Field Test for Ensuring the Functional Safety of Automotive System

A thesis submitted for the degree of doctor of philosophy

By

AL Awadhi Hanan

Supervisor:

Hiroshi Takahashi (Full Professor)

Department of Computer Science
Graduate School of Science and Engineering

Ehime University

博　士　（　工学　）

Doctor of Engineering

(2020)

ACKNOWLEDGEMENT

First, praises and thanks to GOD, the Almighty, for his shower of blessings throughout my research work to complete the research successfully.

I would like to express my deep and sincere gratitude to my research supervisor, Prof. Hiroshi Takahashi for giving me the opportunity to do the research and providing invaluable guidance, valuable resources and knowledge throughout this research. His dynamism, vision and motivation have deeply inspired me.

I am really grateful to Prof. Senling Wang for his sincerity, patience and guidance throughout this research. He has taught me the methodology to carry out the research and to present the research work as clearly as possible.

I appreciate each of Prof. Yoshinobu Higami, Prof. Hiroshi Kai and Prof. Hirohisa Aman for their collaboration and valuable knowledge during the research. I'm thankful to all professors in computer science and information system departments for their information and support. It was a great privilege and honor to study under their guidance. I would like to thank all my classmates' laboratory for their collaboration in this research. I also would like to thank all teachers and stuff in Ehime University for their empathy and great sense of humor and support during my study.

I'm extremely grateful to my husband for his support and his patience. He was encouraging and supporting me during this research, and he has offered the great environment to carry out the research. I am extending my heartfelt thanks to my son who was my motivation to continue this research. Finally, I express my thanks to my family for their prayer and their love that have encouraged me to complete this thesis successfully.

Abstract

Deep submicron devices such as microcontroller unit and graphical processing unit achieve high performance in real-time high-volume and complex data required for advanced automotive, aircraft, and intelligent transportation systems. However, the ever-shrinking of the deep submicron process makes the devices embedded in such systems vulnerable to aging phenomena. When a deep submicron device is put in actual use for a long time or works in a severe environment, various aging like HCI (Hot Carrier Injection), BTI (Bias Temperature Instability), and NBTI (Negative Bias Temperature Instability) threatens its reliability [37][38].

The device's reliability problem relating to aging phenomena can be relaxed using elaborate manufacturing tests such as burn-in or stress testing. It can also be achieved through redundancy design of the device's hardware and timing margin at the design phase [39]. However, the excessive burn-in test or voltage stress test deteriorates the device lifecycle, so the redundancy design results in high cost and performance degradation [40].

Field Testing is an efficient way to detect aging-induced faults by executing the test while the system is in an idle state, suspended or power-on/off to ensure its reliability. Power-on self-test (POST) is a novel field test technique that applies to many high-demand reliability systems. Generally, the POST executes at system start-up to test the system's critical components before commencing functional operations. The system's real-time state, including the aging-induced faults, can be checked/detected in advance to avoid failures during POST operation. Although POST operations required a limited testing time (e.g., <50ms for an automotive system), large-volume test data is not acceptable to apply within the limited test time at system start-up. Besides, POST requires a high fault coverage target to meet automotive

device testing. For example, a minimum of 90% fault coverage is necessary to satisfy the ISO 26262 ASIL D standard.

Two test strategies are used to implement the POST for advanced systems. The first strategy, **test partitioning technology,** splits the original large test set into subsets to fit the limited test application time and applies the test subsets every time the system starts. The second strategy, **fault detection enhancement technology,** reduces the test volume (test compression) to gain a target fault coverage by improving test quality patterns using design for testability (DFT) technology.

This study focuses on the two POST mentioned above application test strategies to develop the automotive system. We propose the corresponding test techniques to improve test partitioning's technology quality test and the fault detection enhancement technology, respectively.

The major problem of test partitioning is the loss of the fault coverage (FC) and increase in faults detection latency due to subsets missing test patterns. For the FC loss, the original test set needs to be partitioned into several smaller subsets to meet the application test time requirement. An aging-induced fault during the circuit operation may not be detected immediately until the test is applied to cause a system failure. So, it is necessary to improve the FC for each test subset as far as possible to guarantee reliability. For fault detection latency, each subset fault's missing test patterns may not be detected at the following test session right after it occurs. The fault effect would be propagated during a time interval from its occurrence to the detection. Although system failure does not necessarily happen as soon as fault alerts, the longer detection time interval would cause a higher chance of failure. Hence, minimizing the detection time interval for each fault is crucial to guarantee reliability.

We propose two approaches of test pattern partitioning to address these challenges [4][49]. First, we select the faults, which possibly have high aging speed as the ***risky faults***, and propose a greedy pattern partitioning method to improve the detection latency (mean time fault detection (MTFD) and the FC for the risky faults [4]. Experimental results on benchmark circuits demonstrate that the proposed method achieves a 6.8% FC increase with 17.7% MTFD reduction on average (with an initial test set partitioned into 10 subsets compared to random partitioning. Next, we utilize machine learning techniques: the simulated annealing algorithm (SA) and support vector machines model (SVM) for pattern partitioning to achieve optimal partitioning to minimize the aging-induced faults detection latency, MTFD [49].

Experimental results on the benchmark circuit demonstrate that the proposed SA and SVM-based pattern partitioning (within shorter processing time) achieve the same MTFD reduction compared to the genetic algorithm-based partitioning.

Regarding the fault detection enhancement technology for POST, we introduce the multi-cycle test to logic built-in self-test scheme to reduce the required root test data volume to achieve high FC, as specified by ISO 26262 for automotive device testing. The multi-cycle test allows circuit under test (CUT) responses to be reused as test stimuli for additional faults detection before applying root test data. It can reduce the number of root test data to achieve a target FC. However, we raise two key challenges: fault effects vanishing (FEV) problem and capture patterns fault detection degradation (FDD). These challenges obstruct the multi-cycle test effect in reducing the root test data to minimize the POST test application time (TAT).

The FEV problem denotes the fault effects excited at some intermediate capture cycles, which might disappear before their effects are propagated to the final capture cycle for observation due to a long-expanded propagation path that would cause FC loss. To address the FEV problem, we propose a Sequential Observation (SEQ-OB) DFT technique using fault-detection-strengthened (FDS). The proposed SEQ-OB is expected to strengthen the multi-cycle test fault detection capability by directly observing the values of a small part of flip-flops (FFs) at each capture cycle [31][32]. Also, we develop the underlying technologies, including the FDS FF design and an original in-house tool named FEV point-(TPI) Test Point Insertion (FVP-TPI), to compute the most effective FDS_FFs insertion point.

The experimental results of the commercial electronic control unit circuit (250k gates and 10k Flip-Flops) show that the proposed SEQ-OB with FDS_FFs significantly improves the FC (>95%). Also, it reduces the number of scan-in patterns (e.g., 2.4X~3.1X compression) to achieve 90% FC.

The FDD capture patterns denote the decrease capture pattern capability (the CUT test responses) to detect additional faults. To overcome the FDD problem, we propose a DFT method named FF-Control Point Insertion (FF-CPI) technique by modifying the scan FFs captured values during capture operation [35]. Also, we propose methods to determine the FFs candidate for FF-CPI to achieve more fault detection by analyzing the circuit structure w/o any simulations to minimize the DFT development period [47].

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) to achieve the specified target FC compared to the SEQ-

OB method (at most 12.5X). These results confirm that the proposed FF-CPI is suitable to minimize the TAT of POST further.

Table of Contents

ACKNOWLEDGEMENT .... **II**

ABSTRACT .... **III**

TABLE OF CONTENTS .... **VIII**

LIST OF FIGURE .... **X**

LIST OF TABLE .... **XI**

LIST OF ABBREVIATIONS .... **XII**

LIST OF PUBLICATIONS .... **XIV**

1 INTRODUCTION **1**

1.1 BACKGROUND 1
1.2 OBJECTIVES 2
1.2.1 Strategy 1: Test Partitioning Technology 2
1.2.2 Strategy 2: Fault Detection Enhancement DFT Technology 3
1.3 DISSERTATION STRUCTURE 4

2 PRELIMINARY **5**

2.1 LSI TEST 5
2.2 DESIGN FOR TESTABILITY (DFT) 5
2.2.1 Scan Design 6
2.2.2 Built-in Self-test (BIST). 7
2.3 FAULT MODELS 8
2.3.1 Stuck-At Faults Model 8
2.3.2 Transition Faults Model 8
2.4 FAULT SIMULATION 10

3 AGING INDUCED RELIABILITY CHALLENGES AND THE SOLUTION **12**

3.1 AGING ISSUE 12
3.2 FIELD TEST AND CHALLENGES 13
3.3 POWER-ON SELF-TEST (POST) 14

4 TEST PARTITIONING TECHNOLOGY FOR POST **15**

4.1 CONCEPT OF TEST PARTITION 15
4.2 THE PREVIOUS WORKS ON TEST PARTITIONING AND THE PROBLEM 15
4.3 APPROACH 1: PATTERN PARTITIONING CONSIDERING THE AGING SPEED 16
4.3.1 Aging Speed in the Field 17
4.3.2 Pattern partition method 17
4.3.3 Experimental results 22
4.4 APPROACH 2: TEST PARTITIONING UTILIZING THE MACHINE LEARNING 24
4.4.1 The Problem of Pattern Partitioning 24
4.4.2 Machine Learning Algorithm 25
4.4.3 Machine Learning Based Test Partitioning 25
4.4.4 Experimental Results 30
4.5 CONCLUSIONS 31

5 FAULT DETECTION ENHANCEMENT DFT FOR POST **32**

5.1 BACKGROUD OF POST 32
5.1.1 Functional Safety Standard: ISO26262 32
5.1.2 Previous DFT techniques for POST 33
5.2 MULTI-CYCLE TESTING 33
5.2.1 The Benefit of Multi-Cycle Test 34

5.2.2 The problems of Multi-cycle test 36
5.3 DFT APPROACHE TO ADDRESS FEV PROBLEM 37
5.3.1 Sequential Observation for Multi-cycle test 37
5.3.2 Problems of FF Selection for Observing 38
5.3.3 Evaluation Methods for Fault effects vanishing Point FF 40
5.3.4 FDS-FF Selection for Sequential Observation 43
5.4 EXPERIMENTAL RESULTS 44
5.5 CASE STUDY 46
5.6 CONCLUSIONS 48

6 DFT METHOD TO ADDRESS FDD PROBLEM 49

6.1 ANALYSIS OF FDD PROBLEM 49
6.2 FF-CONTROL POINT INSERTION (FF-CPI) FOR MULTI-CYCLE TEST 53
6.3 FF-CONTROL POINT INSERTION TECHNIQUE 57
6.3.1 FF-Reversing CPI 58
6.3.2 Random-Load 58
6.4 FFS SELECTION FOR FF-CONTROL POINT INSERTION 59
6.4.1 Method 1: Transition Probability Increment (TrPI) 59
6.4.2 Method 2: Logic Impact Area of FFs (LIMA) 61
6.4.3 Method 3: Hybrid Evaluation Metric (HEM) by TOPSIS 61
6.5 EVALUATION EXPERIMENTS 63
6.5.1 Experimental setup 63
6.5.2 Scan-in pattern Reduction by FF-CPI 64
6.5.3 Efficiency of the FF selection methods for FF-CPI 68
6.6 CONCLUSIONS 69

7 CONCLUSION 71

8 REFERENCES 74

List of Figure

Figure 2.1 Basic Principle of LSI Testing [1] 5
Figure 2.2 transforming a sequential circuit (a) to scan design circuit (b) [1] ... 6
Figure 2.3 Built-In Self-Test Architecture [45] 7
Figure 2.4 Four-Stage LFSR [1] 7
Figure 2.5 Four-Stage MISR [1] 7
Figure 2.6 Stuck-At Faults Model [1] 8
Figure 2.7 Transition Faults Model [3] 9
Figure 2.8 (a) Launch-off-Capture (LoC) and (b) Launch-off-shift (LoS) [46] 10
Figure 2.9 Fault Simulation Scheme [45] 11
Figure 3.1. Bathtub curve for LSI life cycle in reliability engineering 13
Figure 4.1 Test Partitioning for Field Testing 15
Figure 4.2 Switching probability computations 18
Figure 4.3 MTFD model of partitioning test 19
Figure 4.4 Flow chart for Simulated Annealing partitioning method 27
Figure 4.5 Flow chart for SVM based patter partition 28
Figure 5.1 Clock Design for Scan Testing 34
Figure 5.2 Stuck-at Fault Coverage of b13 35
Figure 5.3 Fault effects vanishing in time-frame expansion circuit 37
Figure 5.4 Sequential Observation for Multi-Cycle Test 38
Figure 5.5 Definition of fault effects vanishing point FF 39
Figure 5.6 an example to calculate the Gate-FF Connection Complexity 41
Figure 5.7 Structural connections between gates and FFs 42
Figure 5.8. Branch Reachable Rate 43
Figure 5.9 Stuck-at fault coverage in multi-cycle test 47
Figure 5.10 Stuck-at fault coverage of 10-cycle test by sequential observation 47
Figure 6.1 Example to calculate the TpC for a FF 51
Figure 6.2 TpC of each FF at capture cycles 51
Figure 6.3 Average TpC of FFs & Number of additional faults detected at capture cycles 52
Figure 6.4 TpC of each FF at capture cycles by reversing capture value of FF6 57
Figure 6.5 Average TpC of FFs & Number of additional faults detected 57
at capture cycles by reversing capture value of FF6 57
Figure 6.7 Random-Load for captures pattern control [35] 58
Figure 6.6 FF-Reversing captures pattern control [35] 58
Figure 6.8 computing the state transition probability by COP [35] 60
Figure 6.9 Evaluation for LIMA [35] 62
Figure 6.10 Fault coverage curve of s13207 65

List of Table

TABLE 4.1 PROBABILISTIC CONTROLLABILITY COMPUTATIONS 18
TABLE 4.2 PATTERNS DETECTED FAULTS 21
TABLE 4.3 SIMILARITY OF SCAN-IN PATTERNS 21
TABLE 4.4 PROBABILISTIC CONTROLLABILITY COMPUTATIONS 23
TABLE 4.5 EXPERIMENTAL RESULTS FOR N_{SET}=10... 23
TABLE 4.6 EXPERIMENTAL RESULTS FOR N_{SET} =20. 24
TABLE 4.7 CALCULATION OF HAMMING DISTANCE 29
TABLE 4.8 THE MTFD OF GA, SA AND SVM BASED PATTERN PARTITIONING 31
TABLE 4.9 THE RUNTIME (SEC) OF GA, SA AND SVM BASED PATTERN PARTITIONING 31
TABLE 5.1 FAULT COVERAGE IMPROVEMENT AND PATTERN REDUCTION BY SEQUENTIAL OBSERVATION 46
TABLE 5.2 INFORMATION ECU CIRCUIT 46
TABLE 5.3 PATTERN REDUCTION FOR TESTING THE ECU CIRCUIT 48
TABLE 6.1 THE NUMBER OF THE ADDITIONAL FAULTS DETECTED AT CAPTURE CYCLES (308 STUCK-AT FAULTS OF THE COMBINATIONAL CIRCUIT OF S298)................................ 56
TABLE 6.2 PATTERN REDUCTION BY FF-CPI.. 68
TABLE 6.3 RUNTIME FOR FF RANKING 69

List of Abbreviations

ICs	Integrated circuits
SSI	Small-Scale integration
MSI	Medium-scale integration
LSI	large-scale integration
VLSI	Very-large-scale integration
CUT	Circuit under test
NBTI	Negative Bias Temperature Instability
TDDB	Time Dependent Dielectric Breakdown
HCI	Hot carrier injection
POST	Power-On-Self-Test
TAT	Test Application Time
DFT	Design for testability
ML	Machine Learning
SA	Simulated Annealing
SVMs	Support Vector Machines
TPI	Test Point Insertion
LSSD	Level-Sensitive Scan Design
TPG	Test Pattern Generator
PIs	Primary Inputs
POs	Primary Outputs
TFM	Transition Fault Model
LoS	Launch-off-Shift
LoC	Launch-off-Capture
GA	Genetic Algorithm
ICT	Information Communication Technology
RFs	Risky Faults
FEO	Faults Easy to Occur
FHO	Faults Hard to Occur
SWP	Switching Probability

SOT	Sequential Observation Technique.
FDS_FF	Fault-Detection Strengthened FF
CC	Connection Complexity
OC-SVM	One-Class Support Vector Machine
HD	Hamming Distance
FDD	Fault Detection Degradation
TMS	Tri-Modal Scan
LIMA	Logic Impact Area
TrPI	Transition Probability Increment
HEM	Hybrid Evaluation Metric
TpC	Transitions per Cycle

List of Publications

Journal or Transactions (peer reviewed publication):

1. 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, Senling Wang, Yoshinobu Higami, Hiroshi Takahashi, Hiroyuki Iwata, Yoichi Maeda, Jun Matsushima, IEICE Transaction on Information and Systems, Vol. E103.D, NO. 11, pp.2289-2301, 2020, **DOI**: 10.1587/transinf.2019EDP7235 (Peer-reviewed)

Conference Papers (peer reviewed publications):

2. Feasibility of Machine Learning Algorithm for Test Partitioning
   Senling Wang, Hanan T. Al-Awadhi, Masatoshi Aohagi, Yoshinobu Higami and Hiroshi Takahashi
   International Technical Conference on Circuits, Systems, Computers, and Communications (ITC-CSCC2019), pp. 1-4, 2019 (Peer-reviewed)
3. Pattern Partitioning based Field Testing for Improving the Detection Latency of Aging-induced Delay Faults
   Hanan Al Awadhi, Senling Wang, Yoshinobu Higami and Hiroshi Takahashi
   International Technical Conference on Circuits, Systems, Computers, and Communications (ITC-CSCC2017), pp.21-24, 2017 (Peer-reviewed)
4. Pattern Partitioning for Field Testing Considering the Aging speed
   Hanan Al Awadhi, Senling Wang, Yoshinobu Higami and Hiroshi Takahashi
   Workshop on RTLT and High-Level Testing (WRTLT2016), pp.72-76, 2016 (Peer-reviewed)
5. Structure-Based Methods for Selecting Fault-Detection-Strengthened FF under Multi-Cycle Test with Sequential Observation
   Senling Wang, Hanan T. Al-Awadhi, Soh Hamada, Yoshinobu Higami and Hiroshi Takahashi, Hiroyuki Iwata and Jun Matsushima
   IEEE Asian Test Symposium (ATS), pp.209-214, 2016 (Peer-reviewed)

others:

6. マルチサイクルテストにおける故障検出強化のための FF トグル制御ポイントの選択法
   青野智己, Hanan T.Al-Awadhi, 王　森レイ, 樋上喜信, 高橋　寛, 岩田浩幸, 前田洋一, 松嶋　潤
   信学技報, Vol. 118, No. 456, pp. 49 – 54, 2019 年 2 月.

Chapter 1

1 Introduction

1.1 Background

Integrated circuits, microchips, are accompanied by the need to test devices. The ***Small-Scale integration*** devices consist of tens of transistors in the early 1960s, whereas ***medium-scale integration*** devices contain hundreds of transistors in the late 1960s. ***Large-scale integration (LSI)*** devices scale to tens of thousands of transistors in the 1970s, creating some challenges for testing these devices. In the 1980s, ***very-large-scale integration (VLSI)*** devices, which contain hundreds of thousands of transistors are introduced to increase the testing challenges. Today's computers and other electronic devices like mobile phones come with millions of transistors, resulting in shrinking the feature size of transistors and interconnecting wires. The reduction in the feature size of transistors results in increased operating frequencies that led to increasing the clock speeds. On the other hand, the reduction of feature size increases the chip defect and faulty chip, affecting the reliability; Therefore, system testing must pass through several manufacturing stages, production stages, and testing during system operation to ensure a fault-free system.

The VLSI testing is important for designers, test engineers, manufacturers, and end-users. The testing consists of scan-in patterns that apply to the circuit under test (CUT) input and analyze all test stimuli's output responses to ensure defect-free chip. The circuits that match with output responses are considered faulty chips. All the faulty chips must be removed during the production test and then ship only the clean chips to the customers/end-users. However, when the clean chips work for long times within various temperatures/changeable environment, aging phenomena problems such as negative bias temperature instability (NBTI), dependent dielectric breakdown (TDDB), and hot carrier injection occur [5] [6]. These aging phenomena can cause latent faults (multiple faults would violate the safety goal) that degrade the deep submicron VLSI and evolve major faults with certain subsequent severe conditions during the system operation lifecycle. Field Testing is an important way to detect aging-induced faults by executing tests during

the devices' operation to ensure system reliability. A few field testing approaches have been proposed in the past, and they are classified into concurrent testing and non-concurrent testing. The concurrent testing approach checks the real state of the circuit during system operation and detects the aging faults or soft errors as soon as they occur. However, this approach requires large overhead and causes performance degradation due to the special circuit architecture and redundancy design in terms of hardware, timing, or information. Non-concurrent testing such as power-on self-test (POST) recently gains much attention and has been applied to many systems that require high reliability, such as an automotive system. Generally, non-concurrent testing is performed when the system is idle, suspended, or power-on/off, which helps detect the aging-induced faults with less impact on system performance and small hardware overhead.

A major challenge for applying non-concurrent field testing is that the test application time (TAT) is usually very short (e.g., TAT < 10～50 msec). Therefore, using complete testing with many test patterns within a limited test time is hard. Moreover, as required by ISO26262 standard [3], POST must achieve high fault coverage (FC) (e.g., >90%) for the latent faults to meet the standard functional safety requirement, and that makes the POST application difficult.

1.2 Objectives

To guarantee the VLSI reliability in field testing, POST is a good way to detect latent faults as they occur. However, the limitation of TAT field testing is a major challenge, as discussed earlier. In this dissertation, we demonstrate the POST problems and their proposed solutions. There are two strategies to implement POST for advanced systems.

1.2.1 Strategy 1: Test Partitioning Technology

The major problems of test partitioning are the FC loss and the increase of faults detection latency due to the subsets missing test patterns. As for the loss of FC, the original test set needs to be partitioned into several smaller subsets to meet the TAT requirement. Once an aging-induced fault occurs during circuit operation, it might not be detected immediately until a test is applied to cause a system failure. It is necessary to improve the FC for each test subset to guarantee reliability. For the fault detection latency, each subset fault's missing test patterns may not be detected during the test session right after it occurs. The fault effect propagates at a time interval from its occurrence to the

detection. Although a system failure is not usually caused as soon as a fault alert, the longer detection time interval could cause a high failure rate. Therefore, minimizing the detection time interval for each fault is essential to guarantee reliability.

In this study, we propose two approaches of test pattern partitioning to address these problems. First, we select the faults, which have high aging speed as the *risky faults*, and propose a greedy pattern partitioning method to improve the detection latency and the FC for the risky faults. Second, we utilize machine learning techniques: the simulated annealing algorithm (SA) and support vector machines model (SVM) for pattern partitioning. The techniques help us achieve optimal partitioning to minimize detection latency (mean time fault detection (MTFD) of aging-induced faults.

1.2.2 Strategy 2: Fault Detection Enhancement DFT Technology

In this strategy, we target to improve the POST test data quality. We introduce a multi-cycle test to the logic- built-in self-test (BIST) scheme to reduce the root test data volume required to achieve high FC. The multi-cycle test allows the CUT test responses to be reused as test stimuli for testing to detect additional faults before applying root test data. However, we raise two major challenges that obstruct the multi-cycle test's effect for minimizing the POST TAT. The challenges are fault effects vanishing (FEV) problem and capture patterns fault detection degradation (FDD).

The FEV problem suggests that fault effects excited at some intermediate capture cycles might disappear before their effects are propagated to the final capture cycle for observation due to a long-expanded propagation path that would cause FC loss. To address the FEV problem, we proposed a DFT technique, Fault-Detection-Strengthened (FDS) method. The proposed method strengthens the multi-cycle test’s fault detection capability by directly observing the values of a small part of flip-flops (FFs) at each capture cycle. Further, we developed the underlying technologies, including the FDS FF design and an original in-house tool named FEV point-TPI (FVP-TPI), to compute the most effective insertion point of FDS_FFs.

The captured patterns FDD denotes the decrease of the capability of capture pattern (the test responses of CUT) to detect more additional faults. To overcome the FDD problem, we propose FF-Control Point Insertion (FF-CPI) technique, a DFT method, by modifying the scan FFs captured values during the capture operation. Also, we propose

methods to evaluate the FFs for determining the candidate FFs for FF-CPI to achieve more fault detection, by analyzing the circuit structure w/o any simulations to minimize the DFT development period.

1.3 Dissertation Structure

This dissertation is organized as follows. Chapter 1 presents the introduction and introduces some concepts of LSI testing, DFT, fault models, fault simulation, and test generation in Chapter 2. We introduce aging phenomena and POST in Chapters 3 and discuss POST's multi-cycle test in Chapter 4. The solution to improve test quality for BIST, approach 1: fault detection enhancement for the POST, is presented in Chapter 5, whereas the solution to enhance the quality of test for BIST, approach 2: Solution for fault detection degradation problem is discussed in Chapter 6. Finally, in Chapter 7, we present the dissertation conclusion.

Chapter 2

2 Preliminary

In this chapter will be discussed the concepts, the main principles, and architectures for each of LSI test, DFT technique, fault modeling and fault simulation.

2.1 LSI Test

Large-Scale integration (LSI) is the process of integrating on semiconductor device. Testing is crucial to ensure the reliability of an IC (Integrated Chip) device [1]. A complete test procedure passes through several stages (manufacturing process, and operation process) which defined **IC's lifecycle**. To ensure the quality of an IC before it is shipped to the market, manufacturing tests including the functional test, structure test, burn-in, stress test are conducted to determine any faulty ICs. Only the pass chip/device will go to the packaging process. Figure 2.1 illustrates the basic procedure of an IC testing, where a set of data used for testing called the test pattern is applied to the accessible input pins of a CUT (Circuit Under Test), then comparing the particular output responses with the pre-simulated responses called the golden responses. A circuit can be considered to be fault-free if the particular responses matched to the golden one.

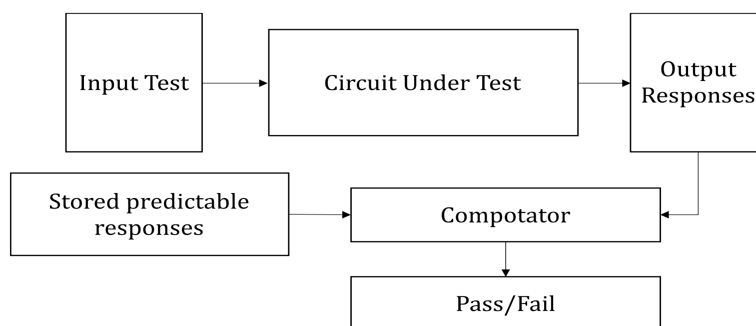


Figure 2.1 Basic Principle of LSI Testing [1]

2.2 Design for Testability (DFT)

With the progressing in the manufacturing technology, IC become more and more complicate to satisfy the demands on high-performance, multi-functions, high-speed and

low-power for IC. In a modern circuit, millions of logic gates and ten or hundred thousand sequential elements such as flip-flops or latches are embedded in very small silicon area (e.g.:1mm^2), and the accessible pins of an IC are extremely limited (e.g.: 100 ports). To control (observe) all the internal state of such a complicate circuit through the limited external pins is general impossible. DFT is a state-of-the-art to improve the controllability and observability of the internal logic of the circuit by modifying or adding special logic which are helpful to testing into the CUT, so as to make the CUT easier to test. The typical DFT techniques are as follows.

2.2.1 Scan Design

Scan Design is one of the most common techniques used in DFT methodology, the main aim of scan design is to improve the controllability and observability of the circuit by providing an easier way to set and observe the flip-flop in a sequential circuit. The structure of scan design is shown in Figure 2.2 [1].

Connecting the D Flip-Flops in the sequential circuit in serial by multiplexer will create a scan chain. Where, the input of the first FF of the scan chain is denoted by scan-in pin, and the output of the last FF of the scan chain is denoted by scan-out pin. A TC (Test Control) signal is used to control the operation mode of scan chain. There are three test modes during the scan designed circuit operation (normal mode, shift mode, and capture mode).

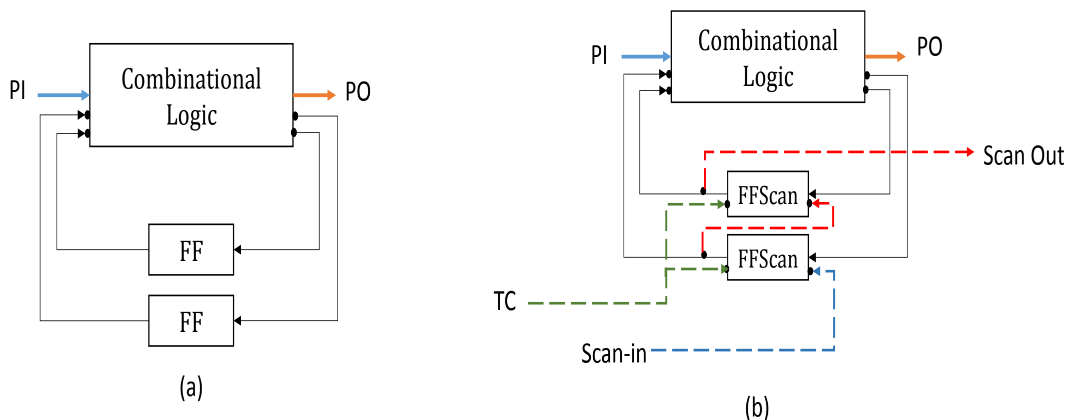


Figure 2.2 transforming a sequential circuit (a) to scan design circuit (b) [1]

2.2.2 Built-in Self-test (BIST).

BIST is a state-of-the-art to let the IC test itself. It is a solution to make the electrical testing of chip easy, fast and low cost, since the ATE testing becomes very complex, and high cost. BIST is used in field testing that because of the advantages of implementation such as shorting test time with a high-test quality [45]. Figure 2.3 shows BIST architecture.

In Logic-BIST architectures, Linear Feedback Shift Registers (LFSRs) and most of BIST used LFSR to generate the random patterns (TPG) [1].

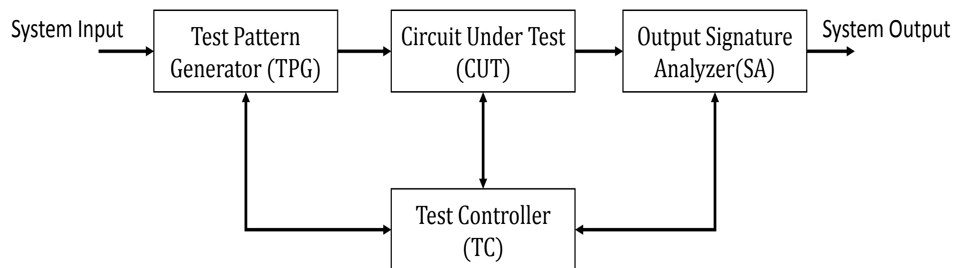


Figure 2.3 Built-In Self-Test Architecture [45]

Figure 2.4 shows the structure of LFSR. The signature analyzers (SAs) are commonly constructed from multiple-input signature registers (MISRs) [1] as shown in Figure 2.5. The MISR assists to improve the detection of defects, by applying large number of scan-in patterns. The design basically an LFSR has an extra XOR gate at the input of the flip-flop for compressing the output responses of the CUT into the LFSR during shift operation.

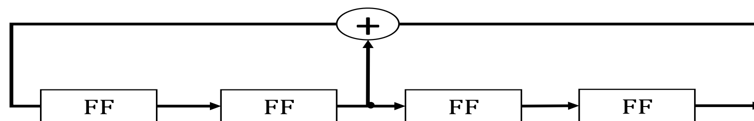


Figure 2.4 Four-Stage LFSR [1]

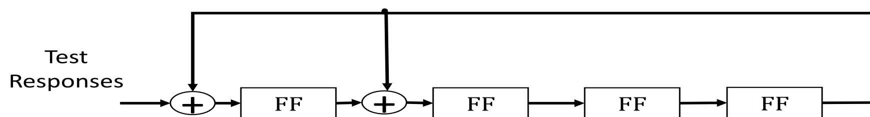


Figure 2.5 Four-Stage MISR [1]

2.3 Fault Models

Manufacturing process would generate various types of physical defects like short, open, bridge and etc. Generating the test to cover the electrical characteristics for all physical defects is difficult and impossible in general. For high quality testing, sophisticated fault model that can represent the behavior of a real defect in the CUT is necessary. A good fault model should satisfy two criteria: (1) can accurately reflect the defect behavior, and (2) be efficient to fault simulation for test pattern generation [1].

The following will introduce the most common fault models, the stuck-at fault model and the transistor faults model.

2.3.1 Stuck-At Faults Model

Stuck-at fault describe a faulty behavior on signal line such as primary inputs (PIs), primary outputs (POs), internal gate inputs and outputs, fan-out stems, and fan-out branches, in the CUT. The faulty signal line may be either logic 0 (stuck-at-0) or logic 1 (stuck-at-1). Figure 2.6 shows an example of stuck-at fault for a signal line [1]. Stuck-at fault model cannot represent the timing-related behavior such as the delay of signal propagation due to some resistive defects. Therefore, other faults models such as transition fault model are necessary which will be introduced in the following section.

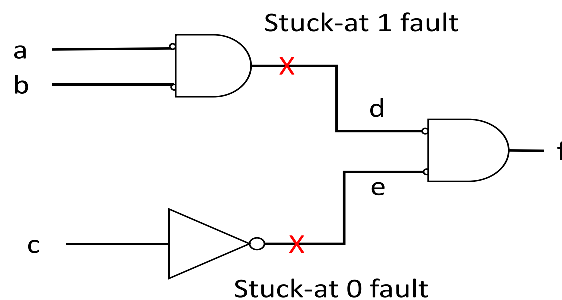


Figure 2.6 Stuck-At Faults Model [1]

2.3.2 Transition Faults Model

Transition Fault reffers to the delay of signal when the signal is propagation through wires, or logic gates. It is usually caused by some resistive defects such as resisitive open on a wire. A transition fault will occur when the output of a gate switching from 0(1) to 1(0) takes longer time than normal. Figure 2.7 shows an example of transition fault. There are two transition faults associated with each gate: a slow-to-rise fault and a

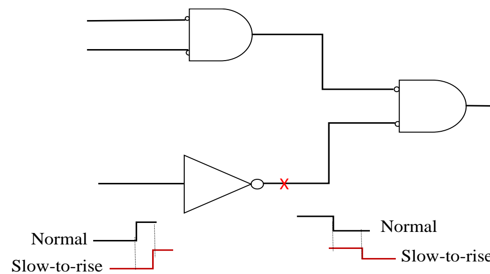


Figure 2.7 Transition Faults Model [3]

slow-to-fall fault. A slow-to-rise (slow-to-fall) fault denotes the transition from 0 to 1 (1 to 1) will not reach any output within the stipulated time. Detecting a transition fault requires at least two vectors (V1,V2). The first test vectro V1 initializes the state of transition fault at the first clock cycle (e.g.: for slow-to-rise fault, initialize to 0). The following second test vector V2 will propagate the effect of the transition fault toward the output or observation points. If a transition cannot be observed at the output of any propagation paths, it is considerd to be fault free, otherwise, a transition fault is detected. . There are two kinds of scheme to test a transition fault model, Launch-off-Shift (LoS) and Launch-off-Capture (LoC).

2.3.2.1 Launch-off-Capthure(LoC)

LoC is a test schem for transition fault detecting, in which the first test vector V_1 is applied to the scan chain from an external test generator (ATE or TPG). The inputs of combinational block and first functional clock is used to launch transition in the combination block the scan-enable (SE) signal is de-asserted after V_1. The V_2 is derived by the combinational circuit’s response. The second functional clock would capture the propagated transition at the output, and SE would be asserted. At LoC the scan-in patterns scan shifting at slow speed, then in test mode would shift the test responses to be captured[46]. Figure 2.8-(a) shows LoC

2.3.2.2 Launch-off-Shift (LoS)

In LoS test scheme, test vectors (V_1, V_2) are both scanned to the scan chain. When the first test vector V_1 is loaded into the scan chain, a launch clock will follows the last scan shift clock to initialize the transition fault. Then, the circuit will be switched to the normal

operation by change SE to 0, and immediately followed by a very fast capture clock to capture the fault effect. [46]. Figure 2.8-(b) shows LoS.

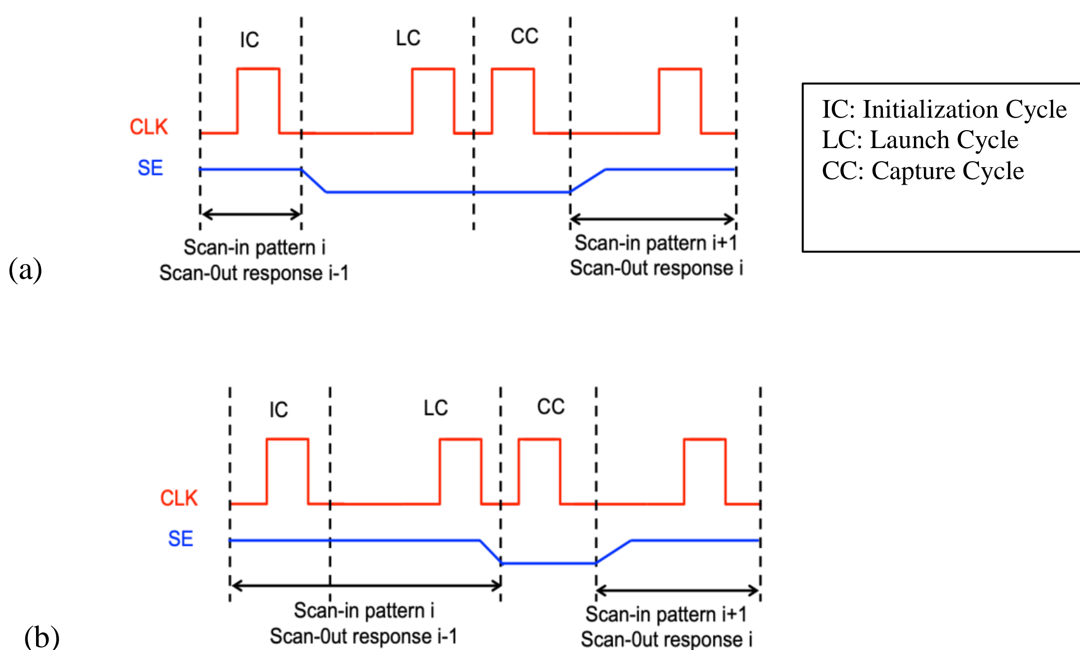


Figure 2.8 (a) Launch-off-Capture (LoC) and (b) Launch-off-shift (LoS) [46]

2.4 Fault Simulation

A fault simulator emulates the digital circuit behave, by simulate the faults in the circuit using fault model then evaluate the test set quality to detect the fault. Then feeding the faults to CUT and compare the test response and expected fault response to determine the faulty circuit [45]. Fault coverage measures the fault detection capabilities for given test set for target fault model which define as:

$$\text{Fault coverage} = \frac{\text{number of detected faults}}{\text{total number of faults}} \tag{2.1}$$

The main goal of Automatic Scan-in pattern Generation (ATPG) is to find a set of scan-in pattern that detect all faults in circuit. The figure 2.9 shows the scheme of fault simulation.

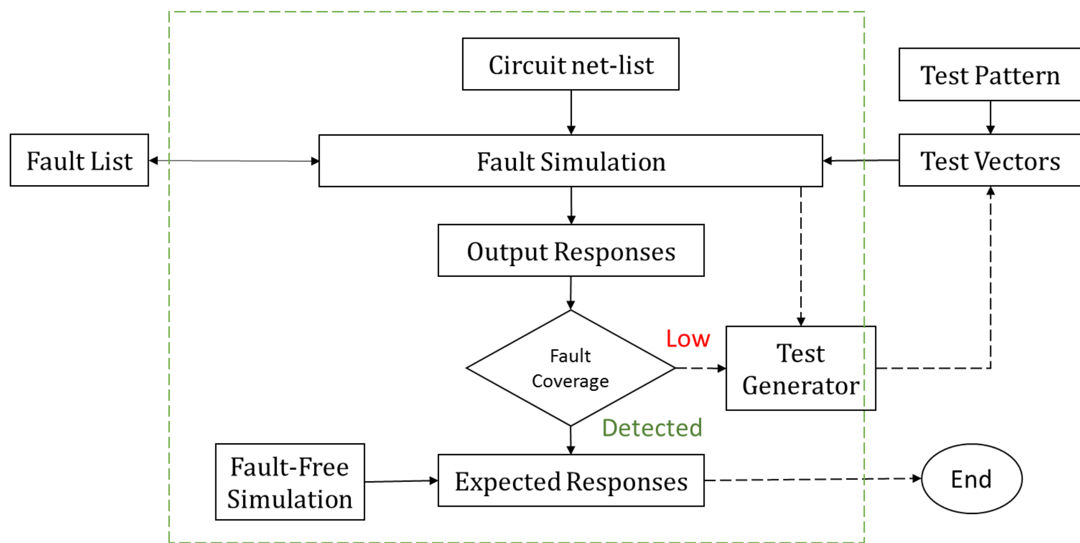


Figure 2.9 Fault Simulation Scheme [45]

Chapter 3

3 Aging induced Reliability Challenges and the solution: Field test

Field-Testing that executes the test when a system is in idle/starting-up state is a promising way to guarantee the reliability of an advanced system. However, the extremely limited test application time obstructs the implementation of field test. Test partitioning and rotating test is an effective way to satisfy such a constraint. In this chapter, we introduce two approaches of a scan-in pattern partitioning. First approach is pattern partitioning for field-testing considering the aging speed. Second approach is feasibility of the machine learning algorithm for the test partitioning.

3.1 Aging Issue

It becomes critical to ensure the reliability for advance systems where shrinking the features sizes and every-growing integration of the deep sub-micron process make the VLSIs vulnerable to the aging phenomenon, such as HCI (Hot Carrier Injection), NBTI (Negative-Bias Temperature Instability) and Time TDDB (Dependent Dielectric Breakdown)[5][6][38]. Aging issue is depending on the application, a system may just degrade, or it may fail from the same amount of aging. For example, a microprocessor degradation may lead to lower performance, necessitating a slowdown, but not necessary failures. In mission-critical applications, such as ADAS (Advanced Driver-Sssistance Systems), a sensor degradation may directly lead to failures and hence system failure.

Figure 3.1 the bathtub curve explains the LSI life cycle and the failure rate is varies with time. The curve is common used in reliability engineering where it divided into three stages: the "infant mortality" failure stage with decreasing failure rate, the "constant/random" failure stage with constant failure rate, and the "wear-out" failure stage with increasing failure rate [41][42]. In the field application. The aging-induced faults can cause serious reliability problems on the circuit when the circuit works for a long time and infringe on

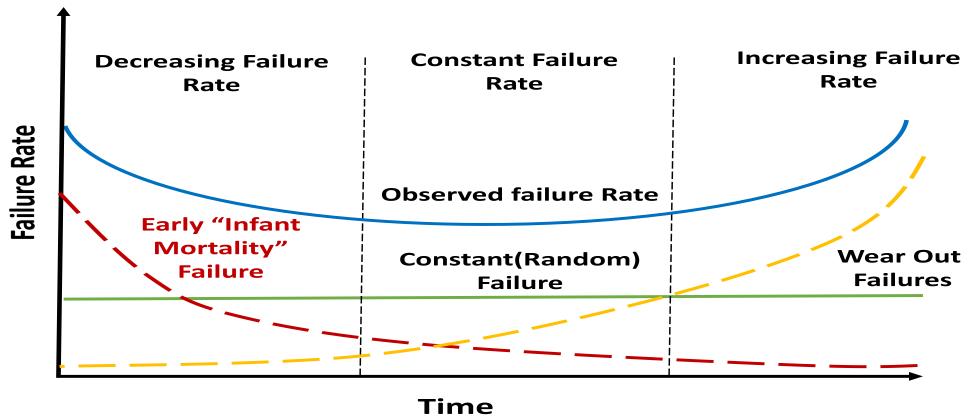


Figure 3.1. Bathtub curve for LSI life cycle in reliability engineering [41] [42]

the functional safety of the system in turn [4]. Field test such as on-line testing and power-on testing is a promising way to ensure the reliability of LSI [7].

3.2 Field test and challenges

Many approaches for field-testing have been proposed in the past, and they can be classified into Concurrent testing and Non-Concurrent Testing.

Concurrent Testing approaches check the real state of the circuit during system operation; it thus can detect the aging faults or soft errors as soon as they appear. However, this approach requires large overhead and causes performance degradation due to the special circuit architecture and redundancy design in terms of hardware, timing or information's [7].

Non-concurrent test such as on-line testing and power-on testing recently grabs much attention and has been applied to some systems that requires high reliability such as autonomous vehicle. Generally, non-concurrent testing is performed during period when the system is in idle, suspended or power-on/ off state that helps to detect the aging-induced faults with less impact to system performance and small hardware overhead. Even for non-stop systems, BIST (Built-In-Self Test) based test architecture is introduced in [8] which performs the testing at test mode periodically in field. Thus, non-concurrent field test like power-on test should be the most promising way to ensure the reliability of circuit.

A big challenge for applying non-concurrent field test is that the test application time must be very short. For example, in power-on for a vehicle control system, the test is executed during the start-up of engine and the time, which is severely limited (e.g.: 10 msec). If the number of test data for a complete testing is large, it is not acceptable to apply all the tests to the circuit within the required test time.

3.3 Power-On Self-Test (POST)

A power-on self-test (POST) is a well-known field-testing technique and has been applied to many systems that demand high reliability [8]. Generally, the POST is executed during the start-up of the system to test the critical components of the system before starting any functional operations, thus the real-time state of the system including the aging-induced faults can be checked/detected in advance to avoid failures. However, POST suffers from a big challenge that the time allowed for testing should be very limited (e.g.: <50ms for an automotive system) [43].

Thus, a test data with large volume is not acceptable to apply within the limited test time during the start-up of system. Moreover, depending on the system, the POST is required to meet a target fault coverage which is usually very high, e.g.: in case of testing an automotive device, at least 90% fault coverage is required to comply with the requirement of ISO26262 standard [2]. In order to enable the POST for a very large scale circuit, many sophisticated test compaction technologies have been investigated deeply in the past such like a test re-seeding technology, TPI (Test Point Insertion) technology [22], enhanced scan testing architecture technology [23], [24].

In such a situation, a partitioning testing is an effective way to enable the application of POST, which divides the original large, test set into several small subsets and applies each subset for testing when the system is in idle or starting-up states. The partitioning testing suffers from the loss of fault coverage of each subset that leads to the increase of detection time interval of the aging-induced faults. To guarantee the reliability, it is necessary to improve the fault coverage for each test subset as far as possible. Therefore, we introduced two strategies. First, to enable the POST and satisfied the limitation of test time we introduced **Test Partitioning Technology.** Second, to improve the fault coverage, we introduced **Fault Detection Enhancement DFT Technology**.

Chapter 4

4 Test Partitioning Technology for POST

4.1 Concept of Test partition

The concept of pattern partitioning and rotating test as shows figure 4.1. The original test set (N_{org}) that has large number of scan-in patterns is divided into several subsets, and then applies one subset at one test session.

In each test session, the number of scan-in patterns N_{set} that are able to apply to the circuit is determined by the limited test application time. Then the number of subsets N_{set} should be N_{org}/ N_{set}.

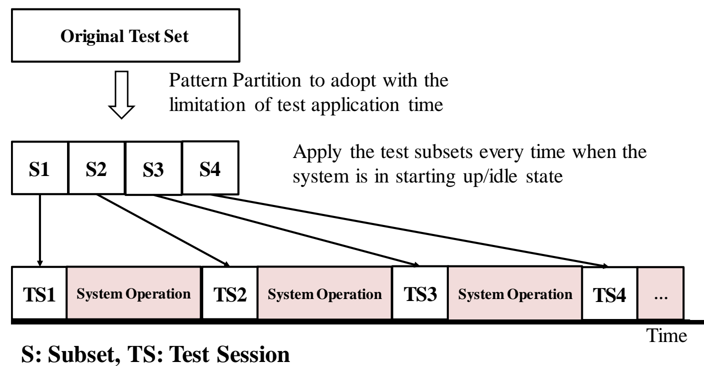


Figure 4.1 Test Partitioning for Field Testing

4.2 The previous works on test partitioning and the problem

In [9], the authors introduced a test partition and rotating test approach to satisfied the constraint of test application time, in which the original test set for complete testing is partitioned into several subsets and apply one subset for the circuit at one test session (when system is in idle or starting-up states). Since each subset consists of small number of patterns, that causes fault coverage loss in each test session. While an effective partition algorithm has also been proposed to maximize the fault coverage for each subset, it is an NP complete problem to achieve high level of fault coverage same as a complete testing. According to the functional safety of electrical /electronic programmable IEC 61508-2[2], it is required to satisfy the requirement of functional safety high fault coverage (>90%),

with test partitioning where it is impossible to achieve that in each subset. Therefore, maximize fault coverage is not enough to avoid the failure.

In [10], the authors discussed the detection interval of faults in test partition, and introduced a test latency named MTFD (Mean Time to Fault Detection), which expresses how soon a fault can be detected, as it appears to evaluate the test partition. In order to minimize the MTFD, they also proposed a GA (Genetic Algorithm) based on test partition approach. In [10], the MTFD shows significant decrease by GA partition method, however the detection interval of the faults, which are detected by small number of scan-in patterns, cannot be reduced. It is very likely to cause a failure because all of the faults only can be detected at one test session and will always be missed at the other test sessions until the detection patterns are applied. For example, if we partition an original test set $T_{org} = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ into 3 subsets $T_{set} = \{T_1, T_2, T_3\}$, suppose a fault f_1 can only be detected by t_1. The faults f_1 will always be missed at two test sessions and has the longest detection latency. Therefore, it is important to find a test partition so that the average detection interval of all faults can be reduced as far as possible.

Moreover, it is known that the aging speed at a transistor is significantly accelerated when the transistor is on state (on/off) during a long time due to the frequent current flow [8]. In a circuit, the gates, which have more switching activities, are most likely to cause aging faults. For high reliability, the faults at such gates need be detected as soon as they appear.

4.3 Approach 1: Pattern Partitioning Considering the Aging Speed

The field test such as the on-line testing and the power-on testing is a promising way to ensure the reliability of LSI. Extremely limited test time makes the application of field-testing difficult. Test partitioning and rotating test is a best to way to satisfy TAT obstructs, however, fault coverage loss caused by the pattern partition leads to the increase of the detection time interval of aging-induced faults. The longer detection time interval is the higher likelihood of a system failure. In this chapter presents a pattern partitioning for field-testing [4] aiming to detect the aging-induced faults.

4.3.1 Aging Speed in the Field

It is known that the aging speed at a transistor is significantly accelerated when the transistor is on state (on/off) during a long time due to the frequent charge/discharge operation [8]. Therefore, an aging-induced fault is most likely to appear at the gate which have more switching activities. In order to achieve high reliability, the faults at the gates whose value toggles frequently during the normal operation need be detected in preference.

4.3.2 Pattern partition method

In this propose method, we select the faults which possibly has high aging speed as the **Risky Faults**, and perform the test partition target on reducing the detection interval of the risky faults. To estimate the aging speed of faults, we proposed a method by calculating the switching activities of each gate. Some definitions are as follows:

- **FEO: fault easy to occur:** It denotes the fault at output of gate which has more switching activities.
- **FHOs: fault hard to occur:** It denotes the faults at the output of the gate which has less switching activities.
- **RF: Risky fault:** A FEO that only be detected by few scan-in patterns (e.g.: one pattern).

The method consists of two phases: an aging fault classification phase and a pattern partition phase. In phase 1 (the aging fault classification phase) we determine the RF by evaluating the switching activities of gates. In phase 2 (the pattern partition phase), we introduce a pattern partition procedure in order to minimize the detection latency for all faults and Risky Faults.

4.3.2.1 Aging Fault classification

Large switching activities at a gate during functional operation can accelerate the aging speed, and the faults at such gate should occur easily. To evaluate the switching activity of gates during the functional operation, as shown in figure 4.2, we expand the circuit to two time-frames and utilize the Probabilistic Testability Measures approach name COP to calculate the *0/1* controllability (*C0 and C1*) of each signal line in each time-frame denoted by $C0_1$, $C1_1$, $C0_2$, $C1_2$, respectively. The calculations of *C0* and *C1* for different gates are shown in table 4.1.

Table 4.1 Probabilistic Controllability Computations

	0/1 controllability (C0 and C1) *a, b: inputs, z: output
AND	C1 (z)= C1 (a)* C1 (b)
NAND	C0 (z)=C1 (a)* C1 (b)
OR	C1 (z)=1- (1-C1 (a))*(1- C1 (b))
NOR	C0 (z)=1- (1-C1 (a))*(1- C1 (b))
NOT	C1 (z)= 1-C1 (a)
BRANCH	C1 (z1)=C1 (z2)…C1 (zn)=C1 (a)

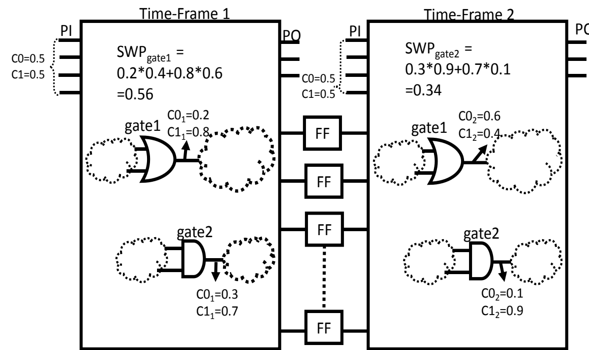


Figure 4.2 Switching probability computations

In order to estimate the switching probability of all gates (include the FFs) for functional operation, only the value of *C0* and *C1* of the primitive inputs (PI) are initialized to 0.5 and 0.5.

We calculate the switching probability (SWP) of the gate value by eq.4.1.

$$\mathrm{SWP} = \mathrm{C0}_1 \times \mathrm{C1}_2 + \mathrm{C1}_1 \times \mathrm{C0}_2 \qquad (4.1)$$

If a gate has high switching probability, the faults at the output of this gate is denoted as FEO, otherwise the faults will be denoted as FHO. For the faults easy to occur (FEO), it is needed to detect them as soon as they appear to ensure the reliability.In other words, the detection latency of FEO should be shortened as far as possible. In partition test, if a FEO can be detected by many scan-in patterns, the detection latency of this FEO can be reduced easily by evenly distributing the patterns to the subsets.

However, if a FEO only can be detected by few scan-in patterns (e.g.: one pattern), the detection latency will always be very large by the conventional pattern partition approaches such as the method in [9] and [10]. In this chapter, we define the faults easy to occur and detected by few patterns as the **risky faults**. Reducing the detection latency for the risky faults should be crucial for ensuring the high reliability.

4.3.2.2 Pattern partition algorithm

According to fault classification, there are faults which have long detection time interval and others have short detection time interval based on that we have to follow two techniques to distributed the scan-in patterns into subsets

4.3.2.3 Evaluation metric of pattern partition

We utilized the mean time to fault detection (MTFD) proposed in GA to evaluate the pattern partition. MTFD is defined as the sum of the time interval between fault occurrence and the fault detection of all the faults divided by the number of faults. It describes how fast the system can react to a fault once it appears. The computation for MTFD is based on three assumptions: 1) no fault occurs during testing application moment; 2) the time for applying the test can be neglected; and 3) the probability of fault occurrence is uniform over time. Figure 4.3 shows the computing model of MTFD for partitioning test, and MTFD can be calculated by eq.4.2 and eq.4.3.

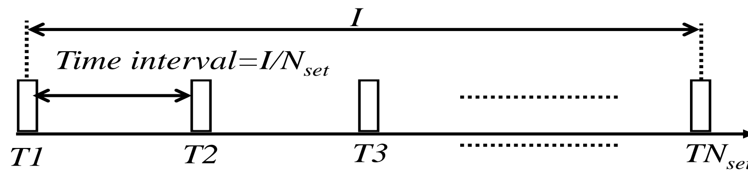


Figure 4.3 MTFD model of partitioning test

$$\mathrm{MTFD} = \frac{I}{N_{set}}\left\{\sum_{j=1}^{N_{set}} \overline{\Delta f_j}\,(2j-1)\right\} \tag{4.2}$$

Where

$$\overline{\Delta f_j} = \frac{I}{N_{set}}\sum_{i=1}^{N_{set}}\left(\begin{array}{c}\text{fraction of faults undetected by}\\ T_{i+1}, T_i, \ldots T_{j-1}, \text{but detected by } T_j\end{array}\right) \tag{4.3}$$

4.3.2.4 Distribute scan-in patterns evenly

In order to reduce the detection time interval for all faults, it is required to distribute scan-in patterns evenly. For this purpose, we do calculate the similarity between the scan-

in patterns to guide the pattern partition. Similarity is defined as: given test set T_{org}, for a couple of scan-in patterns t_i and t_j, if they detected more than same faults, t_i and t_j then they are more similar, the number of these faults is defined as the similarity between t_i and t_j. In order to calculate the Similarity between scan-in patterns we can perform fault simulation. Table 4.2 shows the fault list detected by each scan-in pattern and it gives an example of similarity calculation.

Partition procedure as followings: at first we define the terminology where we suppose partition a test set $T_{org} = \{t_j, t_{j+1}, t_{j+2}... t_n\}$ which detected number of faults f_1~f_k into the subsets $T_{set} = \{T_i, T_{i+1}, ...T_m\}$, N_{org} number of scan-in patterns, N_{set} is the size of each subset. M denoted as the number of subsets, $M = N_{org} / N_{set}$. In the partition procedure, we need to comply following two constraints.

- The similarity between the new pattern and the patterns already exist in the subset must be the smallest.
- The similarity between subset must be the largest.

The partition procedure based on similarity applied, to distribute the scan-in pattern evenly to reduce the detection time interval for faults, which can be detected, by the large number of scan-in patterns. However, the risky faults still have long detection time interval and these faults have been missed in many subsets. Table 4.3 clarifies the result of the first phase of distributing scan-in patterns based on similarity. We denoted the faults, which undetected in subset as (1), and detected faults denoted as (0). The faults f_1, f_5, f_7, f_8 and f_9 are missed at many subsets, if the faults are risky faults, it will be easy to cause a failure. For that, the second phase, which is repeating scan-in patterns that detected the risky faults, is targeted to reduce the detection time interval for all faults. To partition test set T_{org} into several subsets must be applied the coming steps.

1) For T_{org} perform fault simulation with fault dropping after N detection in order of scan-in patterns and for each fault record the ID of the first N patterns which detected the fault.

2) Every scan-in pattern pair $t_i \in N_{set}$ and $t_j \in T_{org}$ where $(i \neq j)$ to count the number of faults which are detected by two scan-in patterns as shows in table 4.3. Distribute

a scan-in pattern into a subset must follow the similarity constraints which are as followings:

- **In the case of where the similarity between subsets is large**.

We selected the scan-in pattern N_{set} that with largest similarity between them from table 4.3. Then distribute them into different subsets. If $M > N_{set}$ then scan-in pattern is exist and calculate the sum of similarity for t_i and t_j. Then distribute the smallest value of N_{set} into a subset.

Table 4.2 Patterns detected faults

	f1	f2	f3	f4	f5	f6	f7	f8	f9
t1	1	1		1					
t2			1			1			
t3		1		1	1	1			
t4				1			1		
t5						1		1	
t6				1					1

Table 4.3 Similarity of scan-in patterns

	t1	t2	t3	t4	t5	t6
t1	-	0	2	1	0	1
t2	0	-	2	0	1	0
t3	2	1	-	1	1	1
t4	1	0	1	-	0	1
t5	0	1	1	0	-	0
t6	1	0	1	1	0	-

- **In the case of where the similarity between t_j and t_i is small.**

We find out the pattern t_j that has the smallest value of similarity and the pattern t_i that has already existed in the subset. In case if there is more than one pattern is

qualified, calculate the sum of similarity for pending patterns and the pattern t_i, which has already existed in other subsets. Then we distribute the patterns, which have the largest sum value of similarity to the subsets.

3) Perform the step 2 until all given test in T_{org} are distributed.

4.3.2.5 Repetation of the scan-in patterns that detect the risky faults

We repeatedly distribute the patterns, which can detect more Risky Faults to different subsets in order to reduce the detection time interval for the risky faults.

The procedures steps as followings;

a) First, find out the scan-in pattern that detected largest number of risky faults from subset T_i, if found then mark as t_x.

b) According to the pervious phase, when the similarity between the subsets is large, it means the faults which are detected in T_i are mostly have been detected in subset T_{i+2} except risky faults. Therefore, from the subset T_i we find out the scan-in pattern that detected the fewest number of additional faults comparing with subset T_{i+1} then mark as t_y.

c) Repeat to assign t_x into subset T_{i+2} and extract t_y from T_{i+2} into subset *(T_{new})*. T_{new} is a new subset we have added as a template for the extracted scan-in patterns.

d) We moved to the next subset T_{i+1}, and then perform procedures in the steps a, b and c until the subset size of *(T_{new}) > (N/M)*.

e) The metric of Mean Time to Detect the Fault (MTFD) used to evaluate the partition procedure. If the detection time interval has been reduced for all faults then again goes through the repeating procedure, otherwise stop the procedure.

4.3.3 Experimental results

To demonstrate the effectiveness of the proposed test partition method, we implemented it using C language and performed experiments for ISCAS'89 circuits. In the experiments, we used scan-in patterns generated for single stuck-at faults by an in-house ATPG program. We set the maximum number of patterns of each subset denoted by N_{set} to 10 and 20. Number of scan-in patterns distributed to each subset will never exceed N_{set}. For determining the risky faults, the faults located at the gate whose switching probability is larger than the 1.5 times of the average switching probability of all gates will be selected as the risky faults. Table 4.4 shows the results of Switching Probability of faults and the number of risky faults for ISCAS'89 circuits. In order to verify the effectiveness of the proposed approach in improving the detection latency for rotating test, we performed experiments for random partition as a comparison.

In the first experiment, N_{set} is set to 10 and the results are given in Table 4.5, the first column gives the name of the circuit, followed by the number of scan-in patterns of the original test set. Column headed by "N_{set}" gives the number of subsets. The columns denoted by "AveDet", "Ave.Fcov.", "MTFD" and "LB_MTFD" show the results of average detection times of all faults, the average of fault coverage of the sunsets, the MTFD of all faults and the lower bound of MTFD for N_{set} partition, respectively. The

results of MTFD of risky faults are also shown in the table 4.4. For the proposed partition method, the number of subsets N_{set} is larger than #Pattern/N_{test} due to some new subsets are created for reducing the MTFD of risky faults by repeatedly distributing patterns. Note the results of MTFD, compared to random partitioning, the proposed method achieved smaller MTFD for all faults, and the MTFD of risky faults are also reduced. In random partitioning, scan-in patterns of risky faults are not repeat assigned to the subsets, the risky faults have the longest detection time interval that causes the largest value for MTFD which is 0.5. In the proposed method, MTFD of all faults are very close to the lower bound value of MTFD, and even smaller than the lower bound value of MTFD in s13207 and s15850. This is because that some patterns are repeat assigned to different subsets that not only increased the detection time (AveDet) for risky faults but also for the non-risky faults.

In the second experiment, N_{set} is set to 20 and the results are given in Table 4.6 the results also shows the effectiveness of the proposed method compared to random partition as same as the first experiment. However, while the average fault coverage of subsets is increased, the MTFD becomes larger than that of N_{set} =10 shown in table 4.5. It suggests increasing the number of test sessions during a fixed period can reduce the MTFD thus ensure the high reliability if the time cost for applying test can be neglected.

Table 4.4 Probabilistic Controllability computations

Circuits	#Faults	Ave. SWP	Threshold of SWP for risky faults	#risky faults
s5378	4563	0.2880	0.4320	84
s9234	6475	0.2931	0.4396	387
s15850	11336	0.2579	0.3868	567
s38417	31015	0.2873	0.4309	875
AVE	13347	0.2816	0.4223	478

Table 4.5 Experimental results for N_{set}=10.

Circuits	#Pattern	Proposed Partition Method						Random Partitioning					
		N_{set}	AveDet	Ave.Fcov.	MTFD	LB_MTFD	MTFD of risky fault	N_{set}	AveDetInt	Ave.Fcov.	MTFD	LB_MTFD	MTFD of risky fault
s5378	101	11	7.87	71.56	0.123	0.109	0.458	11	7.48	68.004	0.138	0.109	0.5
s9234	100	14	7.76	55.446	0.159	0.15	0.425	5	6.72	48.023	0.199	0.15	0.5
s13207	235	29	17.71	61.082	0.122	0.125	0.482	29	14.46	49.846	0.157	0.125	0.5
s15850	111	12	8.13	67.722	0.136	0.138	0.429	12	7.2	59.981	0.17	0.138	0.5
s38417	235	10	7.01	70.109	0.136	0.123	0.431	10	6.57	65.691	0.153	0.123	0.5
AVE	-	-	9.696	65.184	0.135	0.129	0.445	-	8.485	58.309	0.164	0.129	0.5

Table 4.6 Experimental results for N_{set} =20.

Circuits	#Pattern	Proposed Partition Method						Random Partitioning					
		N_{set}	AveDet	Ave.Fcov.	MTFD	LB_MTFD	MTFD of risky fault	N_{set}	AveDetInt	Ave.Fcov.	MTFD	LB_MTFD	MTFD of risky fault
s5378	101	6	4.98	83.012	0.137	0.134	0.443	6	4.7	78.25	0.16	0.134	0.5
s9234	100	7	4.85	69.277	0.171	0.164	0.408	7	4.32	61.728	0.209	0.164	0.5
s13207	235	14	9.57	68.355	0.128	0.133	0.441	14	8.35	59.619	0.163	0.133	0.5
s15850	111	6	4.7	78.333	0.155	0.161	0.402	6	4.29	71.443	0.191	0.161	0.5
s38417	235	5	4.11	82.198	0.162	0.155	0.413	5	3.93	78.525	0.181	0.155	0.5
AVE	-	-	5.64	76.235	0.151	0.149	0.421	-	5.12	69.913	0.181	0.149	0.5

4.4 Approach 2: Test partitioning utilizing the Machine Learning

4.4.1 The Problem of Pattern Partitioning

To execute the field test, we can partition the large original test set into many small subsets and apply each subset to the test sessions (when system is in starting up/idle state). However, test partitioning suffers from a reliability challenge that refers to the **increase of fault detection latency**.

Due to the missing scan-in patterns of each subset, a fault may not be detected at the following test session right after it occurs. The fault effect would be propagated during a time interval from its occurrence to the detection. While a system failure is not caused necessarily as soon as a fault sensitized, the longer detection time interval would cause the higher probability of a failure. Therefore, shortening the detection time interval for each fault is crucial to guarantee the reliability.

In [9] authors analyzed the mechanism of fault detection latency for pattern partitioning and proposed a metric named MTFD (Mean Time to Fault Detection) to evaluate the fault detection latency, and authors proposed GA based partitioning algorithm to minimize the MTFD for all faults. While GA based partitioning achieved significant MTFD reduction, it is too time-consuming to apply for very large circuit, e.g.: circuit with several million gates.

In this chapter, we employ two machine learning algorithms SA and SVM for pattern partitioning problem to derive the optimal partitioning solutions for minimizing the MTFD

of faults and shortening the runtime of pattern partitioning that will be described in the following sections.

4.4.2 Machine Learning Algorithm

Machine learning (ML) is subset of artificial intelligence and it build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so [44].

Recently, machine-learning technique is gaining increased attentions in data classification problem and optimization problem such as the SA (Simulated Annealing algorithm) and SVM (Support Vector Machine) model are well known. In this chapter, we introduce SA and SVM for test partitioning problem, respectively, aiming for achieving the optimal partitioning to minimize the MTFD. The SA based partitioning focus on exploring a global optimization solution for pattern partitioning based on the behavior of cooling metal. In SVM based partitioning, we utilize the One-Class SVM to classify the scan-in patterns into some groups according to the feature of scan-in patterns (e.g.: the number of detectable faults of each pattern) that each group will comprise the patterns with the same (or similar) feature. Then, disperse the patterns of each group to different subsets to derive the best solution of pattern partitioning with small MTFD.

The major contributions of this chapter are as follows:

1) We successfully employed two machine-learning algorithms SA and SVM to solve the pattern-partitioning problem.
2) Experimental results on ISCAS85 benchmark circuits show that both the SA and SVM based partitioning method can achieve very small MTFD.
3) We suggest that SVM based method can derive the best pattern partitioning solution within very short processing time, which is much more efficient than SA that is a big contribution to the pattern partitioning for very large circuit.

4.4.3 Machine Learning Based Test Partitioning

In this part, we introduce two machine-learning models for test partitioning by using SA and SVM, respectively.

4.4.3.1 SA based Test Partitioning

The simulated annealing is a global solution for a given combinatorial optimization problem based on the behavior of cooling metal, and has been widely apply to solve many NP-hard problems [17]. Figure 4.4 shows the procedure of SA based test partitioning. In the procedure, first we randomly assign N scan-in patterns to each subset and calculate the MTFD of faults. Then we set the initial temperature T_{ini}=100 at the beginning of SA partitioning. In the following SA process, we first exchange one pattern ($t_i \leftrightarrow t_j$) of any two subset S_m and S_n, where $t_i \in S_m$ and $t_j \in S_n$, and evaluate the MTFD of faults after the pattern exchanging. Then, an Acceptance Judgment process is done to check whether or not the current pattern exchanging is acceptable by acceptance probability (ACP), which can be calculated by the following formula.

$$\mathrm{ACP} = \begin{cases} 1 & \Delta\mathrm{MTFD} < 0 \\ \exp\left(-\frac{\Delta\mathrm{MTFD}}{\mathrm{T_k}}\right) & \text{otherwise} \end{cases} \tag{4.4}$$

Here, ΔMTFD denotes the difference between the MTFD of the faults before and after the pattern exchanging. When ΔMTFD is smaller than 0, the ACP equals to 1, which means the pattern exchanging would achieve more MTFD reduction; otherwise, the pattern exchanging would cause MTFD increase. In SA based partitioning, for the pattern exchanging, which caused MTFD increase (ΔMTFD>=0), it denotes that the pattern exchanging would be deduced a bad local partitioning solution but may not be worse.

It is still possible to explore a global optimal solution from a bad local solution. Therefore, we need to check how much worse the local solution is.

In the SA algorithm, we randomly generate a value between 0 and 1 as the random acceptance probability denoted by ACP_{rand}, and compare it with the ACP of the current pattern exchanging. If $ACP > ACP_{rand}$, the pattern exchanging will be accepted, otherwise restore the pattern exchanging. When all patterns in the subsets are exchanged in a temperature condition T_k (start from 100), the temperature will be decreased to the 80% of the previous temperature, and repeat the pattern exchanging process until $T_k < 1$.

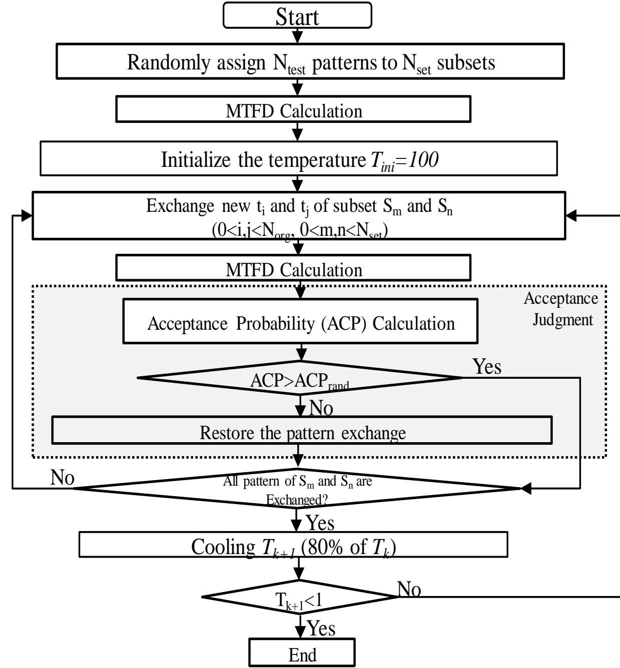


Figure 4.4 Flow chart for Simulated Annealing partitioning method

4.4.3.2 SVM based Test Partitioning

SVMs are well-known machine learning models, which have been widely applied to solve pattern recognition and classification problems because of their flexibility, computational efficiency and capacity to handle high dimensional data [18]. OC-SVM (One-Class Support Vector Machine) is an unsupervised SVM generally used to identify outliers from a given data set. In OC-SVM, the support vector model is trained on data that has only one class. The OC-SVM infers the features of elements from the dataset, and predicts which elements are unlike the other elements of the dataset so as to separate the elements with large different features. In other words, the most similar elements (with more same features) can be classified into the same class (group). In pattern partitioning problem, our target is to distribute the scan-in patterns with the same features to different subsets as even as possible for reducing the MTFD. It is important to distinguish the scan-in patterns with the same features of the original test set at first.

In this chapter, we propose a novel pattern partitioning approach by utilizing the OC-SVM model. Figure 4.5 shows the whole flow of the procedure for 4 subsets partitioning. The pattern partitioning approach consists of two phases: OC-SVM based Pattern

Grouping, and HD (Hamming Distance) based Pattern Distributing. In the Pattern Grouping phase, we utilize the OC-SVM to classify the scan-in patterns of a give test set into two groups based on the following features of pattern.

F1: Total number of faults that are detected by scan-in pattern t_x.

F2: Total number of faults that are only detected by scan-in pattern t_x.

F3: Total number of faults that are not only detected by scan-in pattern t_x.

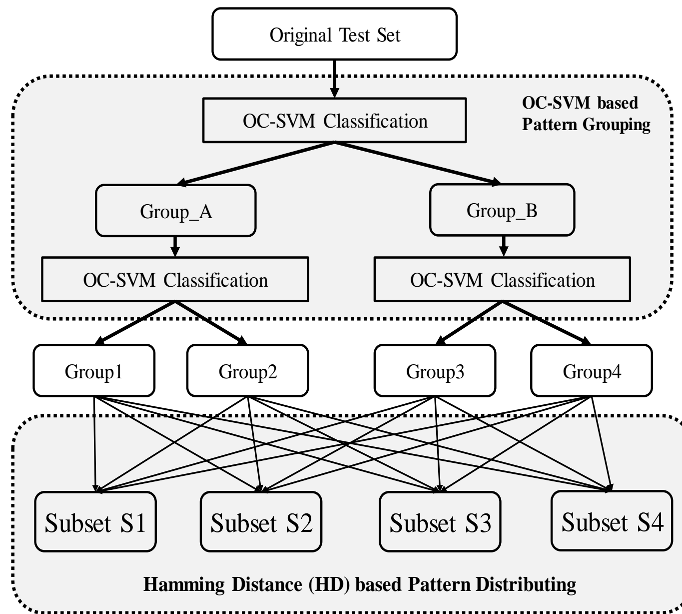


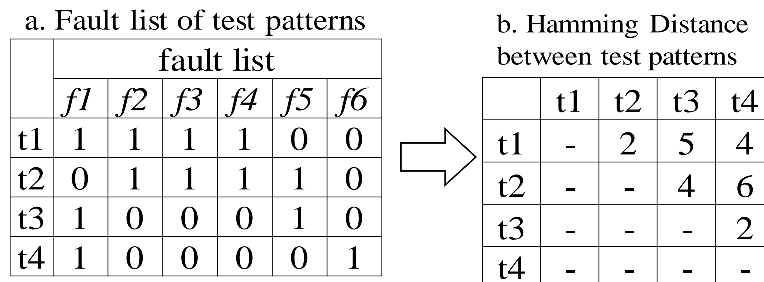
Figure 4.5 Flow chart for SVM based patter partition

For a given test set, every time the OC-SVM will generate two classes (groups), the scan-in patterns with the most similar features (*F_1, F_2 and F_3*) will be classified into the same group. For N_{set} (N_{set}= 4, 8 or 16…etc.) pattern partitioning, Pattern Grouping needs to generate N_{set} groups for the following Pattern Distributing phase; therefore, we repeat the OC-SVM Pattern Grouping process until N_{set} groups are generated. After the Pattern Grouping phase, we can get N_{set} groups of patterns, however these groups cannot be used as the subsets for field testing because each group composes the scan-in patterns with the most similar features, and the features between groups are very different which would cause large MTFD. For example, suppose that if SVM classifies the test set into two groups based on feature 1 (F_1) as given above, the patterns that can detect a large number of faults

will be included in Group1, and the patterns that can detect small number of faults will be classified into Group2. When applying Group1 and Group2 to the test, a large number of faults detected by Group1 would be missed by Group2 that can cause long non-detection interval for such faults. Therefore, in order to minimize the MTFD, we perform the Pattern Distributing follow the OC-SVM Pattern Grouping phase. In Pattern Distributing phase, we focus on dispersing the patterns of the groups to different subsets as even as possible. We utilize HD between scan-in patterns to guide the Pattern Distributing. Table 4.7 gives an example to calculate the HD between scan-in patterns. For the original test set, we perform fault simulation to compute the list of detectable faults for each scan-in pattern, as shown in Table 4.7 (a), if a fault is detectable by a pattern, in the fault list the fault will be labeled with 1, otherwise with 0. The HD measures the difference of fault list between two patterns. As shown in Table 4.7 (b), the fault list of scan-in pattern t_1 and t_2 are [111100] and [011110], and thus there are two bits different, then the HD between t_1 and t_2 is 2.

In Pattern distributing procedure, in order to minimize the MTFD, we assign a pattern to a subset from the groups in accordance with the following constrains.

Table 4.7 Calculation of Hamming Distance



a. Fault list of test patterns

	fault list					
	f1	*f2*	*f3*	*f4*	*f5*	*f6*
t1	1	1	1	1	0	0
t2	0	1	1	1	1	0
t3	1	0	0	0	1	0
t4	1	0	0	0	0	1

b. Hamming Distance between test patterns

	t1	t2	t3	t4
t1	-	2	5	4
t2	-	-	4	6
t3	-	-	-	2
t4	-	-	-	-

C1. The HD between subsets is small

C2. The HD between scan-in patterns in each subset is large

The procedure of Pattern Distributing is as follows.

Step1. Create N_{set} subsets denoted by S_i ($0<i<=N_{set}$).

Step2. For a subset S_i, pick up one pattern t_x ($0<x\leq N_{test}$) from group G_j ($0<j\leq N_{set}$) that has the largest HD between other patterns in G_j, and assign t_x to S_i.

Step3. Move to the next group G_{j+1}, pick up one pattern t_y $(0<y\leq N_{test})$ from group $G_{j+1}(0<j\leq N_{set})$ that has the largest HD between t_y and the patterns already exist in S_i, and assign t_y to S_i.

Step4. Repeat Step3 until N_{set} patterns are assigned to S_i.

Step5. Repeat Step2~Step4 until all subsets are filled up with N_{set} patterns.

4.4.4 Experimental Results

To demonstrate the effectiveness of the two machine learning techniques: SA partitioning and OC-SVM partitioning, we conducted the experiments on Ubuntu 16.04.5 OS using CPU (Intel® CoreTM i7-3770 @3.40GHz) with memory 16GB. We used python 3.6 to implement the partitioning algorithms and conducted the experiments on ISCAS85 benchmark circuits. We used scikit-learn library [19] to implement the OC-SVM based partitioning. For comparison, we also implemented the GA based partitioning algorithm presented in [9], the generation number is set to 1000. For all circuit, we divided the original test set into 4, 8 and 16 subsets, respectively.

Table 4.8 shows the results of MTFD by GA, SA and SVM based pattern-partitioning methods. As for the result, it can be seen that increase the number of subsets can reduce the MTFD of faults which suggests that during a fixed period of system operation, conduct more test operations can shorten the detection interval of faults that contributes to improve the reliability of system. Regarding to the pattern partitioning method, SA and SVM based partitioning method achieved smaller MFTD compared to the GA based partitioning for most circuits. While the MTFD results of SVM based partitioning are larger than SA, the difference is very small which only 0.001 on average, and the processing time (runtime) for pattern partitioning is much shorter than SA as shown in table 4.9.

In table 4.9, we evaluated the total runtime to derive the subsets by GA, SA and SVM based pattern partitioning methods. It can be seen that GA based method requires the most time to generate the final subsets. SA based method significantly shortened the runtime compared to GA. Moreover, the SVM based method shows much more efficient than SA, which can generate the subsets within 1s to achieve almost the same level of MTFD with SA.

Table 4.8 The MTFD of GA, SA and SVM based pattern partitioning

Circuits	# of Patterns	4 subsets partitioning			8 subsets partitioning			16 subsets partitioning
		GA[13]	SA	SVM	GA[13]	SA	SVM	GA[13]
c1355	104	0.179	0.178	0.181	0.132	0.13	0.132	0.113
c7552	382	0.144	0.143	0.142	0.093	0.091	0.09	0.07
cs9234	174	0.211	0.206	0.208	0.162	0.171	0.176	0.162
cs38584	203	0.17	0.169	0.169	0.123	0.122	0.123	0.103
Ave	216	0.176	0.174	0.175	0.127	0.129	0.13	0.112

Table 4.9 the runtime (sec) of GA, SA and SVM based pattern partitioning

	4 subsets partitioning			8 subsets partitioning			16 subsets partitioning		
Circuits	GA[13]	SA	SVM	GA[13]	SA	SVM	GA[13]	SA	SVM
c1355	94.26	18.91	0.04	336.91	29.43	0.03	1271.51	60.03	0.03
c7552	454.55	259.30	1.99	1642.86	289.60	1.07	6249.80	392.60	0.66
cs9234	413.89	89.02	0.33	1465.70	121.10	0.18	5298.27	506.50	0.13
cs38584	2095.12	506.50	2.42	7437.12	612.80	1.33	27909.10	1072.00	0.86
Ave	764.46	218.43	1.19	2720.65	263.23	0.65	10182.17	507.78	0.42

4.5 Conclusions

In this chapter, we introduced two approaches to implement field-testing, first approach we presented a test partition method that takes the aging speed of faults into account. First, find out the risky faults, which are most likely to occur considering the aging speed though the evaluation of the switching probability of gates. In the test partition procedure, at first distribute the patterns into the subsets evenly and then perform the pattern replacement to repeatedly distribute the patterns, which can detect more risky faults to different subsets for reducing the MTFD of the risky faults. The experimental results of ISCAS89 bench circuits confirmed the effectiveness of the proposed test partition method.

The second approach we presented Machine learning technologies SA and SVM into test partitioning problem. Experimental results on the benchmark circuit show that both the SA and SVM based partitioning achieved smaller MTFD than the GA based partitioning. SVM based partitioning can generate the subsets with large MTFD improvement within the only 1s runtime which is much more efficient than other methods. To enhance the reliability, we improve the quality of test patterns that applied in POST, and that we introduced **fault detection enhancement technology** that presented in chapter 5 and chapter 6.

Chapter 5

5 Fault Detection Enhancement DFT for POST

From this chapter we are targeting improve the test quality of test data that are applied in POST. We introduced the multi-cycle test to the Logic-BIST scheme for reducing the volume of the root test data required for achieving high fault coverage. The multi-cycle test allows the test responses of the CUT to be reused as test stimuli for testing that could detect additional faults before the following root test data is applied. However, we raise two major issues that obstruct the effect of multi-cycle test for shorting the test application time (TAT) of POST, which are:

1. Fault effects vanishing (FEV) problem.
2. Fault Detection Degradation of capture patterns (FDD).

To address the FEV problem, we proposed a DFT approach named Fault-Detection-Strengthened (FDS) method that presented in chapter 5. To overcome the FDD problem, we proposed a DFT method named FF-Control Point Insertion (FF-CPI) technique that presented in chapter 6.

5.1 Backgroud of POST

Recently, Power-on Self-Test (POST) is gaining increased attention in the automotive industry for ensuring the functional safety of the advanced automotive system in the field. Generally, the POST is executed by performing the Logic-BIST (Logic Built-in self-test) application during the start-up of the engine to test the automotive devices before starting any functional operations, thus, latent faults (multiple faults would violate the safety goal whose presence is not detected by a safety mechanism) in the devices can be detected at the early stage so as to avoid failures and guarantee the functional safety of the system.

5.1.1 Functional Safety Standard: ISO26262

Functional safety of electrical /electronic programmable IEC 61508-2, it is required to achieve high fault coverage. **Automotive Safety Integrity Level (ASIL)** is a risk classification scheme defined by the ISO 26262 (The most stringent automotive safety integrity level (ASIL D) requires at least 90% of latent fault metric to avoid a random hardware failure due to permanent, intermittent or transient faults [2].

5.1.2 Previous DFT techniques for POST

To test an automotive device, the POST is required to meet several constraints including: 1) The specified fault coverage (stuck-at fault), 2) The limited test application time, and 3) The low power consumption. 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 [21]. 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 [22], scan-shift clock reusing [23], TMS (Tri-Modal Scan) test [24] and capture-per-cycle hybrid-TPI [25]. 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) [26-28]. 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 the 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 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 [29]. 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.

5.2 Multi-cycle Testing

Multi-cycle test is a promising way to achieve high fault coverage with less scan-in pattern. Following discuss the benefit and the problem of Multi-cycle test in detail.

5.2.1 The Benefit of Multi-Cycle Test

In this part, we describe the benefit of the multi-cycle test, we point out an issue of fault effects vanishing in the multi-cycle test. Then we describe a sequential observation technique to handle the issue of the fault effects vanishing for multi-cycle test.

5.2.1.1 Multi-cycle test for fault coverage improvement

Figure 5.1.a shows the clock design of broadside 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 cycle, and only one clock (capture clock) is used for testing. To achieve target fault coverage, LBIST usually requires large number of pseudo-random patterns (scan-in patterns) that cause great amount of test application time.

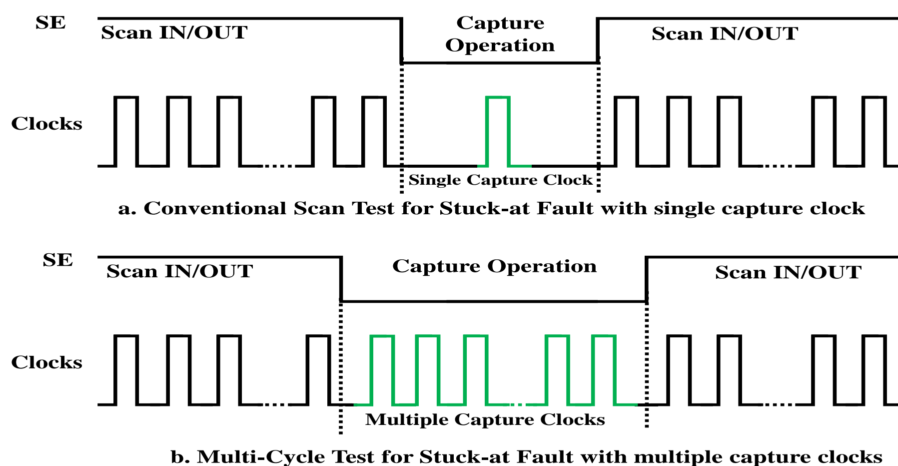


Figure 5.1 Clock Design for Scan Testing

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 Figure 5.1.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 not detected by the root scan-in pattern. It has promising potential to reduce the number of scan-in patterns for achieving 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.

Compared to the conventional scan test, multi-cycle test provides more chances for fault excitation to detect more additional faults, which are not detected by scan-in patterns. Therefore, it has promising potential to achieve high fault coverage with less scan-in pattern. Figure 5.2 shows the stuck-at fault coverage of scan test with single capture clock and multi-cycle test with 10 capture clocks for b13 of ITC99 benchmark.

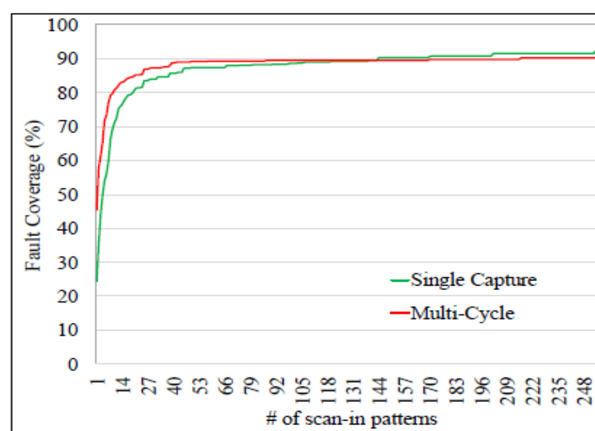


Figure 5.2 Stuck-at Fault Coverage of b13

It shows that multi-cycle test achieved higher fault coverage with a smaller number of scan-in patterns than single capture test. However, the fault coverage of multi-cycle test becomes lower than single capture test by applying more scan-in patterns to the circuit (>200 scan-in patterns). This issue will be discussed in the following.

5.2.2 The problems of Multi-cycle test

There are two major problems obstruct the effect of multi-cycle test to reduce the scan-in patterns, **1) fault effects vanishing and 2) fault detection degradation** of capture patterns. The fault effects vanishing problem 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. The Fault Detection Degradation (FDD) of capture patterns denotes the decrease of capability of capture pattern (the test responses of CUT) to detect more additional faults.

In the following of this chapter, we analyze the mechanism in detail and propose a novel DFT technique named Sequential Observation (SEQ-OB) by FDS-FFs to address the fault effects vanishing problem. In chapter 6, we will describe the fault detection degradation problem and its solution in detail.

5.2.2.1 Fault Effects Vanishing (FEV)

Generally, the response captured at the last capture cycle is observed in the multi-cycle test. A major problem is that a faulty value excited at a middle capture cycle might disappear before it is propagated to the last capture cycle. We call it fault vanishing. Figure5.3 shows the time-frame expansion of a sequential circuit in multi-cycle test, a fault at gate *A* is excited in the first capture cycle, and the faulty value of *A* can be propagated to FF4 and FF5. In the second capture cycle, the faulty value can be propagated to input of the gate *B, C* and *D*, but vanish at their outputs because the output value of gate *B, C* and *D* are dominated by the input values 0. When the number of capture cycles becomes larger, fault vanishing will be more possible that can cause fault coverage loss. This is the reason why the fault coverage of multi-cycle test becomes worse than single capture test shown in Figure 5.2 as increase the number of scan-in patterns.

To address the fault effects vanishing problem, 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 [30-32].

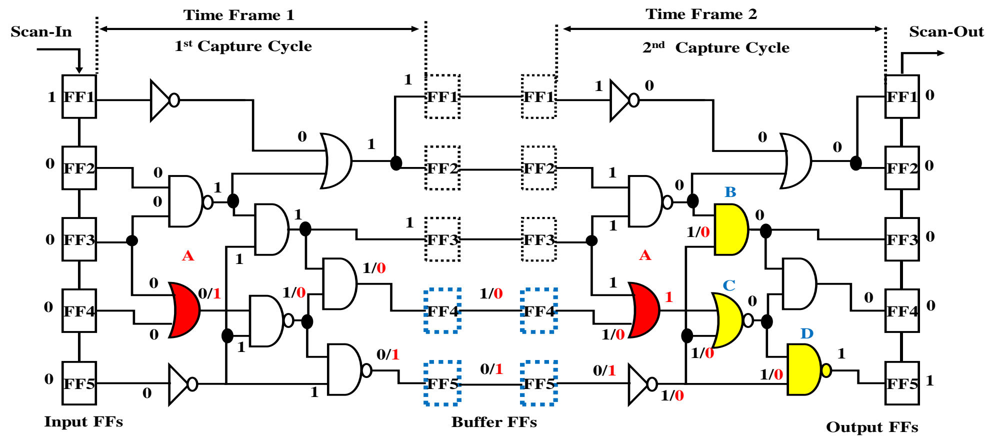


Figure 5.3 Fault effects vanishing in time-frame expansion circuit

5.3 DFT approache to address FEV problem

5.3.1 Sequential Observation for Multi-cycle test

In order to handle the issue of the fault effects vanishing for multi-cycle test, a novel method that directly observe the value of FFs at each capture cycle has been introduced in [13], we call it **Sequential Observation Technique (SOT)**. Figure 5.4 shows the overall structure of the method. In a scan based BIST design with multi-cycle test, scan-in pattern is loaded into the FFs through the scan chain during scan shift operation. Then, the responses of the CUT are captured into FFs at each capture cycle, and scanned-out to the comparator circuit (e.g.: MISR) after the last capture cycle. In order to avoid fault effects vanishing during multiple capture cycles, the outputs of each FF are connected to an additional comparator circuit so that the values of each FF captured at each capture cycle can be directly observed. In this way, the fault effects propagated to FFs and vanished at a capture cycle can be detected. However, observing the value of all FFs requires large hardware overhead (additional comparator) that is not feasible. In [14], the authors improved their scheme to reduce the hardware overhead by observing a small part of FFs. They have reported that the hardware overhead when observe 20% scan FFs only cause 2.0% overhead increase which is far lower than 9.3% of the full observation (observe all FFs). However, the method for selecting the FFs for sequential observing that can achieve maximum fault coverage has not been established yet.

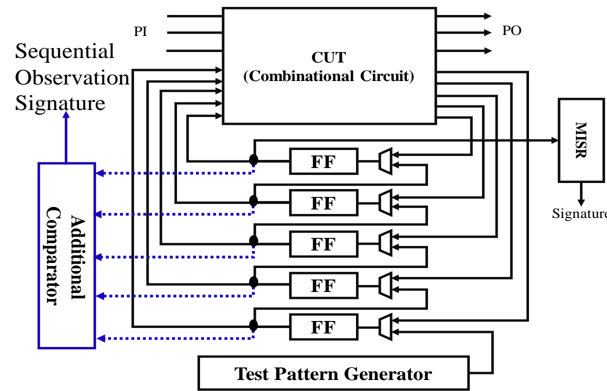


Figure 5.4 Sequential Observation for Multi-Cycle Test

5.3.2 Problems of FF Selection for Observing

The sequential observation is useful to overcome the fault effects vanishing issue for multi-cycle test. However, hardware overhead is needed to perform the sequential observation in the multi-cycle test. Full observation for all FFs is impractical way. Therefore, it is needed to select the FFs that can detect more vanishing faults by sequential observation. We define the terminologies for the proposed method as follows.

- **Definition 1: vanishing fault and fault effects vanishing gate**

The **vanishing fault** is defined as the fault which is excited by a pattern (scan-in pattern or capture pattern) but vanish at some gates in the following capture cycles.The gate where the fault vanishes is called **fault effects vanishing gate**.

- **Definition 2: fault effects vanishing point FF**

In multi-cycle test, if a sensitive path between a fault node and its vanishing gate passes through a FF, sequentially observing the FF's value can detect the vanishing fault. We call such FF the **fault effects vanishing point FF**. Figure 5.5 shows an example of the vanishing fault, fault effects vanishing gate and fault effects vanishing point FF, respectively.

- **Definition 3: fault-detection-strengthened FF**

If large number of vanishing faults share the same vanishing point FF, observing the FF is effective to improve the fault coverage. We define such FF as the **fault-detection**

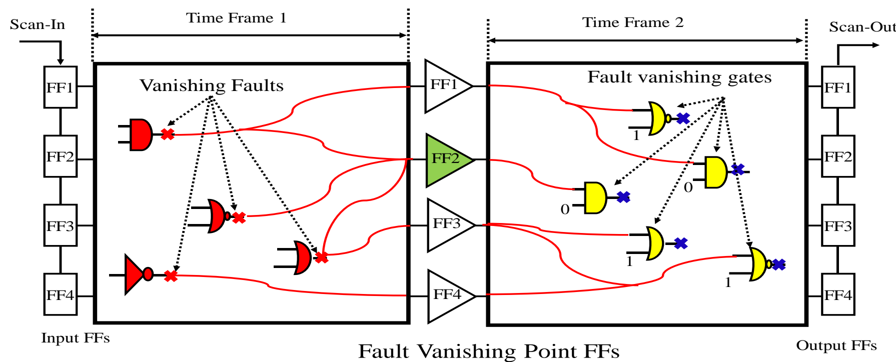


Figure 5.5 Definition of fault effects vanishing point FF

strengthened FF (FDS_FF). For example, in Figure 5.5, FF2 should be an FDS_FF. One problem of FDS_FF selection for sequential observation is how to evaluate the fault effects vanishing point FFs. Furthermore, when apply multi-cycle test with N capture cycles to a CUT, the circuit complexity will become N-times due to time-frame expansion. For very large circuit (e.g.: with several million or more gates), the efficiency (shorter processing time) of FF selection is strongly required.

Performing a full fault simulation (without fault dropping) can calculate all of the vanishing faults and find out the most effective FDS_FFs. However, it is too time-consuming to apply for very large circuit. In [14], the authors proposed a simple algorithm for evaluating fault effects vanishing point FF based on SCOAP testability analysis method. In the method, the observability of FFs in a time-frame expansion circuit are calculated to evaluate the fault-vanishing point FFs, and FFs with higher value of observability (difficult to observe from external output) are selected as FDS_FFs for sequential observation. Experimental results reported in [14] shows the improvement of fault coverage compared to random selection and less CPU time than fault simulation-based selection. However, since large value of observability of a FF denotes that the signal value at the FF is difficult to be propagated/observed at the last capture cycle in multi-cycle test. It cannot be proved that more vanishing faulty values will pass through the FF, whose value is difficult to be observed because the vanishing faults always disappear at vanishing gates but not at the FFs. Therefore, we believe that the fault coverage improvement should be not enough.

5.3.3 Evaluation Methods for Fault effects vanishing Point FF

In this part, we introduce three selection methods of fault detection- strengthened FFs (FDS_FFs) for sequential observation. In the methods, structure-based metrics are proposed to evaluate the fault effects vanishing point FFs.

5.3.3.1 Method 1: Gate-FF Connection Complexity based Evaluation

For a gate g_i in the combinational logic part of a sequential circuit, if its inputs connect with many FFs through input structural paths (from FFs to gate), the fault at g_i should be more possible to be excited by scan-in pattern (scan-in or capture pattern). Moreover, if the output of g_i connects with many FFs through output structural paths (from gate to FFs), the excited faulty value at g_i should have more chances to be propagated to FFs. In addition, if large number of gates exist on the input/output structural paths of g_i, the faulty value at g_i is most likely to vanish. We call such connection information between gates and FFs the "Gate-FF Connection Complexity" and define six parameters to evaluate it.

Definition: P1~P6 for each gate g_i

- P1. The number of FFs connected with the input(s) of g_i by structural input paths
- P2. The number of FFs connected with the output of g_i by structural output paths.
- P3. The total number of gates on the structural input paths of g_i.
- P4. The total number of gates on the structural output paths of g_i.
- P5. The length (gate levels) of the longest structural input paths of g_i.
- P6. The length (gate levels) of the longest structural output paths of g_i.

Figure 5.6 shows an example to calculate the value of P1~P6. For each FF, we calculate the sum value of P1~P6 of all gates which can be observed by the FF through back-tracing as **the Connection Complexity (CC)** metric for evaluating fault effects vanishing point FF. The calculation formula is given as follow.

$$\mathrm{CC(FF)} = \sum_{i=1}^{N} \sum_{j=1}^{6} \mathrm{P_j} \tag{5.1}$$

Where, N denotes the number of all gates observed by a FF. We believe that sequentially observing the FFs with higher connection complexity should be more effective for vanishing faults detection.

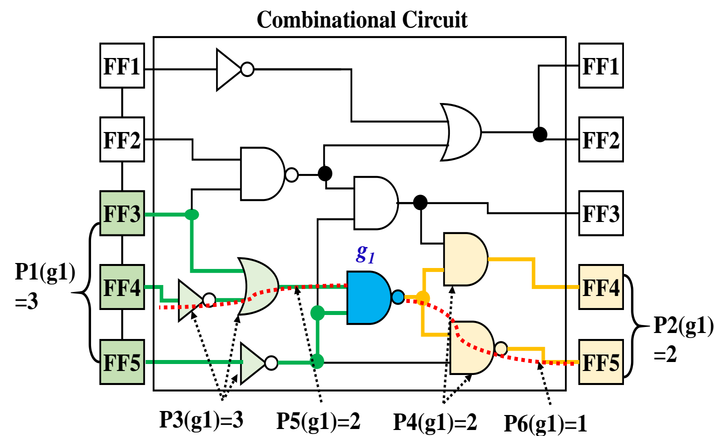


Figure 5.6 an example to calculate the Gate-FF Connection Complexity

5.3.3.2 Method 2: Structural propagation path of fault in sequential circuit

If a vanishing fault excited in one capture cycle and its value can be propagated to a FF, it can be detected by sequentially observing FFs. Therefore, evaluating the structural propagation paths between gates and FFs in a sequential circuit should be effective to improve the fault coverage. Figure 5.7 shows a timeframe expansion circuit for 2-capture cycles test. In the figure 5.7, FFs are modeled as buffers in the middle capture cycles. We consider three types with different constraints, and calculate the number of gates that satisfy the constraints as metrics to evaluate the fault effects vanishing point FFs. Three types are shown as follows.

- **Type 1: For each FF, calculate the number of gates that satisfy the following 2 constraints (denoted by C) such as gate g1 shown in Figure 5.7.**

  C1. The gates only connect with FF_i by structural output paths.

  C2. The gates connect with FFi by structural input paths.

- **Type 2: For each FF, calculate the number of gates that satisfy the following 3 constraints such as gate g2 shown in Figure 5.7.**

  C1. The gates only connect with FF_i by structural output paths.

  C2. The gates do not connect with FF_i by structural input paths.

  C3. Structural paths exist between the input and output of FF_i.

❖ **Type 3: For each FF, calculate the number of gates that satisfy the following 3 Constraints such as gate g3 shown in Figure 5.7.**

C1. The gates only connect with FF_i by structural output paths.

C2. The gates do not connect with FF_i by structural input paths.

C3. Structural paths do not exist between the input and output of FF_i.

For all types, we believe that sequentially observing the FFs with larger number of gates that satisfy the constraints should be effective to detect vanishing faults.

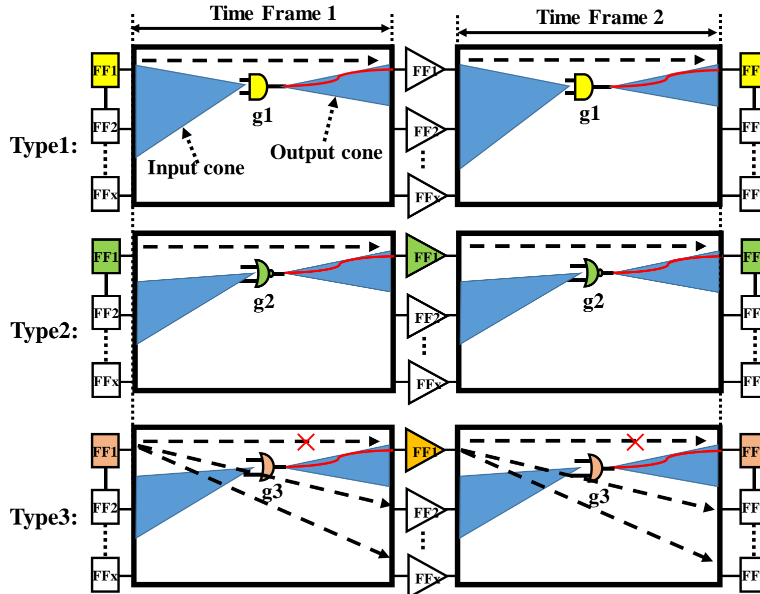


Figure 5.7 Structural connections between gates and FFs

5.3.3.3 Method 3: Branch Reachable Rate based evaluation

Generally, more branches in a circuit can create more structural propagation paths for faulty values. For a FF, if large number of branch signal lines exist in its input cone (the logic region observed by FF as shown in Figure 5.8 and can reach to the F through structural paths, observing the FF is most likely to detect more faults. Therefore, we use the Branch Reachable Rate of FF as a metric to evaluate the fault effects vanishing point FF that is defined as the ratio of the branches that can reach to the FF through structural paths. An example is shown in figure 5.8. The Branch Reachable Rate of FF can be computed by the following formula.

$$BR(FF) = \frac{\#\text{ of reachable branches of FF}}{\#\text{ of branches exist in the observable logic parts of FF}} \tag{5.2}$$

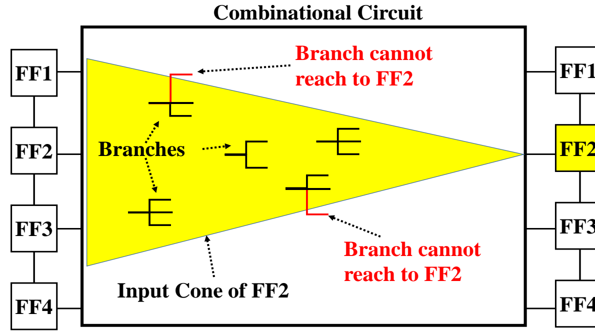


Figure 5.8. Branch Reachable Rate

Sequentially observing the FFs with higher Branch Reachable Rate might contribute to vanishing faults detection.

5.3.4 FDS-FF Selection for Sequential Observation

In this part, we propose the method for selecting the fault detection- strengthened FFs (FDS_FFs) for the multi-cycle test with the sequential observation. Selecting the FDS_FFs for sequential observation using the evaluation methods individually might be effective to the fault coverage improvement of multi-cycle test for the circuits with special (corresponding) structure. We believe that comprehensively evaluating the metrics of vanishing fault detection capability derived by the evaluation methods could achieve a better solution of FDS_FFs selection for general circuits. In [15], Multi-Criteria Decision Analysis method named TOPSIS [16] has been introduced to find out an optimal solution of FFs selection for low power problem of BIST. We employ the TOPSIS algorithm on our proposed structure-based evaluation methods of fault effects vanishing point FFs to select the most effective FDS_FFs for improving the fault coverage of multi-cycle test with sequential observation. The procedure for selecting FDS_FFs based on the TOPSIS is model as follows.

Procedure: Selecting FDS_FFs based on TOPSIS

Step1. Create an evaluation matrix consisting of M alternatives (number of FFs) and N criteria with the intersection of each alternative and criteria given as t_{ij}, we therefore have a matrix:

$$T = (t_{ij})_{M\times N},\quad (i = 1,2,\cdots,M,\ j = 1,2,\cdots,N) \tag{5.3}$$

Step2. Normalize matrix T using following formula.

$$R=(r_{ij})_{M\times N},\ r_{ij}=t_{ij}\Big/\sqrt{\sum_{i}^{M}t_{ij}^{2}} \tag{5.4}$$

Step3. Calculate the weighted normalized decision matrix v_{ij} by (5.5)

$$v_{ij}=w_j r_{ij},\ \sum_{j}^{N}w_j=1 \tag{5.5}$$

Step4. Determine the worst alternative (v_j -: minimum value of each criteria) and the best alternative (v_j+: 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 (5.6).

$$S_i^+=\sqrt{\sum_{j=1}^{N}(v_{ij}-v_j^+)^2},\ S_i^-=\sqrt{\sum_{j=1}^{N}(v_{ij}-v_j^-)^2} \tag{5.6}$$

Step5. Calculate the similarity to the worst condition (C_i) for each alternative by (5.7):

$$C_i=\frac{S_i^-}{S_i^+ + S_i^-} \tag{5.7}$$

Select the FFs with large C_i as FDS_FFs for sequential observation.

5.4 Experimental Results

We evaluated the proposed fault-detection-strengthened FFs (FDS_FFs) selection methods for sequential observation using ISCAS89 and ITC99 benchmark circuits. A 16-bits internal type LFSR (characteristic polynomial: $X_{16}+X_{15}+X_{13}+X_4+1$) is used to generate 10k pseudo-random scan-in patterns. A parallel scan structure with 100 FF-length of scan-chains is adopted (when number of FFs > 1600, 200 FF-length). A multi-cycle BIST with 10 capture clocks is applied to the circuit, and only 20% of FFs are selected as FDS_FFs, and their values will be sequentially observed during multi-cycle test due to the area overhead concern.

Conventional scan test with single capture clock (SCAN), 10 capture clocks test without sequential observation (MULTI_CAP) and with full observation (FULL: observing all FFs), and the SCOAP based selection method are performed for comparison. Table 5.1 shows the results of fault coverage and pattern reduction by sequential observation.

For each circuit, the results of final fault coverage achieved by 10k patterns are shown in the upper row, and the number of patterns to achieve the largest fault coverage is shown in the lower row. From the results of fault coverage, it can be seen that multi-cycle test (MULTI-CAP) achieved significant increase of fault coverage for most of the circuits, but decrease for s13207 and s15850, compared to the conventional scan test. This is because more faults vanished during the multi-cycle test as described in section 5.2. Sequentially observing all FFs can achieve the most fault coverage improvement, which can be used as upper bound of fault coverage increase to evaluate the effect of the proposed FDS_FFs selection methods. When observing 20% FFs, SCOAP based selection method only achieved small increase of fault coverage for ISCAS89 circuits and b20 compared to multi-cycle test.

In the three FDS_FFs selection methods proposed, the fault coverage of all circuits could be improved by at least an individual method, and the results are much closer to (or even same with) the upper bound of fault coverage achieved by full observation.

The TOPSIS based selection method comprehensively evaluates the three methods (same weight values: 0.2), for all circuits it therefore can achieve the most fault coverage improvement. Note the results of the number of scan-in patterns (null means 10k patterns cannot achieve the fault coverage of SCAN); our proposed FF selection methods also achieve more pattern reduction compared to the multi-cycle test and the SCOAP method.

Table 5.1 Fault coverage improvement and pattern reduction by sequential observation

circuit	#FF	#Chain/ Max.Length	#Faults	SCAN	MULTI_ CAP	Multi-cycle test with sequential observation							
						FULL	SCOAP	Method1	Method2			Method3	TOPSIS
									Type1	Type2	Type3		
s13207	669	7/96	9815	86.78	80.98	89.49	88.01	88.96	87.09	87.07	88.85	88.60	88.24
				10000	null	4650	6427	5263	8524	8524	5263	5709	6051
s15850	597	6/100	11725	86.88	85.91	88.40	87.75	87.69	87.31	87.71	86.97	86.73	87.61
				9908	null	4481	5622	6588	7580	6424	9441	null	6994
s38417	1636	9/182	31180	91.93	94.67	95.72	94.75	95.32	95.31	95.38	95.26	94.68	95.33
				9947	3095	1297	2551	1497	2005	1946	1610	3095	1472
b13	53	1/53	857	94.63	93.00	94.63	93.00	94.63	93.00	94.63	94.63	94.63	94.63
				2613	null	1137	null	2316	null	null	2316	1137	2316
b14	245	3/82	12811	84.35	86.89	86.93	86.89	86.89	86.92	86.89	86.89	86.92	86.92
				9678	2011	2011	2011	2011	2011	2011	2011	2011	2011
b15	449	5/90	23528	69.60	91.10	91.16	91.10	91.11	91.10	91.14	91.13	91.14	91.13
				6761	158	158	158	158	158	158	158	158	158
b21	490	5/98	26580	85.45	89.00	89.13	89.01	89.12	89.12	89.12	89.00	89.12	89.12
				9956	1056	652	1012	693	699	693	939	699	699
AVE	-	-	-	85.66	88.79	90.78	90.07	90.53	89.98	90.28	90.39	90.26	90.42
				8409	-	2055	-	2647	-	-	3105	-	2814

5.5 Case Study

We conducted a case study on an electronic control unit (ECU) circuit provided by Renesas System Design Corp. using Test tools to verify the feasible of the proposed FDS_FFs selection methods for sequential observation. The information of the ECU circuit is detailed in Table 5.2. A special scan cell for sequential observation that has small area is developed by Renesas and has been applied to the DFT of the ECU circuit.

In order to evaluate the effect of multi-cycle test on fault coverage improvement, we applied 10k random pattern to the ECU circuit, and performed testing with 1~10 capture clocks, respective.

Table 5.2 Information ECU circuit

# of PIs	*2364*
# of POs	*2600*
# of Gates	*251796*
# of FFs	*13159*
# of Scan Chain	*134*
Max. Chain Length	*100*
# of Stuck-at Faults	*1419234*

Figure 5.9 shows the results of fault coverage. It can be seen that the conventional scan test with single capture clock achieves 87.7% fault coverage using 10k random patterns that cannot satisfy the requirement (>90%) for the functional safety. Applying more capture cycles increases the fault coverage, and reduce the number of scan-in patterns for achieving 90% fault coverage. For example, only 6144 random patterns (scan-in pattern)

are needed when applying 10 capture clocks. To evaluate the efficiency of the proposed FF selection methods, 20% of FFs are selected by the proposed methods and their values are sequentially observed during the multi-cycle test. Figure 5.10 shows the fault coverage results for 10 capture cycles test.

Compared to the results of 10 capture cycles test without sequential observation (Non_Observation), all the proposed FDS_FFs selection methods can further improve the fault coverage. Method 1 and Type 2 of Method 2 achieved the largest increase in fault coverage (>95%), and Type 1 and Type 3 of method 2 show some fault coverage improvement with SCOAP method. While Method 3 achieved higher fault coverage for ITC99 bench circuits, it shows lower fault coverage for ECU circuit compared to SCOA method.

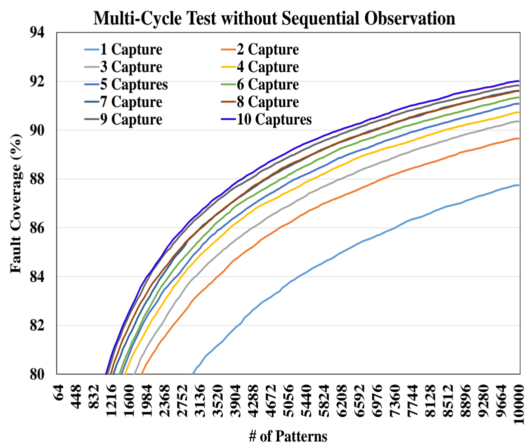


Figure 5.9 Stuck-at fault coverage in multi-cycle test

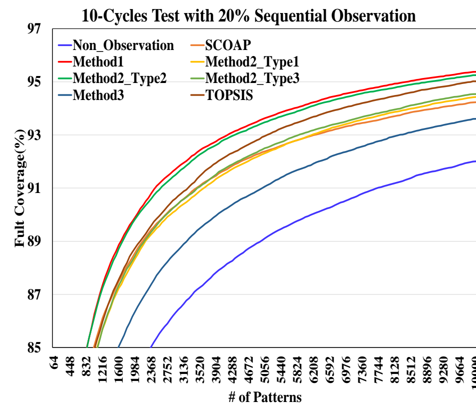


Figure 5.10 Stuck-at fault coverage of 10-cycle test by sequential observation

TOPSIS based FDS_FFs selection method comprehensively evaluates the metrics of Method 1~3, and achieves larger fault coverage than SCOAP but lower than Method 1 and Type 2 of method 2. This is because of the impact of Method 3, which might be less effective to improve fault coverage for very large circuit. However, it is worth noting that TOPSIS based selection method is effective to achieve higher fault coverage for an unknown circuit especially for a very large circuit. For pattern reduction, the number of random patterns required to achieve 90% fault coverage by the proposed FF selection methods are shown in Table 5.3.

Table 5.3 Pattern reduction for testing the ECU circuit

	Multi-Cycle (10 Capture)	SCOAP	Method1	Type1	Type2	Type3	Method3	TOPSIS
# of Patterns	6144	2688	1984	2816	2048	2688	3904	2560
Compression Rate (X)	-	2.3	3.1	2.2	3	2.3	1.6	2.4

It can be seen that the number of random patterns for 90% fault coverage in 10-cycle test can be reduced to a half or a third of 6144 patterns by sequentially observing 20% FFs. Method 1, Type1 of Method 2 and TOPSIS show higher pattern compression rate (the number of patterns for non-observation/the number of patterns for sequential observation) than SCOAP method which are 3.1X, 3.0X and 2.4X, respectively. Method 1 achieved the largest compression rate (3.1X).

5.6 Conclusions

In order to handle the fault effects vanishing issue in multi-cycle test, we have proposed three methods to select the FDS_FFs for sequential observation by evaluating the structure of circuits. Experimental results of ITC99 benchmark circuits shows that the proposed methods are much more effective to the most of circuits in fault coverage improvement and random pattern reduction than SCOAP based method (the existing method). We also performed a case study on a real ECU circuit, which consists of 250k gates and 10k FFs. The results show that selecting FDS_FFs for sequential observation by evaluating the "Gate-FF Connection Complexity" (Method1) and the structural propagation path of fault in sequential circuit (Method2) can significantly improve the fault coverage (>95%) and can reduce the number of scan-in patterns (e.g.: 2.4X~3.1X compression) to achieve 90% fault coverage. We proclaim that the multi-cycle test with the sequential observation for the selected FFs can achieve 90% fault coverage that satisfies the requirement of functional safety by using small number of random scan-in patterns. To overcome of fault Detection Degradation (FDD) of capture patterns issue, we proposed a DFT method named FF-Control Point Insertion (FF-CPI) technique that we introduced in chapter 6.

Chapter 6

6 DFT method to address FDD Problem

As discussed in chapter 5, we raised two major issues that obstruct the effect of multi-cycle test to reduce the scan-in test pattern for shorting the test application time (TAT) of POST, which are fault effects vanishing (FEV) problem and fault detection degradation of capture patterns (FDD). In this chapter, we focus on the Fault detection degradation problem of multi-cycle test, we first analyze the mechanism of FDD problem in detail, then, we will introduce a DFT method we proposed named FF-Control Point Insertion (FF-CPI) technique by modifying the captured values of scan Flip-Flops (FFs) during capture operation [35][47] to overcome the FDD problem. Also, we proposed the methods to evaluate the FFs for determining the candidate FFs for FF-CPI that can achieve more fault detection, by analyzing the circuit structure w/o any simulations for the purpose of shortening the development period of DFT.

The main contributions of this chapter 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 without any simulation that can shorter 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 pattern reduction with smaller hardware overhead.

6.1 Analysis of FDD problem

The Fault Detection Degradation (FDD) of capture patterns denotes the decrease of capability of capture pattern (the test responses of CUT) to detect more additional 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 captures cycles that is helpful to low 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 numbers of capture cycles are applied. Since the value of FFs (capture pattern) are reused as scan-in patterns at the subsequent capture cycles, large number of FFs with constant values would cause the loss of randomness property of the capture pattern. We believe that this behavior obstructs to detect more additional stuck-at faults by using the capture pattern. To verify such assumption, we design a preliminary experiment as follows.

We defined a $TpC_{i,j}$ (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 after a capture cycle (j) during a complete test. For a complete test with k scan-in patterns, the $TpC_{i,j}$ can be calculated by following formula.

$$\mathrm{TpC_{i,j}} = \sum_{k=1}^{\text{Number of Test}} v_{i,j-1,k} \oplus v_{i,j,k} \tag{6.1}$$

Where, $V_{i,j,k}$ denotes the value of FF_i at the j capture cycle when the k pattern is applied. Figure 6.1 shows an example to calculate the TpC for FF1 under 4 capture cycles test with 3 patterns. In the example, $V_{1,0,k}$ denotes the initial state of FF1 of the kth test which comes from the scan-in pattern, $V_{1,j,k}$ denotes the state of FF1 at the j capture cycle in the k test which are the responses of CUT after the j capture cycle For each test, we compare the state of FF1 at each capture cycle with its state at the previous cycle to detect a transition of FF1 at the cycles When k ($k=3$) tests are applied, we take the sum of transition at the same capture cycle as the TpC of FF1. The value of TpC in the example expresses that the state of FF1 changed in 3 tests at the first capture cycle, 2 tests at the second capture cycle, and 1 test at the third and the fourth capture cycle during the complete test. Evaluation of TpC of FFs at each capture cycle First, we execute a logic simulation with 50 patterns under 10 capture cycles test to calculate the TpC for each FF. Figure 6.2 shows the TpC of each FF at 10 capture cycles. The vertical axis denotes the total TpC during 50

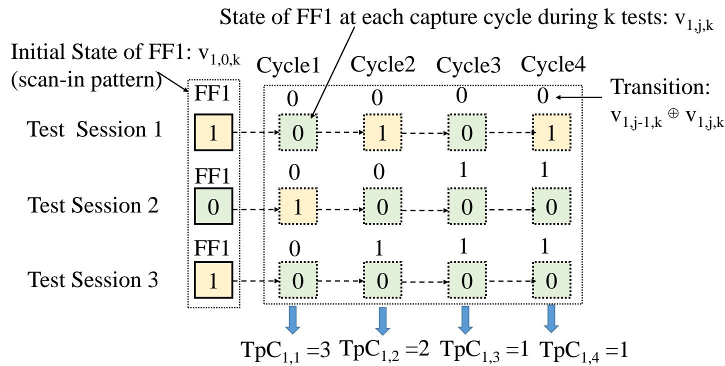


Figure 6.1 Example to calculate the TpC for a FF

tests, the horizontal axis denotes the capture cycle number, and the lines denote the TpC of each FF at the corresponding capture cycle. It can be observed that the number of tests in which the state transition occurred in each FF decreases as applying more capture cycles. For FF4, FF6, FF7, FF8, FF9, and FF12, state transitions occur at the first capture cycle in many tests (16~27 tests), then their states become constant value in most tests after the second capture cycle.

The observation in Figure 6.2 confirmed that the multi-cycle test could cause the state of many FFs become 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.

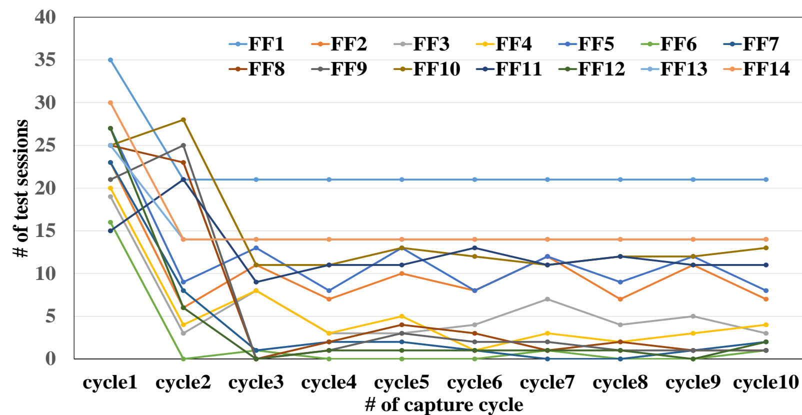


Figure 6.2 TpC of each FF at capture cycles

1) Evaluation the additional fault detection at each capture cycle

The fault-dropping simulation (one-detection drop) with 10 captures 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 6.3 shows the result, where the line denotes the average TpC of all FFs at each capture cycle given in Figure 6.2 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 show 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.

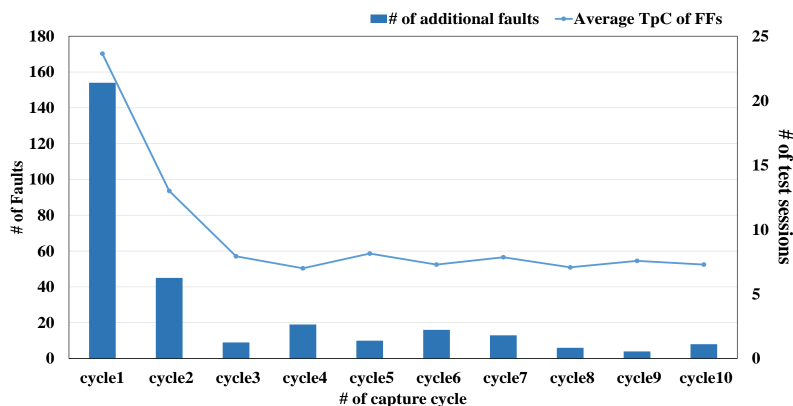


Figure 6.3 Average TpC of FFs & Number of additional faults detected at capture cycles

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.

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 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.

6.2 FF-Control Point Insertion (FF-CPI) for Multi-cycle test

The basic idea of FF-CPI is to overcome the Fault Detection Degradation problem (FDD problem) to enhance the stuck-at fault detection and pattern reduction for multi-cycle test [35].

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 to improve the randomness property is modifying the values of some FFs as to avoid constant states during the capture cycles. For illustration, we performed an additional experiment on s298 to verify the effect of the idea 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 multi-cycle test. We check that 1) how much impact modifying the value of 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 Figure 6.2, 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 registers of the status logic in the traffic light controller, which is directly controlled by a primary input (PI). In multiple cycle tests, 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 Figure 6.2 by reversing the captured value of FF6 during captures operation from the third captures cycle. Figure 6.4 shows the TpC of each FF at 10 capture cycles when reversing the capture value of FF6. It can be observed that the number of test sessions in which the state transition occurred in FF8, FF9, FF10, and FF11 are significantly increased. This is because modifying the value of FF6 in force during capture cycles causes more active mode transitions in the status logic, thus the FFs like FF8, FF9, FF10, and FF11 that are related to the status logic would have significant value changes. This observation provides a perspective that modifying the value of a small number of FFs would break the constant states for large number of the other FFs.

2) The impact on the additional fault detection when modifying the value of FF

We perform the experiment with fault-dropping simulation on s298 circuit by reversing the value of FF6 at the capture cycles from the third capture cycle. As well as the experimental setup of Figure 6.3, any fault that is detected at a certain capture cycle 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. The scan-in patterns and the order of the scan-in patterns are same with that of Figure 6.3.

In Figure 6.5, 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 number of additional stuck-at faults detected at the corresponding capture cycle and the dotted line shows the average TpC 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 table 6.1 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 6.1 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 target fault coverage. As the results shown in Table 6.1, 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.

Table 6.1 the number of the additional faults detected at capture cycles (308 stuck-at faults of the combinational circuit of s298)

Scan-in Pattern	FF6 Reversing	Capture cycle										Sum of faults	Accumulated fault coverage (%)
		1	2	3	4	5	6	7	8	9	10		
SP1	NO	48	13	0	0	0	0	0	0	0	0	61	19.8
	YES	48	13	17	2	0	0	0	0	0	0	80	26.0
SP2	NO	12	1	0	0	0	0	0	0	0	0	13	24.0
	YES	11	0	0	0	0	0	0	0	0	0	11	29.5
SP3	NO	46	11	6	8	5	8	4	2	2	3	95	35.1
	YES	33	11	8	8	13	8	10	2	9	3	105	37.7

In the experiment of Figure 6.5, 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.

Figure 6.5 shows the effect on additional stuck-at faults detection by reversing the capture value of FF6. We add the average TpC of all FFs at each capture and the number of additional stuck-at faults detected at each capture cycle given by the dotted line and the dotted columns, respectively. It can be seen that reversing the value of FF6 during captures increased the average number of tests in which the state transition occurred at FFs. Consequently, the number of additional stuck-at faults detected at the capture cycles (cycle 3, 5, 8 and 9) start from the third capture are increased. It should be noted that the decrease of the additional stuck-at faults detection at the first and the second capture cycle is caused by the improvement of the stuck-at fault detection capability of capture patterns after the second capture cycle.

More additional stuck-at faults would be detected in advance at the capture cycles before new scan-in pattern is applied; it is therefore effective to reduce the number of scan-in patterns to achieve the target fault coverage for the stuck-at faults. In this experiment, the advantage of the proposed method is that the normal 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.

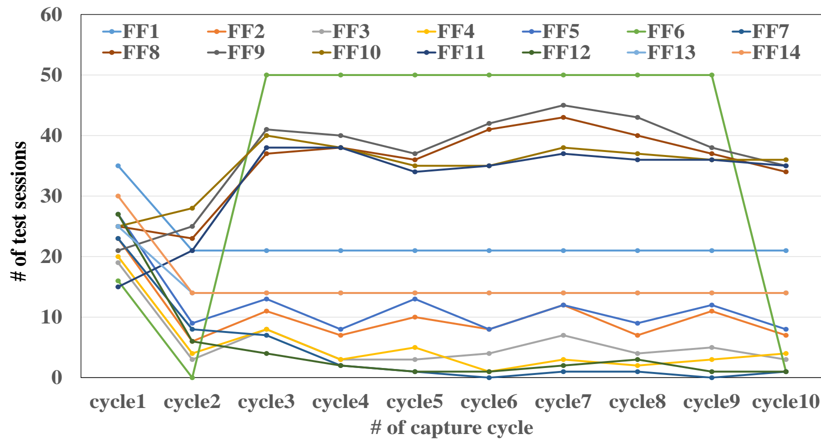


Figure 6.4 TpC of each FF at capture cycles by reversing capture value of FF6

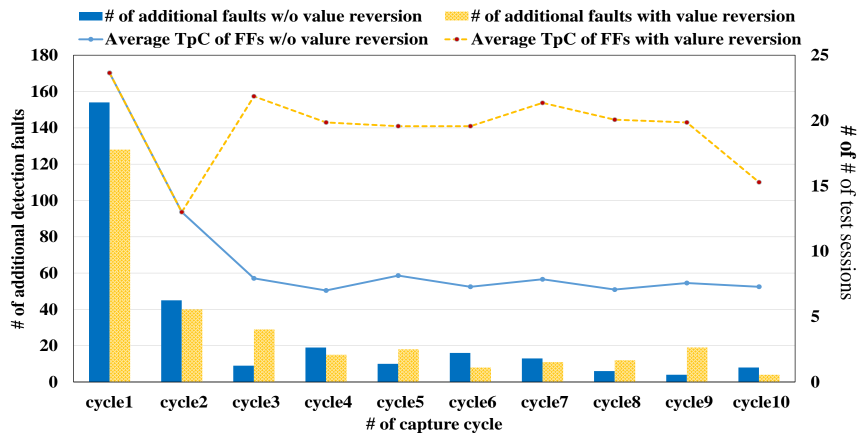


Figure 6.5 Average TpC of FFs & Number of additional faults detected at capture cycles by reversing capture value of FF6

6.3 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 [35], 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 descripted as follows.

6.3.1 FF-Reversing CPI

One FF-CP circuit 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 6.6 shows the basic design concept of FF-Reversing control. In the capture mode, the present-state (Ti: scan-in pattern applied at the current capture cycle) and the next-state (Ri: 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 Ri to the CUT as Ti+1, otherwise apply the current capture response Ri to the CUT for the next capture cycle.

Figure 6.7 shows the structure of the method (details are masked due to the patent concern).

6.3.2 Random-Load

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 scan-in 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.

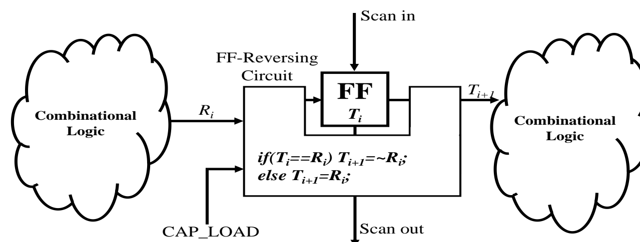


Figure 6.6 FF-Reversing captures pattern control [35]

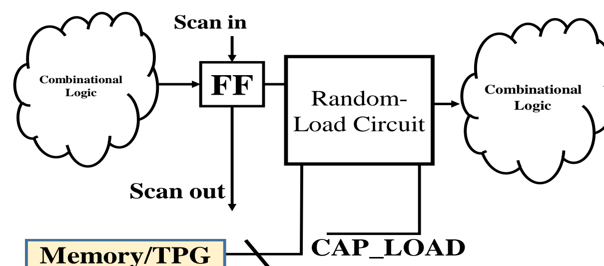


Figure 6.7 Random-Load for captures pattern control [35]

6.4 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 impact modifying the value of FFs would have on the internal states of CUT. When the large number of FFs become to have constant values during capture cycles, the state of the gates and signal lines would also be fixed at each capture cycle. This situation 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 (fault-effects) and contribute to detecting more additional faults.

It should be noted that the TpC metric described previously is not suitable to evaluate the FFs for the control point selection because it can only represent the amount that the value of FF becomes constant during multi-cycle test, but the contribution to fault detection by FF-CPI.

In [35], we have proposed two methods to evaluate the FFs for FF-CPI, they will be explained in this chapter as follows. Main difference between this proposed method and the method in [35], 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.

6.4.1 Method 1: Transition Probability Increment (TrPI)

The main idea is to select the FFs as the candidates for FF-CPI that 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. Figure 6.8 shows the calculation of transition probability of gates in multi-cycle test. We transform the CUT to N-cycle time-frame expansion combinational circuits, and initialize the 0/1 controllability (C0 and 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

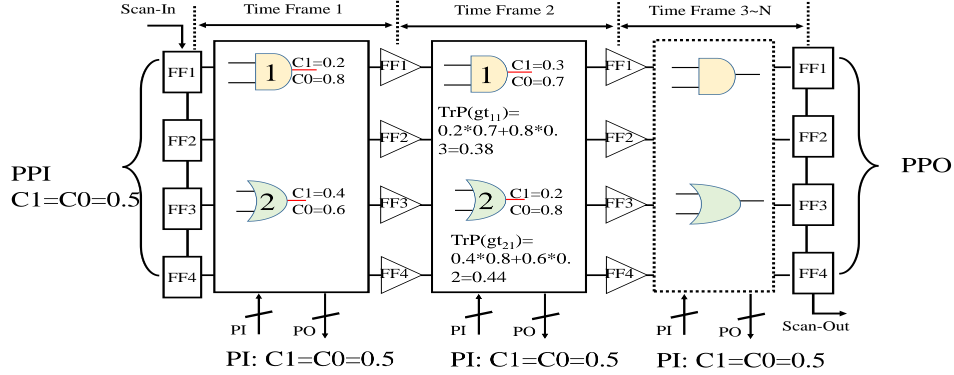


Figure 6.8 computing the state transition probability by COP [35]

gate at each time-frame. The transition probability (TrP) of a gate gt_{ij} ($i<$ number of gates, $j< N$ cycles) can be calculated by the eq.6.2.

$$\mathrm{TrP}(gt_{i,j})=C0_j\times C1_{j+1}+C1_j\times C0_{j+1} \tag{6.2}$$

The average transition probability of all gates during N-cycle test (AVE_TrP) can be calculated by the eq.6.3.

$$\mathrm{AVE_TrP}=\frac{1}{M}\sum\sum \mathrm{TrP}\left(gt_{i,j}\right) \tag{6.3}$$

Where, $1 \le i \le$ number of gates, $0 \le j \le M$.

The Transition Probability Increment (TrPI) is defined as the difference of the average transition probability of all gates before and after FF-CP (FF-Reversing or Random- Load) is inserted. When supposes FF-CP is inserted into a candidate FF, the 0/1 controllability (C0/C1) of the output signal line of the FF is 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:

Step1. Calculate the original average transition probability of all gates during N-cycles test, which is denoted by $\mathrm{AVE_TrP}_{org}$.

Step2. 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 N-cycles denoted by $\mathrm{AVE_TrP}_{FFn}$.

Step3. Calculate the difference between AVE_TrPorg and AVE_TrP_{FFn}, which is denoted by $TrPI_{FFn}$.

Step4. Rank the FFs by the value of TrPI of each FF.

The FF, which has large TrPI, will be selected for FF-CPI.

6.4.2 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 modify 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 ($n=1$ 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 6.9 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.

6.4.3 Method 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 chapter, we propose the

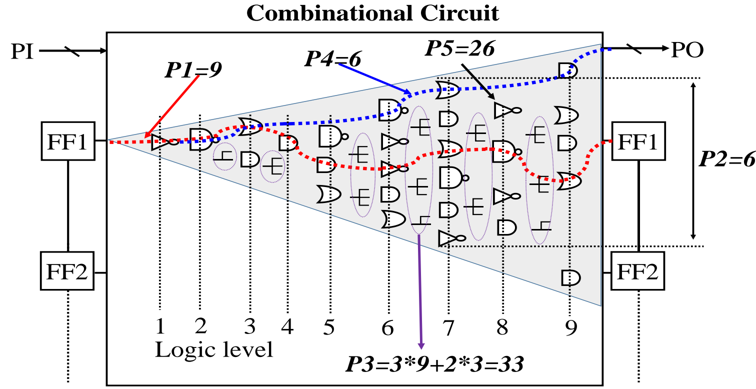


Figure 6.9 Evaluation for LIMA [35]

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 [36] 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.

Step1. Create an evaluation matrix consisting of U alternatives (# of FF_s) and O criteria (# of evaluation metrics of TrPI and P_1~P_5 of LIMA) with the intersection of each alternative and criteria given as t_{ij}, we therefore have a matrix:

$$T = (t_{ij})_{U\times O},\quad (i = 1, 2, \cdots, U,\ j = 1, 2, \cdots, O) \tag{6.4}$$

Step2. Normalize matrix T by following formula.

$$R = (r_{ij})_{U\times O},\quad r_{ij} = t_{ij} \Big/ \sqrt{\sum_{i}^{U} t_{ij}^{2}} \tag{6.5}$$

Step3. Calculate the weighted normalized decision matrix vij by (6.6).

$$v_{ij} = w_j r_{ij},\ \sum_{j}^{o} w_j = 1) \tag{6.6}$$

Step4. Determine the worst alternative (v_j^-: minimum value of each criteria) and the best alternative (v_j^+: 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 (6.7).

$$S_i^+ = \sqrt{\sum_{j=1}^{o}(v_{ij} - v_j^+)^2},\ S_i^- = \sqrt{\sum_{j=1}^{o}(v_{ij} - v_j^-)^2} \quad (6.7)$$

Step5. Calculate the similarity to the worst condition (C_i) for each alternative by (6.8)

$$C_i = \frac{S_i^-}{S_i^+ + S_i^-} \quad (6.8)$$

Step6. Rank the FFs by C_i for FF-CPI.

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

6.5 Evaluation Experiments

6.5.1 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 scan-in 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).

The major purpose of our research 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 >=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 [30] that directly observes the capture values of FFs during the multi-cycle test are conducted to handle the fault effects vanishing problem. In [35], 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 chapter, we conduct the partial observation technique presented in our previous research [30-32] that only 20% of FFs of each circuit are selected

for SEQ-OB to reduce the hardware overhead. The FFs are used for SEQ-OB are selected by the FF selection algorithm presented in [30].

For comparison, the traditional scan test with single capture clock (SCAN), 10 capture clocks test without 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.

6.5.2 Scan-in pattern Reduction by FF-CPI

To evaluate the 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 6.10 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 patterns. When multi-cycle test with 10 capture cycles (MUL) is applied, the fault coverage becomes worse and the number of patterns to achieve 90% fault coverage tends to increase (one million patterns just get 85.56% fault coverage) due to the fault effects vanishing problem.

SEQ-OB technique is effective to handle the fault effects vanishing problem. As the results shown in table 6.2, SEQ-OB by observing all FFs (FULL) and partial observation (P-OB20) significantly improved the fault coverage for multi-cycle test, those only 11,600 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). For easy to read, we just show the fault coverage results of the FF selection method using LIMA metric in Table 6.1 (detailed results of 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 patterns and 6,350 patterns by FF-Reversing, and 1,900 patterns and 2,650 patterns by Random-Load with FULL observation and partial observation, respectively.

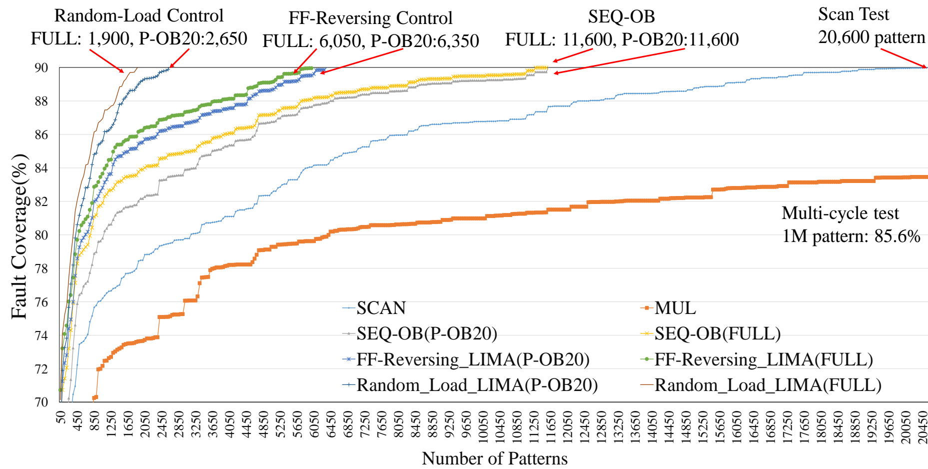


Figure 6.10 Fault coverage curve of s13207

In Table 6.2, we give the detailed pattern reduction results for all circuits. In order to achieve 90% stuck-at fault coverage specified by ISO26262, we perform the experiments by increasing the number of patterns until 90% stuck-at fault coverage is achieved or one million patterns are used. The number of patterns required to achieve 90% stuck-at fault coverage by traditional scan test, 10-cycles capture test without SEQ-OB (MUL) are given in the second and the third columns. For each circuit, if one million patterns cannot achieve 90% stuck-at fault coverage, the final fault coverage achieved by one million patterns are given following with 1M. In the group column denoted by “FULL”, the sub-columns show the number of patterns to achieve 90% stuck-at fault coverage and the pattern compression rate compared with scan test, by full observation of SEQ-OB without 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 without any DFT techniques. Multi-cycle test is a promising DFT technique to reduce the scan-in patterns for scan test. Regarding to the results of multi-cycle test, in s38417, b14, b15 and b20, the multi-cycle test shows significant pattern reduction compared to scan test. However, fault effects vanishing problem caused the fault coverage loss and expanded the number of 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 better pattern reduction with larger pattern compression rate than pure multi-cycle test.

The fault degradation problem is another factor to obstruct the pattern reduction by multi-cycle test. Compared to the results of SEQ-OB, FF-CPI technique shows further 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 pattern reduction (10.5X for s9234, 1.8X for s13207), FF-CPI technique achieved more 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 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 shown the effect to further improve the pattern reduction, which are at most 19.2X, 129.0X, 241.0X and 118.3X by FF-Reversing or Random-Load, respectively.

Regarding to the capture pattern control methods referred to FF-Reversing and Random-Load, we show the results of table 6.2 under the FULL observation. From the experimental results, we 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 loads a random vector to the candidate FFs every cycle, which can directly improve randomness for the capture patterns during capture operations. Random-Load requires additional memory or on-chip scan-in 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 pattern reduction than Random-Load control, it still achieved a significant 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 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 to the FF selection methods referred to TrPI, LIMA and HEM for FF-CPI proposed in 6.1.2, the effects of TrPI and LIMA for pattern reduction might be depend on the circuits used for testing. In s9234 and s13207, LIMA achieved more 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 6.2, for almost all circuits, HEM can achieve better pattern reduction compared to SEQ-OB. For b20 circuit, HEM achieved 118.3X pattern reduction by Random-Load control, which is 2.6X and 86.2X by TrPI and LIMA. However, for b15 circuit, HEM caused significant decrease of 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.

Table 6.2 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	FF-CPI by FF-Reversing			FF-CPI by Random-Load			w/o	FF-CPI by FF-Reversing			FF-CPI by Random-Load		
			FF-CPI	**TrPI**	**LIMA**	**HEM**	**TrPI**	**LIMA**	**HEM**	FF-CPI	**TrPI**	**LIMA**	**HEM**	**TrPI**	**LIMA**	**HEM**
s9234	>1M 88.60%	>1M 85.90%	94,900 10.5X	72,400 **13.8X**	58,800 **17.0X**	68,000 **14.7X**	26,750 **37.4X**	47,900 **20.9X**	53,150 **18.8X**	99,000 10.1X	78,500 **12.7X**	64,150 **15.6X**	70,600 **14.2X**	26,750 **37.4X**	51,000 **19.6X**	54,000 **18.5X**
s13207	20,600	>1M 85.60%	11,600 1.8X	11,600 1.8X	6,050 **3.4X**	5,750 **3.6X**	11,100 **1.9X**	1,900 **10.8X**	2,150 **9.6X**	11,600 1.8X	20,750 1.0X	6,350 **3.2X**	13,400 **1.5X**	12,950 1.6X	2,650 **7.8X**	2,800 **7.4X**
s38417	5,750	1,750 3.3X	600 9.6X	300 **19.2X**	400 **14.4X**	350 **16.4X**	300 **19.2X**	450 **12.8X**	300 **19.2X**	700 8.2X	350 **16.4X**	400 **14.4X**	400 **14.4X**	450 **12.8X**	550 **10.5X**	400 **14.4X**
b14	>1M 85.60%	58300 17.2X	52300 19.1X	92600 10.8X	>1M 88.90%	>1M 89.60%	54300 18.4X	7750 **129.0X**	10700 **93.5X**	56250 17.8X	92600 10.8X	>1M 88.90%	>1M 89.60%	56250 17.8X	8350 **119.8X**	12050 **83.0X**
b15	>1M 69.80%	4,200 238.1X	4,200 238.1X	4,150 **241.0X**	4,900 204.1X	179,350 5.6X	4,150 **241.0X**	4,550 219.8X	9,200 108.7X	4,200 238.1X	4,200 **238.1X**	5,850 170.9X	230,250 4.3X	4,200 **238.1X**	4,700 212.8X	10,350 96.6X
b20	1M 85.40%	237,150 4.2X	165,900 6.0X	375,000 2.7X	315,950 3.2X	352,350 2.8X	390,850 2.6X	11,600 **86.2X**	8,450 **118.3X**	176,650 5.7X	375,300 2.7X	315,950 3.2X	352,350 2.8X	390,900 2.6X	12,100 **82.6X**	9,350 **107.0X**

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 6.2, 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 pattern reduction of FF-CPI with partial observation decreased a little, however, which are still very remarkable achievement compared to the SEQ-OB without FF-CPI.

6.5.3 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 6.1.2, we have proposed three methods TrPI, LIMA and HEM to select the candidate FFs for FF-CPI. Table 6.3 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 N-cycles (here N=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.

Table 6.3 Runtime for FF Ranking

Circuit	# of gates	# of FFs	TrPI(sec)	LIMA(sec)	HEM(sec)	
					Ranking	Total
s9234	5866	228	108.56	0.48	0.089	109.129
s13207	8772	669	470.15	0.9	0.097	471.147
s38417	23949	1636	3299.36	22.34	0.232	3321.932
b15	8893	449	818.15	20.94	0.139	839.229
b20	9419	490	1073.97	46.85	0.123	1120.943
AVE	-	-	1154.038	18.302	0.136	1172.476

6.6 Conclusions

In this chapter, we first reveal the Fault Detection Degradation problem that would obstruct the effect of multi-cycle test for 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 without 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 research [30, 31]. 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.

In addition, the results of partial observation of SEQ-OB confirmed that just observing a small number of FFs could also achieve a significant 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.

Chapter 7

7 Conclusion

Field test is gaining increased attention in the automotive industry for ensuring the functional safety of the advanced automotive system. The POST (Power-On Self-Test) is a novel field test technique. Generally, the POST is executed by performing the Logic-BIST application during the start-up of the engine to test the automotive devices before starting any functional operations, thus, latent faults (multiple faults would violate the safety goal whose presence is not detected by a safety mechanism) in the devices can be detected at the early stage so as to avoid failures and guarantee the functional safety of the system. To test an automotive device, the POST is required to meet several constraints including:

1) The specified fault coverage (stuck-at fault). The most stringent ASIL D (Automotive Safety Integrity Level) of ISO26262 requires at least 90% of latent fault metric to avoid a random hardware failure due to permanent, intermittent or transient faults. Generally, POST only targets on testing the permanent faults, which can be represented by stuck-at fault model.

2) The limited test application time. POST is executed during the engine start-up, 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.

The research has proposed several of approaches to enhance the reliability and function safety in field test. To execute the field test, we can partition the large original test set into many small subsets and apply each subset to the test sessions (when system is in starting up/idle state). We propose the test partition method that takes the aging speed of faults into account. The experimental results of ISCAS89 bench circuits confirmed the effectiveness of the proposed method compared to random partition. However, while the average fault coverage of subsets is increased, the MTFD becomes larger. Therefore, it suggests increasing the number of test sessions during a fixed period can reduce the MTFD.

The other problem is that the test partitioning suffers from a reliability challenge that refers to the **increase of fault detection latency**.

Due to the missing scan-in patterns of each subset, a fault may not be detected at the following test session right after it occurs. Therefore, we propose the approach two machine-learning algorithms SA and SVM to solve the pattern-partitioning problem Experimental results on benchmark circuit show that both the SA and SVM based partitioning achieved smaller MTFD, than the GA based partitioning. And the SVM based partitioning can generate the subsets with large MTFD improvement within only 1s runtime which is much more efficient than other methods.

According, pervious research Multi-cycle test is one of the promising ways to achieve a good trade-off among the fault coverage with few of scan-in patterns, TAT and power consumption for POST. However, 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 order to handle the fault vanishing problem in multi-cycle test, we have proposed three methods to select the FDS_FFs for sequential observation by evaluating the structure of circuits. The results show that selecting FDS_FFs for sequential observation by evaluating the "Gate-FF Connection Complexity" (Method1) and the structural propagation path of fault in sequential circuit (Method2) can significantly improve the fault coverage (>95%) and can reduce the number of scan-in patterns (e.g.: 2.4X~3.1X compression) to achieve 90% fault coverage. The proposed methods are much more effective to the most of circuits in fault coverage improvement and pattern reduction than SCOAP based method (the existing method).

In order to handle the Fault Detection Degradation problem in multi-cycle test, we have proposed the novel solution named FF-CPI technique. We proposed two methods referred to the FF-Reversing Control and the Random-Load Control 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. 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.

Finally, the research has faced challenge such as HEM can achieve better pattern reduction compared to SEQ-OB. For b20 circuit, HEM achieved 118.3X pattern reduction by Random-Load control, which is 2.6X and 86.2X by TrPI and LIMA. However, for b15 circuit, HEM caused significant decrease of 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. A part of our future work is to develop an efficient determination algorithm for the weight values of HEM.

The future work to do experiments on larger ECU circuit (>2M gates) New DFT approaches for ensuring the functional safety. the other future works is to develop a method to minimize the hardware overhead w/o full observation by SEQ-OB.

8 REFERENCES

[1] Laung-Terng Wang, Cheng-Wen Wu , Xiaoqing Wen, "VLSI Test Principles and Architectures: Design for Testability (Systems on Silicon)," *Morgan Kaufmann Publishers Inc*., San Francisco, CA, 2006.

[2] Standard of ISO 26262, part5, Road Vehicles - Functional Safety, May 24th. 2016[Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:26262:-5:ed-1:v1:en

[3] J. Savir and S. Patil, "Scan-based Transition Test," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1993, vol. 12, no. 8, pp. 1232-1241, doi: 10.1109/43.238615.

[4] Hanan T. Al-Awadhi, Senling Wang, Yoshinobu Higami, HiroshiTakahashi,"Pattern Partitioning for Field Testing Considering the Aging Speed," Hiroshima, Japan, 2016, pp.72-76.

[5] H. Puchner, L. Hinh, "NBTI Reliability Analysis for a 90nm CMOS Technology," European Solid-State Device Research Conf., 2014, pp.257-260.

[6] Fen Cheng, M. Shinosky, "Addressing Cu/Low-k Dielectric TDDB-Reliability Challenges for Advanced CMOS Technologies," IEEE Trans. on Electron Devices, 2009, vol. 56, no.1, pp.2-12.

[7] M. Nicolaidis, Y. Zorian, "On-Line Testing for VLSI – A Compendium of Approaches," Journal of Electronic Testing: Theory and Applications, 1998, vol.12, pp. 7-20.

[8] Y. Sato, Seiji Kajihara, Y. Miura, T. Yoneda, S. Ohtake, M. Inoue, H. Fujiwara, "A Circuit Failure Prediction Mechanism (DART) for High Field Reliability," 8th IEEE Int'l Conf. on ASIC, 2009, pp. 581-584.

[9] X. Fan, S.M Reddy, S. Wang, S. Kajihara, Y. Sato, "Genetic algorithm based approach for segmented testing," IEEE/IFIP Int'l Conf. on DSNW, 2011, pp.85-90.

[10] S.Wang, Seiji Kajihara, Yasuo SATO, X. Fan, S.M Reddy,"A pattern Paritioning Algorithm for field test," *Proc. 2nd Int'l Workshop on Reliability Aware System Design and Test (RASDAT'11)*, 2011, pp.31-36.

[11] S. Dikic, L.-J Fritz, D.Dell'Aquia, "BIST and Fault Insertion Re-use in Telecom Systems," Int'l Test Conf., 2001, pp.1011-1016.

[12] J. Braden, Q. Lin, B. Smith, "Use of BIST in FIRETM Servers," Int'l Test Conf, 200, pp.1017-1022.

[13] S. Kajihara, M. Matsuzono, H. Yamaguchi, Y. Sato, K. Miyase, and X. Wen, "On Scan-in Pattern Compaction with Multi-Cycle and Multi-Observation Scan Test," Int. Sympo. on Communications and Information Technologies, 2010, pp. 723-726. **doi**: 10.1109/ISCIT.2010.5665084.

[14] Y. Sato, H. Yamaguchi, M. Matsuzono and S. Kajihara, "Multi-Cycle Test with Partial Observation on Scan-Based BIST Structure," Asian Test Symp., pp. 54-59, **doi**: 10.1109/ATS.2011.34.

[15] Senling WANG, Yasuo SATO, Seiji KAJIHARA, Kohei MIYASE, “Scan-Out Power Reduction for Logic BIST,” IEICE Transactions on Information and Systems,vol.E96-D, no.9, pp.2012-2020, doi: 10.1587/transinf.E96.D.

[16] K. Yoon, “A Reconciliation among Discrete Compromise Situation,” Journal of Operational Research Society,1987,vol.38, no. 2, pp.277-286.

[17] D. Bertsimas and J.N. Tsitsiklis, “Simulated Annealing,” Statistical Science, 1993, vol. 8, no. 1, pp. 10-15.

[18] Minh Hoai Nguyen, Fernando de la Torre, “Pattern Recognition: Optimal Feature Selection for Support Vector Machines,” *Proc. 43 Robotics Institute*, Carnegie Mellon University, Pittsburgh, USA, 2010, pp. 584–591.

[19] Scikit-Learn, “Machine Learning in Python,” 2019, [Online]. Available: https://scikit-learn.org/stable/

[20] P. Girard, N. Bicolici, and X. Wen, “Power-Aware Testing and Test Strategies for Low Power Devices,” Springer, ISBN 978-1-4419-0927-5, New York, 2010.

[21] H. Iwata and J. Matsushima, “Multi-Configuration Scan Structure for Various Purposes”, *Proc. IEEE 25th Asian Test Symposium,* Hiroshima, pp. 131-131, Nov. 2016. **doi**: 10.1109/ATS.2016.32

[22] N. A. Touba, “Survey of Test Vector Compaction Techniques”, IEEE Design & Test of Computers, vol. 23, no. 4, pp. 294-303, April 2006. **doi**: 10.1109/MDT.2006.105

[23] F. Zhang et al., “Putting Wasted Clock Cycles to Use: Enhancing Fortuitous Cell-Aware Fault Detection with Scan Shift Capture”, *Proc. IEEE Int’l Test Conf.,* Fort Worth, pp. 1-10. **doi**: 10.1109/TEST.2016.7805828

[24] G. Mrugalski, J. Rajski, J. Solecki, J. Tyszer and C. Wang, “Trimodal Scan-Based Test Paradigm,” IEEE Trans. on VLSI Systems, vol. 25, no. 3, pp. 1112-1125, **doi**: 10.1109/TVLSI.2016.2608984.

[25] S. Milewski, N. Mukherjee, J. Rajski, J. Solecki, J. Tyszer and J. Zawada, “Full-Scan LBIST with Capture-per-Cycle Hybrid Test Points,” *Proc. IEEE Int’l Test Conf.*, Fort Worth, 2017, pp. 1-9, **doi**: 10.1109/TEST.2017.8242036

[26] H.-C. Tai, K.-T. Cheng and S. Bhawmik, “Improving the Test Quality for Scan-based BIST Using a General Test Application Scheme,” *Proc. Design Automation Conf.*, New Orleans, 1999, pp. 748-753, **doi**: 10.1109/DAC.1999.782113

[27] Y. Huang, I. Pomeranz, S. M. Reddy and J. Rajski, “Improving the Proportion of At-Speed Tests in Scan BIST,” Int’l. Conf. on Computer Aided Design, San Jose, 2000, pp. 459-463, **doi**: 10.1109/ICCAD.2000.896514.

[28] I. Pomeranz, “A Multicycle Test Set Based on a Two-Cycle Test Set With Constant Primary Input Vectors,” IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 2015, vol. 34, no. 7, pp. 1124-1132, **doi**: 10.1109/TCAD.2015.2408257

[29] J. Rearick, “Too Much Delay Fault Coverage is a Bad Thing”, *Proc. Int’l Test Conf., Baltimore*, MD, 2001, pp. 624-633, **doi**: 10.1109/TEST.2001.966682 .

[30] S. Wang, Hanan T. Al-Awadhi, et al., “Structure-Based Methods for Selecting Fault-Detection-Strengthened FF under Multi-cycle Test with Sequential Observation,” *Proc. IEEE Asian Test Symposium,* Hiroshima, 2016, pp. 209-214, **doi**: 10.1109/ATS.2016.40.

[31] S. Wang, Y. Higami, H. Iwata, J. Matsushima and H. Takahashi, “Automotive Functional Safety Assurance by POST with Sequential Observation,” IEEE Design & Test Magazine, 2018, vol.35, no.3, pp.39-45, **doi**: 10.1109/MDAT.2018.2799801.

[32] S. Wang, Y. Higami, H. Takahashi, H. Iwata, Y. Maeda and J. Matsushima, “Fault-Detection-Strengthened Method to Enable the POST for Very-Large Automotive MCU in Compliance with ISO26262,” *Proc. IEEE 23rd European Test Symposium, Bremen*, 2018, pp. 1-2, **doi**: 10.1109/ETS.2018.8400707.

[33] E. K. Moghaddam, J. Rajski, S. M. Reddy and M. Kassab, “At-Speed Scan Test with Low Switching Activity,” Proc. IEEE 28th VLSI Test Symposium, Santa Cruz, 2010, pp.177-182, **doi**: 10.1109/VTS.2010.5469580.

[34] Y. Sato, S. Wang, T. Kato, K. Miyase and S. Kajihara, “Low Power BIST for Scan-Shift and Capture Power”, Proc. IEEE Asian Test Symposium, Niigata, pp. 173-178, 2012. **doi**: 10.1109/ATS.2012.27

[35] S. Wang, et al., “Capture-Pattern-Control to Address the Fault Detection Degradation Problem of Multi-Cycle Test in Logic BIST,” Proc. IEEE Asian Test Symposium, Hefei, 2018, pp.155-160.

[36] K. Yoon, “A Reconciliation among Discrete Compromise Situation,” Journal of Operational Research Society, vol.38, no. 2, pp.277-286, 1987. **doi**: 10.2307/2581948

[37] M. Noda, S. Kajihara, Y. Sato, K. Miyase, X. Wen and Y. Miura, “On Estimation of NBTI-Induced Delay Degradation,” 15th IEEE European Test Symposium, Praha, 2010, pp. 107-111, **doi**: 10.1109/ETSYM.2010.5512772.

[38] D. Sengupta and S. S. Sapatnekar, “Estimating Circuit Aging Due to BTI and HCI Using Ring-Oscillator-Based Sensors,” in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 36, no. 10, pp. 1688-1701, Oct. 2017, **doi**: 10.1109/TCAD.2017.2648840.

[39] T. M. Mak, “Infant Mortality-The Lesser Known Reliability Issue,” 13th IEEE International On-Line Testing Symposium (IOLTS 2007), Crete, 2007, pp. 122-122, **doi**: 10.1109/IOLTS.2007.40.

[40] A. BENSO, A. BOSIO, S. D. Carlo, G. D. Natale and P. PRINETTO, “ATPG for Dynamic Burn-In Test in Full-Scan Circuits,” 2006 15th Asian Test Symposium, Fukuoka, 2006, pp. 75-82, **doi**: 10.1109/ATS.2006.260996.

[41] Wang, Senling, “Studies on Test Application at Field Test and Low Power Logic-BIST,” Kyushu Institute of Technology, 2017, Japan

[42] Brian baily,“Aging Problems At 5nm And Below,” 2020, [Online]. Available https://semiengineering.com/aging-problems-at-5nm-and-below.

[43] H. Iwata and J. Matsushima, “Multi-configuration Scan Structure for Various Purposes,” 2016

proc. IEEE Asian Test Symposium (ATS), Hiroshima, pp. 131-131.

[44] Koza, John R.; Bennett, Forrest H.; Andre, David; Keane, Martin A., (1996), " Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming," Artificial Intelligence in Design '96. Springer, Dordrecht. pp. 151–170. **doi**:10.1007/978-94-009-0279-4_9.

[45] ML.Bushnell,VD.Agrawal, "Essentials of Electronic Testing for Digital, Memory & Mixd-Signal VLSI Circuits," Kluwer Academic Publishers, 2000.

[46] J. Savir, "Skewed-Load Transition Test: Part I, Calculus," *Proc. of ITC,* 1992, pp. 705-713.

[47] Hanan T. Al-Awadhi, Tomoki Aono, Senling Wang, Yoshinobu Higami, Hiroshi Takahashi , Hiroyuki Iwata, Yoichi Maeda, Jun Matsushima, "FF-Control Point Insertion (FF-CPI) to Overcome the Degradation of Fault Detection under Multi-Cycle Test for POST," IEICE Transactions on Information and Systems, 2020, vol. E103.D, no. 11, pp.2289-2301, **doi**: 10.1587/transinf.2019EDP7235.

[48] Hanan Al Awadhi, Senling Wang, Yoshinobu Higami and Hiroshi Takahashi "Pattern Partitioning based Field Testing for Improving the Detection Latency of Aging-induced Delay Faults," International Technical Conference on Circuits, Systems, Computers, and Communications (ITC-CSCC2017), 2017, pp.21-24.

[49] Senling Wang, Hanan T. Al-Awadhi, Masatoshi Aohagi, Yoshinobu Higami, Hiroshi Takahashi, "Feasibility of Machine Learning Algorithm for Test Partitioning," International Technical Conference on Circuits, Systems, Computers, and Communications (ITC-CSCC2019), 2019, pp. 1-4.