Selection for FF-Control Point Insertion

FF-CPI modifies the value of some FFs to avoid constant states during the capture cycles. One problem is that which FFs would contribute mostly to improve the fault detection of capture pattern when modifying their values, and how to evaluate the FFs for FF-CPI. A key is to evaluate how much modifying the value of FFs would have on the internal states of CUT. When the large number of FFs become constant during capture cycles, the state of the gates and signal lines would also be fixed at each capture cycle that is not helpful to excite (propagate) new faults (faulty effects). We consider that modifying the value for the FFs which could generate more state variations (state changes) of CUT at different capture cycles should provide more chances to excite (propagate) new faults (faulty effects) and contribute to detecting more additional faults. It should be noted that the TrP metric described in Sect. 2 is not suitable to evaluate the FFs for FF-CPI. A key is to evaluate how much modifying the value of FFs would have on the internal states of CUT. When the large number of FFs become constant during capture cycles, the state of the gates and signal lines would also be fixed at each capture cycle that is not helpful to excite (propagate) new faults (faulty effects). We consider that modifying the value for the FFs which could generate more state variations (state changes) of CUT at different capture cycles should provide more chances to excite (propagate) new faults (faulty effects) and contribute to detecting more additional faults.

In [19], we have proposed two methods to evaluate the FFs for FF-CPI, they will be explained in this section as follows. Main difference between this paper and [19], we introduce a Multi-Criteria Decision Analysis method to comprehensively evaluate the metrics derived by these methods to make an optimal rank of FFs for FF-CPI.

4.1 Method 1: Transition Probability Increment (TrPI)

The main idea is to select the FFs as the candidates for FF-CPI which could generate more state variations of CUT at different capture cycles when modify their values. In order to evaluate the impact of FFs on state variations of CUT when modifying their values during capture cycles, in this study, we utilize the Probabilistic Testability Measures means named COP to calculate the state transition probability of CUT for each FF under multi-cycle test. The FF-CPI modifies the value of some FFs (i.e., FF-Reversing or Random-Load) is inserted. When suppose a FF-CP is inserted into a candidate FF, the 0/1 controllability (C0 or C1) of PI (primary input) and PPI (pseudo-primary input: FF at the first capture) to 0.5/0.5, then, calculate the value of C0 and C1 for each gate at each time-frame. The transition probability (TrP) of a gate $g_{ij}$ (i < number of gates, j < M cycles) can be calculated by the Eq. (1).

$$\text{TrP}(g_{ij}) = C_0 \times C_{1,i+j} + C_1 \times C_{0,i+j}$$  \hspace{1cm} (1)

The average transition probability of all gates during N-cycle test (AVE_{TrP}) can be calculated by the Eq. (2).

$$\text{AVE}_{\text{TrP}} = \frac{1}{M} \sum \text{TrP}(g_{ij})$$  \hspace{1cm} (2)

where, $1 \leq i \leq \text{number of gates}$, $0 \leq j \leq M$.

The Transition Probability Increment (TrPI) is defined as the difference of the average transition probability of all gates before and after a FF-CP (FF-Reversing or Random-Load) is inserted. When suppose a FF-CP is inserted into a candidate FF, the 0/1 controllability (C0/C1) of the output signal line of the FF are set to 0.5/0.5 in force. The procedure to evaluate the transition increment induced by value modification of each FF is shown as follows.

**Procedure:**

**Step 1.** Calculate the original average transition probability of all gates during M-cycles test, which is denoted by AVE_{TrP}_{org}.

**Step 2.** For each FF $n$ (n = 1 to N, N: the number of FFs), set the value of C0/C1 at all capture cycles to 0.5/0.5, and calculate the average transition probability of all gates during M-cycles denoted by AVE_{TrP}_{FFn}.

**Step 3.** Calculate the difference between AVE_{TrP}_{org} and AVE_{TrP}_{FFn}, which is denoted by TrPI_{FFn}.

**Step 4.** Rank the FFs by the value of TrPI of each FF.

The FF which has large TrPI will be selected for FF-CPI.
4.2 Method 2: Logic Impact Area of FFs (LIMA)

Main idea is to select the FF which has a large output cone (the arrival logic region from the output of FF). We believe that modifying the value of such FFs could cause more state variations at different capture cycles and provide more chances to excite more faults. We define the Logic Impact Area (LIMA) of FF as a metric to evaluate FFs for FF-CPI. LIMA can be calculated by the following five parameters.

For each FF \( n = 1 \) to \( N \), \( N \) : the number of FFs):

- **P1.** The depth of the logic output cone of FF \( n \) which refers to the length of the longest structural path from FF \( n \) to the input of any reachable flip-flops of FF \( n \) (including itself).
- **P2.** The width of the logic output cone of FF \( n \) which denotes the largest number of gates at the same logic level.
- **P3.** The number of branches existing in the logic output cone of FF \( n \).
- **P4.** The distance between FF \( n \) and the POs which refers to the length of the longest structural path from FF \( n \) to POs.
- **P5.** The total number of logic gates existing in the logic output cone of FF \( n \).

Figure 11 shows an example to calculate the value of P1~P5. For each FF, we derive the value of P1~P5 by forward tracing process started from the FF, and calculate the sum of the value of P1~P5 as LIMA for FF selection. We believe that modifying the value of the FFs which have large LIMA should be effective to detect more additional faults.

4.3 Hybrid Evaluation Metric (HEM) by TOPSIS

The effect of FFs selected for FF-CPI individually by the above evaluation methods of TrPI and LIMA highly depends on the circuit structure. In this paper, we propose the Hybrid Evaluation Metric (HEM) to calculate the ranking of the FFs for FF-CPI under the general circuits. The HEM introduces a Multi-Criteria Decision Analysis method named TOPSIS [20] to comprehensively evaluate the metrics derived by TrPI and LIMA with user-specified weights. The procedure of HEM is model as follows.

**Procedure:**

Main procedure is to rank the FFs for FF-CPI by comprehensively evaluating multiple metrics with different features using TOPSIS.

**Step 1.** Create an evaluation matrix consisting of \( U \) alternatives (# of FFs) and \( O \) criteria (# of evaluation metrics of TrPI and P1~P5 of LIMA) with the intersection of each alternative and criteria given as \( t_{ij} \), we therefore have a matrix:

\[
T = (t_{ij})_{N \times O}, \quad (i = 1, 2, \cdots, U, \ j = 1, 2, \cdots, O) \tag{3}
\]

**Step 2.** Normalize matrix \( T \) by following formula.

\[
R = (r_{ij})_{N \times O}, \quad r_{ij} = t_{ij} / \sqrt{\sum_{i=1}^{U} t_{ij}^2} \tag{4}
\]

**Step 3.** Calculate the weighted normalized decision matrix \( v_{ij} \) by (5).

\[
v_{ij} = w_{j} r_{ij}, \quad \sum_{j=1}^{O} w_{j} = 1 \tag{5}
\]

**Step 4.** Determine the worst alternative \( v_{ij}^{-} \): minimum value of each criteria) and the best alternative \( v_{ij}^{+} \): maximum value of each criteria), and calculate the distance between the target alternative \( i \) and the worst condition \( S_{i}^{-} \) and the distance between the alternative \( i \) and the best condition \( S_{i}^{+} \) by formula (6).

\[
S_{i}^{-} = \sqrt{\sum_{j=1}^{O} (v_{ij} - v_{ij}^{-})^2}, \quad S_{i}^{+} = \sqrt{\sum_{j=1}^{O} (v_{ij} - v_{ij}^{+})^2} \tag{6}
\]

**Step 5.** Calculate the similarity to the worst condition \( C_{i} \) for each alternative by (7):

\[
C_{i} = \frac{S_{i}^{-}}{S_{i}^{+} + S_{i}^{-}} \tag{7}
\]

**Step 6.** Rank the FFs by \( C_{i} \) for FF-CPI.

In HEM method, the weight of the metrics derived by TrPI and LIMA can be specified by user depending on circuits. In this paper, we set the equal weight for the metrics of TrPI and LIMA, which is 0.5.

5. Evaluation Experiments

5.1 Experimental Setup

We conducted the experiments on ISCAS89 and ITC99 benchmark circuits to evaluate the effect of FF-CPI technique. A 16-bits internal type LFSR (characteristic polynomial: \( X^{16} + X^{15} + X^{13} + X^{4} + 1 \)) w/o Phase Shifter is used to generate pseudo-random patterns. A parallel scan structure with 100 FF-length of scan-chains is adopted (when # of FFs > 1600, 200 FF-length) into the circuits. A multi-cycle BIST with 10 capture clocks is implemented into the circuits for stuck-at faults testing. In the experiments, we select
10% of the FFs of each circuit for capture pattern control (FF-Reversing Control and Random-Load Control) by the FF selection methods (TrPI, LIMA, and HEM) proposed in Sect. 4, respectively.

The major purpose of our study is to reduce the number of scan-in patterns for shortening the TAT of POST in compliance with the LF metric requirement of ISO26262 (stuck-at fault coverage \( \geq 90\% \)). Therefore, we perform the experiments by increasing the number of scan-in patterns generated by LFSR and recorded the accumulated fault coverage in 50 scan-in patterns increments until 90% fault coverage is achieved or one million scan-in patterns are generated.

SEQ-OB technique proposed in [10] that directly observes the capture values of FFs during the multi-cycle test is conducted to handle the fault effects vanishing problem. In [19], in order to highlight the effect of FF-CPI technique, all FFs of the circuit were used for SEQ-OB to avoid the impact of fault effects vanishing on fault detection. However, observing all FFs will cause too large hardware overhead to practical use. In this paper, we conduct the partial observation technique presented in our previous papers [14]–[16] that only 20% of FFs of each circuit are selected for SEQ-OB to reduce the hardware overhead. The FFs used for SEQ-OB are selected by the FF selection algorithm presented in [14].

For comparison, the traditional scan test with single capture clock (SCAN), 10 capture clocks test w/o SEQ-OB (MUL) and with SEQ-OB by observing all FFs (FULL) and partial observation (P-OB20: 20% of FFs are observed) are performed, respectively.

5.2 Scan-In Pattern Reduction by FF-CPI

To evaluate the scan-in pattern reduction by FF-CPI technique, in the following experiments we control the capture patterns by FF-CPI started from the third capture cycle in order to remain the effect of fault detection achieved by scan-in patterns and the second capture cycle.

Figure 12 shows the curve of fault coverage for s13207. For achieving 90% stuck-at fault coverage, the traditional scan test with single capture clock (SCAN) requires 20,600 scan-in patterns. When multi-cycle test with 10 capture cycles (MUL) is applied, the fault coverage becomes worse and the number of scan-in patterns to achieve 90% fault coverage tends to increase (one million scan-in patterns just get 85.56% fault coverage) due to the fault effects vanishing problem discussed in Sect. 2.

SEQ-OB technique is effective to handle the fault effects vanishing problem. As the results shown in Fig. 12, SEQ-OB by observing all FFs (FULL) and partial observation (P-OB20) significantly improved the fault coverage for multi-cycle test, that only 11,600 scan-in patterns are needed to achieve 90% fault coverage. We indicate that partial observation can get almost the same pattern reduction as well as FULL observation (observing all FFs) which is helpful to reduce the hardware overhead for practical use. In order to further reduce the scan-in patterns, FF-CPI technique with FF-Reversing Control and Random-Load Control is conducted on the circuit. 10% of FFs are selected by the FF selection methods (TrPI, LIMA, and HEM) proposed in Sect. 3. For easy to read, we just show the fault coverage results of the FF selection method using LIMA metric in Fig. 12 (detailed results of scan-in pattern reduction are given in the following table). It can be seen that FF-Reversing and Random-Load control achieved more pattern reduction than SEQ-OB with FULL observation to achieve 90% fault coverage, which are 6,050 and 6,350 scan-in patterns by FF-Reversing, and 1,900 and 2,650 scan-in patterns by Random-Load with FULL observation and partial observation, respectively.

In Table 2, we give the detailed scan-in pattern reduction results for all circuits. In order to achieve 90% stuck-at fault coverage required by ASIL D of ISO26262, we perform the experiments by increasing the number of scan-in patterns until 90% stuck-at fault coverage is achieved or one million scan-in patterns are used. The number of scan-in patterns required to achieve 90% stuck-at fault coverage by traditional scan test, 10-cycles capture test w/o SEQ-OB (MUL) are given in the second and the third columns.
Table 2: Scan-in pattern reduction by FF-CPI

| Circuit | SCAN | MUL | FULL: Observing all FFs by SEQ-OB | P-OB20: Observing 20% of FFs by SEQ-OB |
|---------|------|-----|-----------------------------------|--------------------------------------|
|         | w/o  | w/o | FF-CPI | by FF-Reversing | by Random-Load | FF-CPI | by FF-Reversing | by Random-Load |
| s9234   | >1M  | >1M | 94,900 | 10.5X | 13.8X | 17.0X | 14.7X | 26,750 | 37.4X | 20.9X | 18.8X | 99,000 | 10.1X | 12.7X | 15.6X | 14.2X | 37.4X | 19.6X | 18.5X |
| s13207  | >1M  | >1M | 11,600 | 1.8X  | 1.8X  | 3.4X  | 3.6X  | 1.9X  | 10.8X | 9.6X  | 11,600 | 1.8X  | 1.0X  | 3.2X  | 1.5X  | 1.6X  | 7.8X  | 7.4X  |
| b14     | >1M  | >1M | 5200   | 9.6X  | 19.2X | 14.4X | 16.4X | 19.2X | 12.8X | 19.2X | 56250  | 8.2X  | 16.4X | 14.4X | 14.4X | 12.8X | 10.5X | 14.4X |
| s38417  | >1M  | >1M | 4200   | 19.1X | 10.8X | 88.9% | 89.6% | 18.4X | 129.0X | 93.6% | 17.8X | 10.8X | 88.9% | 89.6% | 17.8X | 119.8X | 83.0X |
| b20     | 1M   | 69.8%| 237.150| 6.0X  | 2.7X  | 3.2X  | 2.8X  | 2.6X  | 86.2X | 118.3X | 238.1X | 170.9X | 4.3X  | 238.1X | 212.8X | 96.6X |

For each circuit, if one million scan-in patterns cannot achieve 90% stuck-at fault coverage, the final fault coverage achieved by one million scan-in patterns are given following with 1M. In the group column denoted by “FULL”, the sub-columns show the number of scan-in patterns to achieve 90% stuck-at fault coverage and the pattern compression rate compared with scan test, by full observation of SEQ-OB w/o FF-CPI, by FF-CPI with FF-Reversing control and Random-Load control when TrPI, LIMA and HEM selection methods are applied, respectively. In the following group column denoted by “P-OB20”, the experiments are conducted by observing only 20% of FFs for each circuit by SEQ-OB technique for the hardware overhead concern.

For all circuits, it is very difficult to achieve 90% stuck-at fault coverage just using a pure scan test w/o any DFT techniques. Multi-cycle test is a promising DFT technique to reduce the scan-in patterns for scan test. Regarding the results of multi-cycle test, in s38417, b14, b15 and b20, multi-cycle test shows significant scan-in pattern reduction compared to scan test. However, fault effects vanishing problem caused the fault coverage loss and expanded the number of scan-in patterns to achieve 90% stuck-at fault coverage in s9234 and s13207. SEQ-OB technique with full observation (column denoted by “w/o FF-CPI”) can avoid the fault effects vanishing problem, which achieved more scan-in pattern reduction with larger pattern compression rate than pure multi-cycle test.

As discussed in Sect. 2, the fault degradation problem is another factor to obstruct the scan-in pattern reduction by multi-cycle test. Compared to the results of SEQ-OB, FF-CPI technique shows further scan-in pattern reduction for all circuits. For s9234 and s13207, while multi-cycle test caused fault coverage loss due to fault effects vanishing problem and SEQ-OB technique has improved the scan-in pattern reduction (10.5X for s9234, 1.8X for s13207), FF-CPI technique achieved more scan-in pattern reduction than SEQ-OB which are at the most 17.0X and 37.4X for s9234, 3.6X and 10.8X for s13207 by FF-Reversing or Random-Load, respectively. For other circuits, even though multi-cycle test with SEQ-OB already achieved large scan-in pattern reduction (9.6X for s38417, 19.1X for b14, 238.1X for b15, 6.0X for b20) compared to scan test, FF-CPI technique also show the effect to further improve the scan-in pattern reduction, which are at most 19.2X, 129.0X, 241.0X and 118.3X by FF-Reversing or Random-Load, respectively.

Regarding the capture pattern control methods referred to FF-Reversing and Random-Load, the results of the FULL observation in Table 2 show that the Random-Load control shows more effective than FF-Reversing control for most circuits. This is because that FF-Reversing control just reverses the captured value of the candidate FFs per cycle. The values of all candidate FFs selected for CPI are changed simultaneously at each cycle, with the same state such as all-zero or all-one which cannot directly improve the randomness for capture patterns. In contrast, Random-Load control directly load a random vector to the candidate FFs every cycle, which can directly improve randomness for the capture patterns during capture operations. As described in Sect. 3, Random-Load requires additional memory or on-chip test pattern generator to store or generate random vectors for FF-CPI that would cause large area overhead and complicates the timing design of a capture operation. While FF-Reversing Control shows less effective for scan-in pattern reduction than Random-Load control, it still achieved a significant scan-in pattern reduction compared to SEQ-OB technique. Moreover, the circuit for FF-Reversing control is very easy to design with very small hardware overhead that can achieve a good trade-off between scan-in pattern reduction and hardware overhead for practical use. Based on the works on the implementation of FF-CPI in commercial automotive ECU supported by Renesas Electronic corp., the gate account required by FF-CPI (either FF-Reversing CPI or Random-load CPI) for each FF is smaller than a multiplexer on average.

Regarding the FF selection methods referred to TrPI, LIMA and HEM for FF-CPI proposed in Sect. 4, the effects of TrPI and LIMA for scan-in pattern reduction might be depend on the circuits used for testing. In s9234 and s13207, LIMA achieved more scan-in pattern reduction (17.0X and 3.4X) than TrPI (13.8X and 1.8X) by FF-Reversing Control, however, which (14.4X and 204.1X) becomes worse than TrPI (19.2X and 241.0X) in s38417 and b15. HEM
comprehensively evaluates the metrics of TrPI and LIMA for ranking the FFs for FF-CPI, it is therefore helpful to relax the structure dependence of FF selection algorithm so as to select the optimal candidates of FFs for general circuits. As the results of FF-Reversing Control with HEM shown in Table 1, for almost all circuits, HEM can achieve better scan-in pattern reduction compared to SEQ-OB. For b20 circuit, HEM achieved 118.3X scan-in pattern reduction by Random-Load control, which are 2.6X and 86.2X by TrPI and LIMA. However, for b15 circuit, HEM caused significant decrease of scan-in pattern reduction compared to TrPI and LIMA. We analyzed the evaluation values of each FF derived by TrPI and LIMA, and found that the FFs with high rank by TrPI have very small evaluation value derived by LIMA, and vice versa. Since HEM method comprehensively evaluate the metrics derived by TrPI and LIMA with user-specified weights (here is equal weight: 0.5), the FFs with both large evaluation value of TrPI and LIMA are assigned with high rank by HEM, and such FFs might not helpful to fault detection by FF-CPI. The results imply that appropriate weight values are crucial to improve the effect of HEM. Exhaustive experiments to investigate the effect of HEM by fine-tuning the weight values are too much time-consuming and inefficient. An efficient determination algorithm for the weight values of HEM will be a part of our future work.

SEQ-OB with full observation is effective to avoid fault effects vanishing problem that highlights the effect of FF-CPI technique. However, observing all FFs will cause too large hardware overhead to practical use. In the group column denoted by “P-OB20” of Table 2, only 20% of FFs of each circuit are selected for SEQ-OB to reduce the hardware overhead. Compared to SEQ-OB with full observation, the scan-in pattern reduction of FF-CPI with partial observation decreased a little, however, which are still very remarkable achievement compared to the SEQ-OB w/o FF-CPI.

5.3 Efficiency of the FF Selection Methods for FF-CPI

When implement FF-CPI technique into a very large circuit such as a commercial automotive device with several million gates, an efficient FF selection method is required to shorten the TAT of DFT period. In Sect. 4, we have proposed three methods TrPI, LIMA and HEM to select the candidate FFs for FF-CPI. Table 3 shows the runtime of these methods to make the rank of FFs for all circuits. Evaluation experiments are conducted on CPU (Intel® Xeon® L5240 @ 3.0GHz) with 32GB memory. The results show that TrPI based selection method takes the longest time to generate the rank of FFs, the reason why is that TrPI needs to transform the CUT to M-cycles (here M = 10) time-frame expansion circuit. The TrPI also calculates the state transition probability of each gate at each timeframe which is very time-consuming. On the other hand, LIMA analyzes the structure of the circuit by forward-tracing started from the output of FFs, and ended at the input of FFs, therefore the elapsed time of LIMA is small. HEM comprehensively evaluates the data derived by TrPI and LIMA to make the rank of FFs for FF-CPI. While the ranking process spend very short time, preparing the data by TrPI and LIMA causes the total runtime becoming longer. We believe that LIMA should be a better solution for shortening the TAT of DFT period for a very large circuit.

6. Conclusions

Multi-cycle Test looks a promising way to reduce the test volume of POST (Power-on Self-Test) to meet the requirement of stuck-at fault coverage specified by ISO26262 standard.

In this paper, we first reveal the Fault Detection Degradation problem that would obstruct the effect of multi-cycle test for scan-in pattern reduction. Based on our analysis, we propose the novel solution named FF-CPI technique to overcome the Fault Detection Degradation problem. Two methods referred to the FF-Reversing Control and the Random-Load Control are proposed to enhance the additional stuck-at fault detection capability of capture patterns by modifying the value of a part of FFs directly during capture operation. Moreover, we propose three FF selection methods for FF-CPI by analyzing the circuit structure w/o any simulation to shorten the period of DFT.

Finally, we show the experimental results on benchmark circuits to validate that the proposed methods can further reduce the number of scan-in patterns for achieving 90% stuck-at fault coverage compared to the SEQ-OB method presented in our previous paper[14, 15]. As a result, the proposed method is helpful to further shorten the TAT of POST.

In addition, the results of partial observation of SEQ-OB confirmed that just observing a small number of FFs can also achieve a significant scan-in pattern reduction by FF-CPI, which contributes to the practical use of FF-CPI for a very large commercial automotive device with smaller hardware overhead.

Acknowledgments

This work was supported in part by KAKENHI (19K11878).

References

[1] ISO26262-5:2011, “Road vehicles - Functional Safety - Part 5: Product development at the hardware level,” Online Browsing Platform, https://www.iso.org/obp/ui/#iso:std:iso:26262:-5:ed-1:v1:en,
S. Wang, T. Aono, Y. Higami, H. Takahashi, H. Iwata, Y. Maeda, and J. Matsushima, “Capture-Pattern-Control to Address the Fault Detection Degradation Problem of Multi-Cycle Test in Logic BIST,” Proc. IEEE Asian Test Symposium, Hefei, pp.155–160, 2018. DOI:10.1109/ats.2018.00038

K. Yoon, “A Reconciliation among Discrete Compromise Solutions,” Journal of Operational Research Society, vol.38, no.3, pp.277–286, 1987. DOI: 10.2307/2581948

Senling Wang received M.S. and Ph.D. degrees from the Department of Computer Science and Electronics, Kyushu Institute of Technology, Japan, in 2011 and 2014, respectively. Currently, he serves as a Senior Assistant Professor in the Graduate School of Science and Engineering, Ehime University, Japan. His research interest includes Field testing, Low power testing, and Design for Testability. He is a member of the IEEE, IEICE, IPSJ and JIEP.

Hanan T. Al-Awadhi received the B.S. degree in Computer Science from Sana’a University, Yemen in 2002, and M.S. degree in Information Technology Management from Staffordshire University (U.K.) collaboration with APU University, Malaysia in 2012. In 2013 she has been a researcher student of the Department of Computer Science in Ehime University, Japan. Currently she is pursuing the Ph.D. degree in Department of Computer sciences in Ehime University, Japan. Her research interest in Field testing and Machine learning.

Tomoki Aono received the B.S. degrees in the electronic information engineering from Ehime University, in 2018. Currently, he is a Master student with the Graduate School of Science and Engineering, Ehime University. His research interest is design for testability.
Yoshinobu Higami received his B.E., M.E., and D.E. degrees from Osaka University in 1991, 1993 and 1996, respectively. Currently he is a full professor at the Graduate School of Science and Engineering, Ehime University. In 1998 and 2006, he was also an honorary fellow at University of Wisconsin Madison, U.S.A. He received the IEICE Best Paper Award in 2005 and 2012, and Best Paper Award of IEEE Computer Society Annual Symposium on VLSI in 2014. His research interests include test generation, design for testability and fault diagnosis of logic circuits. He is a senior member of the IEEE and a member of IEICE.

Hiroshi Takahashi received the Ph.D. degree from Ehime University, Japan in 1996. Since 2010, he has been a Full Professor at Ehime University. From 2018 to 2019, he served as Dean of Faculty of Engineering and Graduate School of Science and Engineering at Ehime University. From May 2000 to March 2001, he was a research fellow at the University of Wisconsin-Madison, USA. He received the IEICE Best Paper Award in 2012, the best Paper Award of IEEE Computer Society Annual Symposium on VLSI in 2014 and the Reliability Engineering Association of Japan Takagi Prize in 2016. His research interests are test generation and fault diagnosis for digital systems. Prof. Takahashi is a senior member of the IEEE, IEICE, and IPSJ. He served as the Program Chair of the 2012 IEEE Asian Test Symposium. He also served as the General Co-Chair of the 2016 IEEE Asian Test Symposium.

Hiroyuki Iwata received the B.E. degree in electrical and computer engineering from Yokohama National University, Yokohama, Japan, in 2003, and the M.E. and Ph.D. degrees in information science from Nara Institute of Science and Technology, Nara, Japan, in 2005 and 2007, respectively. He joined Renesas Technology Corp., Tokyo, Japan in 2007. Since 2010, he has been with Renesas Electronics Corporation, Tokyo. He has been involved in developing DFT methodology and EDA tools. His research interests are VLSI CAD and design for testability.

Yoichi Maeda received the B.E. and M.E. degrees in engineering from the University of Electro-Communications, Tokyo, Japan, in 2000 and 2002, respectively. He joined Hitachi Ltd., Tokyo Japan, in 2002. Since 2010, he has been with Renesas Electronics Corporation, Tokyo. He is currently leading the development of functional safety function using design-for-test technologies for automotive SoCs and MCUs. He is a member of IEICE.

Jun Matsushima received Foundation degree from National Institute of Technology, Nara college, Nara, Japan in 1988. He joined General Purpose Computer Division, Hitachi, Ltd. in 1988, and moved to Renesas Technology Corporation in 2003. Currently, he is with Renesas Electronics Corporation, Tokyo, where he has been responsible for DFT solution and methodology. Since 1990, He is a regular member of Information Processing Society of Japan.