8-28-2012

A truly embedded test structure for design-for-manufacturability, hardware security and VLSI testing

Charles D. Lamech

Charles D Lamech

Candidate

Electrical and Computer Engineering

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Dr. Jim Plusquellic , Chairperson

Dr. Payman Zarkesh-Ha

Dr. Chintan Patel

Dr. Abhishek Singh

A Truly Embedded Test Structure for Design-for-Manufacturability, Hardware Security and VLSI Testing

By

Charles D Lamech

B.E., Electrical and Electronic Engineering, Anna University, 2005
M.E., Applied Electronics, Anna University, 2007

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Engineering

The University of New Mexico

Albuquerque, New Mexico

July, 2012

Dedication

This work is dedicated to my parents Lamech and Esther who have made this possible through their love and support.

Acknowledgments

I would like to thank all who helped me directly or indirectly in making this work possible. First of all, I thank God for giving me the courage and wisdom to make this journey a successful one. I owe gratitude to my parents, sister and brother for all their encouragement and support, without which I wouldn't have been able to come this far.

I would like to extend my sincere and heartfelt thanks to my advisor Dr. Jim Plusquellic for his guidance, support, and contributions throughout this work. He is the main source of inspiration for me throughout this research work. I am so grateful for all the opportunities and teachings that he offered me.

I thank my good friend and colleague Jim Aarestad from the bottom of my heart for all his support and help through the course of this work. I also thank Dr. Ryan Helinski, Jing Ju, and all my friends at the Electrical and Computer Engineering department at the University of New Mexico.

A Truly Embedded Test Structure for Design-for-Manufacturability, Hardware Security and VLSI Testing

by

Charles D Lamech

ABSTRACT OF DISSERTATION

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

Engineering

The University of New Mexico

Albuquerque, New Mexico

July, 2012

A Truly Embedded Test Structure for Design-for-Manufacturability, Hardware Security and VLSI Testing

by

Charles D Lamech

B.E., Electrical and Electronic Engineering, Anna University, 2005
M.E., Applied Electronics, Anna University, 2007

Ph.D., Engineering, University of New Mexico, 2012

ABSTRACT

As the feature sizes continue to shrink in advanced VLSI technologies, the impact of process variations on yield losses has become significant. Moreover, the process variations in path delays are increasingly dependent on circuit context. Given this “neighborhood” dependence, characterization of path delays is best carried out using embedded techniques applied to actual product macros, as opposed to scribe line test structures or embedded ring oscillators (RO). In addition to characterizing performance, such delay measurement techniques can be used to improve model-to-hardware correlation. Furthermore, these techniques can also be used in several contexts including defect detection, post-silicon debug and hardware-security.

In this dissertation, I propose a high-precision and low-overhead embedded test structure, called REBEL, for measuring path delays in the circuit context. REBEL is minimally invasive to the design, as it leverages the existing scan structures, and easy to

integrate because it can be completely automated within the DFT synthesis flow. The proposed embedded test structure and two of it's applications, namely path delay variability characterization and hardware Trojan detection, are verified in silicon.

REBEL architecture for both level sensitive scan design (LSSD) and MUX-D scan designs are proposed and verified in 90nm ASICs and 130nm FPGAs respectively. The LSSD-based REBEL structure is integrated into a 32-bit pipelined floating point unit (FPU), implemented in IBM's 90nm technology, and the path delay measurements from 62 copies of the chip are used to analyze die-to-die and within-die variations. The MUX-D-based REBEL is integrated into a pipelined Advanced Encryption Standard (AES) design and wrapped with random built-in self test (BIST) architecture. The system is then implemented in Xilinx Virtex2Pro FPGAs and the data collected from 30 copies of FPGA are used to analyze the variability in path delays.

As a second application, REBEL's effectiveness for detecting delay anomalies introduced by Hardware Trojans is demonstrated. Trojan emulation circuits, designed to model internal wire loads introduced by a hardware Trojan, are inserted into the design at multiple places. The emulation cell incorporates an analog control pin to allow a variety of hardware Trojan loading scenarios to be investigated. Using the hardware data collected from 62 chips fabricated in 90nm CMOS technology, I demonstrate that REBEL can detect the small delay anomalies introduced by hardware Trojans. I evaluate the detection sensitivity of REBEL for detecting hardware Trojans using linear regression analysis, which deals with the chip-to-chip variation effectively and improves the detection sensitivity.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

As VLSI technologies continue the pursuit of Moore's Law, the operating frequencies of the modern integrated circuits have reached multi-gigahertz and the timing requirements has become more rigorous. On the other hand, as feature size shrinks, process variations are increasing in magnitude and have a significant impact on yield loss [1]. With worsening variability due to printability challenges in advanced technology nodes, accurate variation models are required to mitigate yield losses. In addition, process variations are becoming increasingly sensitive to the design-context, which challenges conventional test structures to capture delay variations that occur in actual product macros. High precision embedded delay measurement structures allow efficient characterization of die-to-die and within-die process variations and improve the model-to-hardware correlation.

Delay defects originating from process-related issues have a significant impact on the quality and reliability of the product. The on-chip delay measurement techniques can provide a cost-effective alternative for delay defect detection and silicon debug. Rather than testing the chip with all possible worst-case test vectors, it is better to measure the delays and to check if the slacks are large enough to tolerate all the possible delay

variations. Furthermore, on-chip path delay measurement structures can be used for hardware security enhancements, such as Trojan detection, physical unclonable functions, etc.

Accurate path delay measurement capabilities are also useful for hardware security enhancements. In particular, accurate delay measurement allows the detection of delay anomalies introduced by malicious alternations to the logic design, called as Hardware Trojans. Hardware Trojans manifest themselves either by adding capacitive load to existing wires or by insertion of additional gates in series with those of the original design, which results in modification of the delay signatures of the design. The embedded test structures that are capable of delivering high resolution measurements of path delays are capable of detecting a wide range of delay anomalies introduced by hardware Trojans.

1.1 Design For Manufacturability

As the variability magnitude increase with the scaling, designers increasingly rely on accurate variation models to mitigate yield loss. Since even marginal increases in yield result in a significant improvement in profitability and product quality [1], a broad range of work is ongoing to characterize within-die and die-to-die process variations (PV), spurring the development of area-efficient structures and methods for validating variation models.

Process Variations are increasing in magnitude and are becoming increasingly sensitive to the design-context, which challenges conventional test structures to capture delay variations that occur in actual product macros. These trends are driven by, among

other factors, increasing variability in process control, which worsens with scaling. In sub wavelength processes (<193nm), for example, photo lithographic interference causes distortion in layout structures [2]. Reticle enhancement techniques (RET) help with printability issues, but are less effective for advanced technology nodes [1]. Both random and systematic within-die PVs are growing more severe with shrinking geometries and increasing die size [3][4].

PV challenges are most apparent in an across-field (within-die) context. The main sources of these are due to optical source limitations, and layout-based systematic effects (pitch, line-width variability, and microscopic etch loading [5][6][7]). Unfortunately, traditional die-to-die level testing and measurement methods, e.g., scribe-line structures, are ineffective for context-sensitive, within-die characterization. More recent efforts to embed test structures, such as distributing a set of ring oscillators across the layout, are capable of capturing within-die variations, but are becoming increasingly less accurate as predictors of delay variations in actual product macros. Truly embedded test structures, such as those that measure delay [8][9][10] and power [11] characteristics of the macro itself, offer the best solution, but are difficult to integrate without having an adverse impact on area overhead, yield loss, performance, I/O interface, test cost, etc. of the product design.

1.2 Hardware Trojan Detection

Hardware Trojans have emerged as a new threat to the security and trust of integrated circuits (ICs). Hardware Trojans are defined as deliberate and malicious modifications to the logic function implemented within digital and mixed signal chips.

Hardware Trojans can be designed to shutdown the chip at some per-determined time and/or when some specific signal or data pattern is received. They may also be designed to remain hidden while leaking confidential information covertly to the adversary. Determining whether a hardware Trojan has been inserted into a chip is extremely difficult for a variety of reasons, e.g., nanometer feature sizes and chip design complexity combine to make optical inspection difficult or impossible.

Parametric anomaly detection Strategies (side-channel analysis) has showed better detectability. As the side-channel response signals, delay, dynamic power, leakage current etc., are analog in nature, it enables the use of highly sensitive detection techniques. These detection techniques may be able to detect functional Trojans by measuring their *secondary* action characteristics even without activating them. For example, the additional gates changes the delay signature of the combinational paths associated with them even if those additional gates are not activated. If the additional gates are not associated with any of existing circuits, the leakage power signature will change which in turn will allow Trojan detection.

The path delays of different regions in a chip are used as signature for Trojan detection [12]. Any alternation to the logic will eventually results in change in the delay characteristics of the paths associated with it. ATPG tools are used to generate test patterns that will cover as many parts of the chip as possible. A series of signatures were constructed from path delays of the golden models, that chips are validated against for Trojan Detection.

Although the delay based Trojan detection methods are more effective and minimally invasive, detection sensitivity is critical as the path delays vary from one chip

to other due to Process and Environmental (PE) variations, which sets the lower limit for Trojan detection sensitivity. As the process variations magnitude is high in cutting-edge technologies, PE variations have higher impact on detection sensitivity. Also, On-Chip/Off-Chip measurement capabilities are also critical for Trojan Detection Sensitivity.

In this work, I propose a truly embedded test structure (ETS), called REBEL, for path delay measurement that can be used in several applications including design for manufacturability, delay defect detection, post-silicon debug and hardware security. I describe the architecture of REBEL and demonstrate its effectiveness for detecting delay defects using the simulation data. REBEL is embedded in a test macro, AES SBOX, and simulated with RC-Transistor models to show the path delay measurements. Delay defects are emulated in one of the paths and delay measured using REBEL which is then compared against the delay signatures of defect-Free design and the results are analyzed (Chapter 3). REBEL is integrated with two product macros, AES (Advanced Encryption System) and FPU (Floating Point Unit), and implemented in 90nm CMOS technology. The design automation procedure followed for the test chip that contains REBEL-integrated macros is also presented (Chapter 3).

REBEL is used to measure the path delays of FPU macro and the data collected from 62 copies of the chip are used to analyze the path delay variability in 90nm ASICs. In addition, a BIST system that has REBEL-integrated AES macro as circuit-under-test (CUT) is implemented in 130nm FPGAs to analyze the delay variability in FPGAs. The path delay measurements collected using this BIST system from 30 FPGAs are used for die-to-die and within-die variation analysis (Chapter 4).

Power and Delay Signal-to-Noise requirements for Trojan Detection are first

analyzed experimentally on FPGA platform along with the statistical methods that can improve Trojan detectability (Chapter 5). From this preliminary analysis, it is determined that the hardware Trojans that add capacitive load to the existing wires are harder to detect. Therefore, hardware Trojans that impose additional capacitive load are emulated in several places of the FPU macro using Trojan emulation circuits and REBEL's ability to detect the subtle change in delay signature caused by these Trojans is presented using the data collected from 62 ASICs (Chapter 6). In addition, a statistical method called regression analysis is used to improve the detection sensitivity. I also present the test effort required for identifying the presence of Trojans using a randon-BIST system. Chapter 7 lists the improvements that can be made to the REBEL structure and its other applications. Chapter 8 concludes with the benefits of REBEL for various applications.

CHAPTER 2

Background

2.1 Path Delay Measurement Techniques

With the scaling of semiconductor process technology, the mismatch between the timing models and actual silicon behavior became an important factor and require silicon data to verify the design behavior. Moreover, the silicon data is also used to tune the models and to improve the model-to-hardware correlation. In order to get silicon data several path delay measurement techniques have been proposed. The authors of [13] and [14] propose a technique in which the path delay measurement is achieved by doing multiple captures with fast clocks. This technique suffers from the inductive power supply noise caused by the fast clocks and the delay values are pessimistic as faster clocks cause higher IR drop than that of functional clock.

In [15], the authors propose an on-chip delay measurement hardware for efficient speed binning and the structure is inserted only into the critical paths of the design and provides only limited path delays. The authors of [16] propose an on-chip path delay measurement architecture to measure the delays. However, the path-under-tests are multiplexed into the path delay measurement circuit which can be costly for achieving even a reasonable path coverage. In [17], the authors propose an embedded structure that uses a delay value measurement circuit (DVMC), which computes the delay between two events, to measure the delay between the clock event and the output change. This

structure also suffers from the area overhead and coverage issues and can be used to measure some of the paths. In [18], the authors propose a concurrent online delay measurement architecture for measuring critical path delay measurements.

The authors of [9] and [19] propose path delay measurement using oscillation techniques. In [9], the author propose a method in which consist of sensitizing a path-under-test, incorporating a ring oscillator through a feed back path and measuring the ring oscillator frequency to derive the path delay. However, when a hazard or glitch occurs at the output of the path, the frequency measurement may not give the accurate path delay, in which case the path coverage may be low. The authors of [19] propose a similar technique, in which the feedback loop enters into the path-under-test through a multiplexer instead of flip-flop. This technique may also result in lower coverage.

In this work, I propose an embedded test structure that can measure all the paths that an ATPG can create a test for. Since it leverages the existing scan structures, the area overhead is low. The proposed test structure can measure any path delay, long or short, using functional clock and hence the need for high-speed clock and the power supply noise associated with it can be eliminated. Moreover, the delay values are realistic as it uses the at-speed clock for path delay measurements.

2.2 Delay Variability Characterization

Test structure design for process variation characterization continues to be an active area of research [20][21][22][23][24]. Embedded ring oscillators (EROs) have been successfully used to characterize within-die performance variations [25]. Due to

their simple design and I/O interface, they have been the preferred embedded test structure. For example, the authors of [26] propose a re-configurable Ring Oscillator circuit for measuring delay of individual gates and reported a within-die delay variation of up to 26% on a 65nm process node and a measurement accuracy of 1 ps. Several recent timing structures have been proposed which are fully analog in nature [27], and in [28], a within-die variation characterization system is proposed which uses a on-chip sampling oscilloscope.

Numerous time-to-digital converter (TDC) designs have been previously proposed, simulated, and in some cases, realized in silicon [21][29][25]. Time-to-Digital Converters (TDCs) provide accurate measurement of path delays in the product macros [20][21][29]. Unfortunately, the area overhead on the TDC is high which is not suitable in many applications. In addition, the number of paths that can be measured is also limited by number of TDCs available on the chip. Though several combinational paths can be multiplexed such that different path delays can be measured by selecting the path-under-test, the complexity of the design forces the overhead further. The authors of [27] have proposed a structure to measure on-chip delays using a variety of measurement circuits. However, these devices were intended to serve as the basis for adaptive timing mechanisms to provide on-chip synchronous timing control.

From the references and citations above, it is clear that the use of on-chip ETSs for path-delay timing is not new. Designs of differing global type and complexity have been proposed to address a variety of applications. The main distinguishing characteristic of ETS proposed in this work over those previously proposed is the degree to which it is embedded. The primary focus is on integrating the measurement structure into the

product macros themselves, and on doing so in a minimally invasive manner, i.e., it leverage existing resources such as the scan chain and integrate such that the product timing paths are minimally impacted.

Since the proposed test structure can measure the delay in the circuit-context, the characterization will be accurate. In addition, the proposed structure uses functional or at-speed clock to measure any paths, both short and long paths, and hence the delay values measured are realistic, unlike the faster-than-at-speed tests in which case delay values are pessimistic.

2.3 Hardware Trojan Detection

2.3.1 Trojan Taxonomy

The taxonomy of possible types of Trojan circuits that may present in a chip is presented in [30]. In this paper, the authors categorized the possible Trojan circuits into three major categories according to their physical, activation and action characteristics.

The physical characteristics describes various hardware manifestations of Trojans. Trojans can be, functional(those that changes the functionality of the chip) or parametric (those that changes the parametric properties like delay, power, etc.). Trojans also sub-categorized based on their size relative to the core logic and distribution, tightly coupled in a region of the chip or distributed across.

Activation characteristics based categorization describes when the Trojan is activated. This included Trojans that are always on and that are activated based on some condition (logic or sensor based).

Action characteristics identify the types of disruptive behavior introduced by the Trojan. This includes Trojan that modify the function (add or bypass) and that modify the specifications of the chip(includes defects and reliability issues).

2.3.2 Trojan Detection Strategies

Trojan detection methods can be applied immediately after the chip is returned to the customer, either as a die on a wafer or as a packaged chip, and/or they can be applied continuously during the lifetime of the system. For the later case, board level trusted companions are required. There are three basic approaches for detecting Trojans [30]:

2.3.2.1 IC De-Processing and Failure Analysis Methods

IC de-processing involves applying sophisticated failure analysis techniques such as scanning optical microscopy (SOM), scanning electron microscopy (SEM), pico second imaging circuit analysis (PICA), voltage contrast imaging (VCI), light-induced voltage alternation (LIVA), charge-induced voltage alternation CIVA, etc. Although these techniques can be effective for authentication purposes, they are also extremely time consuming and expensive [30]. Moreover, many require the sample (chip) to be prepared by backside thinning and de-processing operations. Obviously, this approach is not suited for applications in which every chip needs to be authenticated.

2.3.2.2 Functional activation through logic testing

The second approach involves the use of ‘standard’ VLSI fault detection tools, such as automatic test pattern generation (ATPG). Detection of a Trojan is accomplished by applying a digital stimulus and inspecting the digital output of the chip. The digital

stimulus is derived using the netlist of the chip. For Trojans of the parametric type as described in Section 2.3.1 , the netlist of a chip is the same with and without the Trojan. This is true because parametric Trojans are introduced into the existing logic of the chip by violating design rules, i.e., thinning a wire, etc. Therefore, ATPG can be modified to target parametric Trojans [31]. Given their stealthy activation criteria, ATPG directed to generate tests for nodes and paths that are random fault resistant, i.e., difficult to control and/or observe, is likely to yield the best results for activation and detection of Trojans.

Unfortunately, ATPG is not effective for the functional Trojans, which are represented as inserted, additional logic. Without knowledge of this logic and how it is connected to the original logic in the chip, it is impossible for ATPG to perform a directed search for a vector or state that causes activation. If the activation criteria can be determined, then detection would be trivial in many cases, assuming the Trojan modifies the internal state or an output of the chip in some fashion. However, for Trojans that activate and leak information over side channels, e.g., the power supply, digital testing methods are not effective. Therefore, for functional Trojans, an ATPG approach is hindered by two problems, one that deals with activation and another that deals with detection.

2.3.2.3 Parametric Anomaly Detection Strategies

A third approach that can potentially solve these problems involves the measurement and analysis of the chip's side-channel signals [32][33][12][34]. For example, it is possible to stimulate the chip using digital stimuli and then measure the analog response signals of the chip, such as the transient or quiescent power supply current. A second possibility is to stimulate the power grid directly by driving it with a

sine wave at one position and measuring its response at another.

The analog nature of the side-channel response signals enables the use of highly sensitive detection techniques. Such techniques may be able to detect functional Trojans without activating them, i.e., through the measurement of their *secondary* action characteristics as described in Section 2.3.1 . For example, as indicated that functional Trojans are never completely inactive because of the need to continuously monitor for the activation conditions. Consider a Trojan that activates based on a specific state of a data bus in the chip. The implementation of the Trojan, in this case, requires some type of comparator to be installed that monitors the wires of the data bus. The logic of the comparators, e.g., AND gates, switches as the data bus changes and therefore consumes power. Side-channel signal analysis can potentially detect the power anomaly introduced by the operation of the comparator.

Moreover, the highly sensitive nature of side-channel analysis techniques may allow detection of tightly coupled functional Trojans even without the application of a digital stimulus. The presence of the Trojan logic gates adds capacitance to the power grid. The presence of the additional capacitance changes in impulse response of the power grid. The impulse response of the power grid can be tested by injecting an analog stimulus onto the grid at one place and measuring the response at another.

The effectiveness of side-channel-based measurement and analysis techniques can be improved by adopting design-for-hardware-trust (DFHT) techniques, which, for example, add circuitry to support the measurement and analysis processes. On-chip voltage and temperature sensors can be installed to increase the level of sensitivity of side-channel measurement and analysis techniques by providing local observability at

various positions across the 2-D layout of the chip. The DFHT strategy must also incorporate a validation strategy for the on-chip support circuits because of the potential of the adversary to sabotage the sensors.

There has been several work that investigates Trojan detection using functional activation and parametric analysis. In [35], authors used a logic fault simulation based approach for detecting the Trojans that are rare value triggered or time triggered. Also they admit their strategy can be effective in detecting most small combinational Trojans.

Authors in [31] analyzed the amount of time it takes to generate a transition in a functional Trojan and trigger a hardware Trojan. They propose a dummy flip-flop insertion, which eliminates hard-to-activate sites, to increase Trojan activity and hence to reduce the Trojan activation time. This eliminates the need for focusing on the rare conditions proposed in [35]. Insertion of dummy flip-flops increases the area overhead which is not attractive in many product chips.

The side-channel information such as power, temperature, and electromagnetic (EM) profiles have been used to construct a set of signature for an IC family [32]. The set of fingerprints were developed using a few ICs from a batch and only these ICs would have to be invasively tested to ensure that they were all authentic. The remaining ICs are verified using statistical tests against the fingerprints. The results show that Trojans that are 3–4 orders of magnitude smaller than the main circuit can be detected by signal processing techniques. This approach is invasive in nature for developing the signature.

Path delays of nominal chips are collected to construct a series of signature, each one representing one aspect of the total characteristics of a genuine design [12]. Chips are validated by comparing their delay parameters to the signature. The comparison of path

delays makes small Trojan circuits significant from a delay point of view. Authors of [36] have introduced a technique for recovery of characteristics of gates in terms of leakage current, switching power, and delay, which utilizes linear programming to solve a system of equations created using non-destructive measurements of power or delays. This technique is combined with constraint manipulation techniques to detect embedded Hardware Trojans. The effectiveness of the approach is demonstrated on a number of standard benchmarks.

Authors of [37], used a technique in which negatively-skewed shadow registers are placed at the end of a number of combinational paths which will allow measuring the path delays. This combinational delay measurement of an arbitrarily large number of register-to-register paths internal to the functional portion of the IC can be used to provide the desired authentication and design alteration (including Hardware Trojan implantation) detection. They claim that this technique is low-cost delay measurement technique that does not affect the main IC functionality and can be performed at-speed at both test-time and run-time.

Ring oscillator network (RON) based structure is proposed for Trojan detection [38]. This structure featuring the ability to detect Trojans that cause power fluctuations, thereby uncovering the malicious inclusion. A number of ring oscillators (ROs) acting as power monitors, distributed across the entire IC, constitute the RON, which takes into account the noise caused by the Trojan gates and those caused by both inter-die and intra-die process variations. Power signature is created from the output of each ring oscillator. The overhead of this approach is high as the Ring Oscillators has to be placed all across the chip.

Hardware Trojan detection based on the analysis of an ICs power supply transient signals is also proposed [30]. Power supply transient signal analysis method for detecting Trojans that is based on the analysis of multiple power port signals [33]. A calibration technique is proposed to deal with the adverse effects of process variations on Trojan detection resolution.

Chip's IDDQs (steady-state current) is also used as a side-channel signal for Trojan detection methodology. Steady-state current which are measured simultaneously from multiple places on the chip. It has been shown that IDDQ analysis along with calibration technique for eliminating the process variation effects, resolution enhancement up to a 1000 are possible over conventional single power supply current measurement techniques.

The authors of [39] propose a multi-parameter (dynamic current and maximum frequency) approach for detecting hardware Trojans. Although we also analyze these parameters, the focus of our work is on analyzing within-die and chip-to-chip PE variations and their impact on Trojan detection sensitivity. In [40], the authors decompose the design into blocks and use a self referencing approach for improving sensitivity. In Chapter 3, a calibration techniques is proposed that leverages regional correlations of process parameters on the same chip and additionally transforms measured chip data to a single golden model of the design.

Given that the path delay signatures of the chip can be used to detect the Trojan, it is highly important to have on-chip capabilities for measuring path delays for effective and fast Trojan detection process. In this work, the proposed embedded test structure, REBEL is used to measure path delays accurately.

2.3.3 Sensitivity Requirement for Trojan Detection

There are several works that investigate process variations in ASICs and FPGAs. The authors in [18] use at-speed testing to measure the effect of process variations on several 90nm custom-designed LUT arrays. The authors in [41] measure the effect of both random and systematic process variation on 18 Altera Cyclone II devices with 5- and 7-stage ROs. The authors in [42] measure the level of process variation in 10 FPGAs chip using a 135-stage RO. There is a growing body of work that make use of ROs to define PUFs. For example, the authors of [43] propose two methods to improve the quality of the PUF, a hardware primitive which exploits the manufacturing process variation of the die to create a unique signature, in the presence of the environmental noise and variation in the die. The work presented in Chapter 3 extends this research by investigating noise as well as both intra and inter-die power and delay variations, and their impact on reducing measurement sensitivity to anomalies introduced by changes in circuit configurations, i.e., emulated Trojans.

FPGAs have been used in several works to investigate Trojans. The authors of [44] design and implement 8 different types of the Trojans on the LUT based FPGA which are unlikely or nearly impossible to detect with functional testing. The authors in [12] propose delay signature defined from a set of test vectors to distinguish Trojans. The authors in [37] propose the use of a shadow latch to detect delay anomalies for the purpose of both authentication and hardware Trojan detection. Our work builds on this research by analyzing noise and process variations in actual hardware, and treats the problem of detecting Trojan anomalies from a statistical perspective.

CHAPTER 3

REBEL : A Truly Embedded Test Structure for Delay Measurement

3.1 Scan Test Structures

Scan design, the most widely used structured DFT methodology, attempts to improve testability of a circuit by improving the observability and controllability of storage elements in the sequential design. Typically, this is accomplished by converting the sequential design into a scan design with three modes of operation: Functional mode (or Normal mode), Shift mode and Capture mode. In the functional mode all the test signals are turned off and the scan design operates in the functional configuration. In both shift and capture mode, a test signal SE (scan enable) is often used to turn on all the test related fixes that are necessary to simplify the test, debug, diagnosis tasks. These circuit modes are distinguished using additional test signals or test clocks.

There are three widely used scan cells: Mux-D Scan, Clocked-LSSD Scan and LSSD Scan [45].

Mux-D scan cells are the most widely used scan cells. This scan cell composed of D flip-flop and Multiplexer(Fig 3.1(a)). The multiplexer uses SE input to select between scan input (SI) and data input (D). Major advantages of Mux-D scan cells are their compatibility to modern designs using single clock D flip flops and comprehensive support provided by existing automation tools. The disadvantage is that each Mux-D scan cell adds a multiplexer delay to the functional path. An edge triggered clocked-scan cell

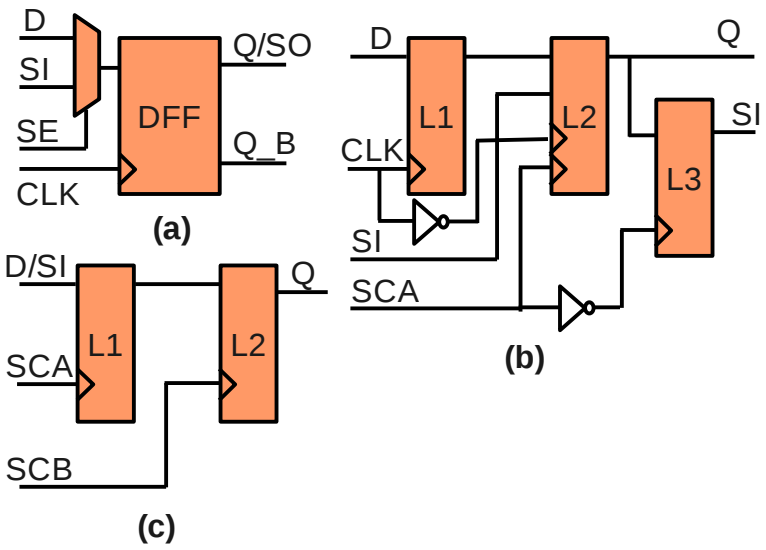


Fig. 3.1: Scan Cell Types (a) Mux-D Scan Cell (b) Clocked-Scan Cell (c) LSSD Scan Cell

also be used to replace a D flip-flop in a scan design. However, in clocked-LSSD scan cell, input selection is conducted using two independent clocks, data clock (CLK) and shift clock (SCA)(See Fig 3.1(b)). The major advantage of using clocked-LSSD scan cells is that it results in no performance degradation on the data input. The major disadvantage is that it requires additional shift clock which has to be routed across the chip. Additional clock (SCB), that is non-overlapping with SCA, can be used to avoid race condition.

While Mux-D and Clocked LSSD scan cells are edge triggered, LSSD scan cell is level sensitive device. LSSD scan cell contains two latches, a master two-port latch and s slave single port latch (Fig. 3.1(c)). Clocks CLK, SCA, SCB are used to select between data input (D) and scan input (SI). In order to guarantee race-free operation, clocks SCA and SCB are applied in non-overlapping manner. Major advantage of LSSD scan cell is that it allows us to insert scan into latch based design. In addition, LSSD are guaranteed to be race-free. But the major disadvantage, however is that it requires routing of

additional clocks (SCA, SCB).

3.2 REBEL – REgional dELay Behavior

In this work, I propose a scan chain based Embedded Test Structure (ETS), called REBEL (REgional dELay Behavior), that is capable of measuring the path delays in the circuit context of actual product macros. The control logic integrated into REBEL allows for both the creation of a delay chain in a segment of scan chain and the selection of a path-under-test (PUT) to drive its input. This is achieved by modifying the logic of the scan control path in the design. A standard launch-off-capture transition test vector is used to create the transition in the PUT which propagates through the core logic and then along this delay chain. The propagation is stopped after a specific period, called the launch-capture (LC) period, which takes a 'digital snap-shot' of the propagating signal, effectively digitizing the voltage behavior of the path's output. The delay of the PUT is calculated by subtracting the delay along the delay chain from the LC period.

3.2.1 Advantages of REBEL

REBEL provides several significant benefits:

1. REBEL allows for the delay measurement of short paths using at-speed clocks by extending them into a delay chain and therefore eliminates the need for high speed clocks which may require expensive automatic test equipment (ATE).

2. Since at-speed or slower-than-at-speed clocks are used for delay measurements, the accuracy of measurement is improved by reducing the power supply noise associated with the high speed clocks.

3. The resolution of the delay measurement can be scaled down to any required value.

4. REBEL is minimally invasive to the design as it leverages the scan structures that already exist in the designs.

5. REBEL can potentially speed up the path delay measurement process because it captures the temporal behavior of paths in a sequence of flip-flops. Therefore, path delay measurements can be obtained using a small number of repeated application of the test pattern, with as few as only one depending on the desired timing accuracy. In contrast, clock strobing techniques require many applications of the test pattern sequence to achieve the same result.

6. Design overhead is negligible as the integration of REBEL can be automated in a design-for-test (DFT) synthesis flow.

These capabilities of REBEL allow it to be used in several contexts including defect detection, design-for-manufacturability, design debug and hardware security.

3.3 REBEL Architecture for CLSSD Style Scan Chain

As mentioned earlier the basic idea behind REBEL is to create a delay chain in a segment of scan chain and connect PUT's output to it. In order to achieve, the scan chain architecture has to be modified. In this section, I present the architectural changes required to integrate REBEL into a LSSD style scan chain. As mentioned before, REBEL ETS leverages the scan chain architecture to measure delay variations, in particular, it uses a special configuration of flush delay mode that is available in LSSD-style scan

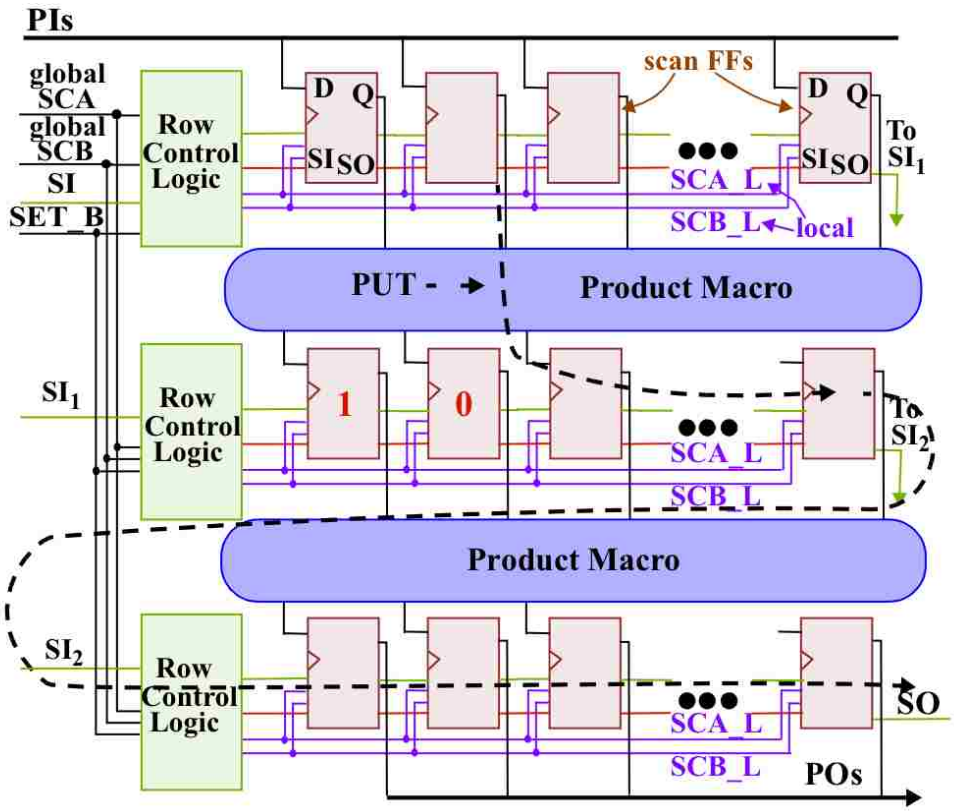


Fig. 3.2: REBEL Integration Strategy

chains. It has been demonstrated in a previous work the promise of capturing regional delay variations using a special launch-capture timing sequence applied while in flush delay mode [10]. This is an extension to that technique here by allowing output signals from design macros to be inserted into the flush delay chain for path delay measurements.

In this section, we describe the overall integration strategy and operation of the REBEL ETS. As indicated previously, REBEL is integrated into the scan chain directly, as shown in Fig. 3.2 for a clocked-LSSD-style scan architecture. Here, the regions labeled ‘Product Macro’ are functional units composed of combinational logic. Three scan chain segments are shown that serve to deliver input and capture output from these macros. The three blocks labeled Row Control Logic identify components of the REBEL ETS, and are described below. Beyond these three ‘header’ blocks, smaller blocks are

also needed for local scan signal control for each of the scan FFs.

The basic idea is to generate a transition on the inputs to the macro using a standard launch-off-capture transition fault test. In this scenario, the scan chain is loaded with the initial pattern of the 2-pattern test and the system clock (CLK) is used to generate a transition in the core logic by capturing the output of a previous block, or by capturing the PI values, as shown in the Fig. 3.2. One or more transitions are propagated through the macro, as shown by the dotted line labeled PUT for ‘path-under-test’. The PUT’s transition emerges on an output of the macro, and drives the D input of a scan FF in the second row. Special control logic associated with the scan FF (to be described) allows the transition to propagate along the scan chain, as shown by the dotted line. CLK is then de-asserted to halt the propagation, which effectively ‘takes a digital snap-shot’ of the signal propagation behavior along the scan chain, including any glitching that may have occurred. This digital snap-shot is then scanned out for analysis.

For designs that make use of LSSD-style scan, propagation along the scan chain is easy to implement. This is true because LSSD supports a flush-delay (FD) mode of operation. In FD mode, both the scan A clock (SCA) and scan B clock (SCB) are held high, effectively making both latches of the FF transparent, i.e., any transition generated on D propagates to Q after a Δt that represents the delay through the FF. FD mode effectively makes the scan chain a combinational inverter chain.

However, the configuration in Fig. 3.2 differs from the traditional definition of FD mode because only a portion of the scan chain is configured in FD mode. In particular, the scan FFs along the top row and those along the middle row to the point of insertion of the PUT operate in functional mode, and only those to the right(and below) of this point

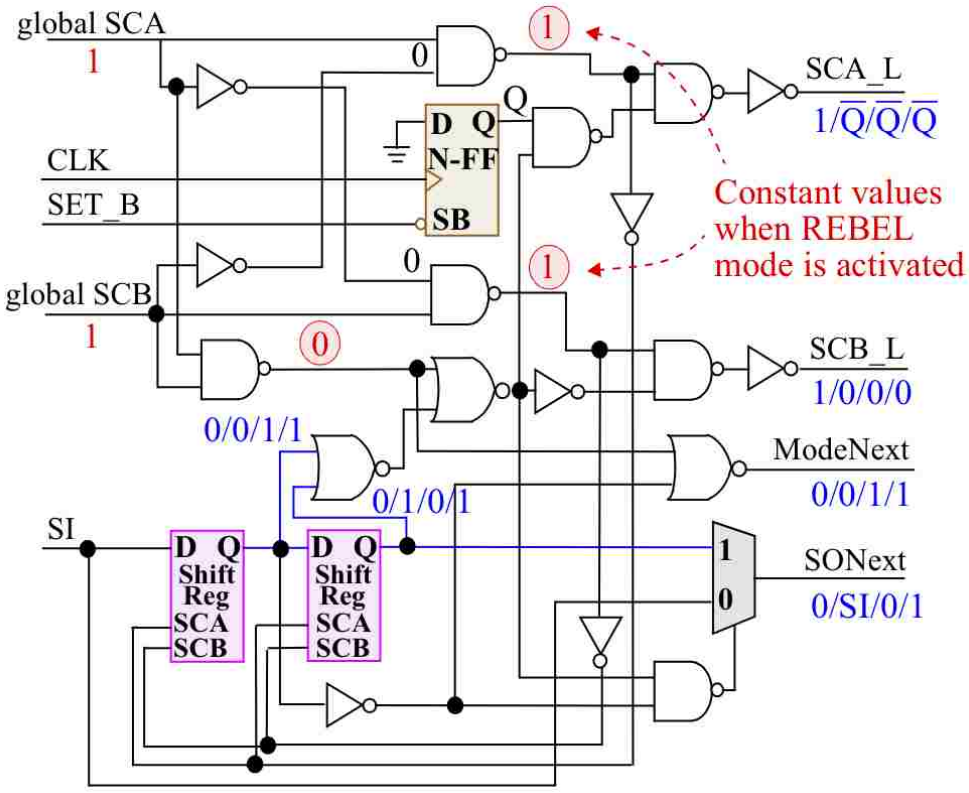


Fig. 3.3: REBEL Row Control Logic for CLSSD Scan Structures

operate in FD mode. In order to realize this configuration, several changes are required to the logic implementing the scan operation.

One of the components to support this dual mode of operation is labeled Row Control Logic (RCL) on the left side of Fig. 3.2. These blocks, in combination with a scan chain encoding scheme and localized scan FF logic, enable the dual mode of operation and provide a mechanism to specify a PUT's output to direct into the scan chain. This is accomplished by configuring several state bits in the RCL, and by loading a specific pattern into the scan chain before the launch-capture (LC) timing sequence (REBEL test) is applied, as described below.

Each RCL block controls a 'row' of scan FFs, called row-FFs, in the following description. Fig. 3.3 shows a schematic diagram of the RCL. The top portion of the

diagram controls local (row-specific) scan clock signals, labeled SCA_L and SCB_L (_L for local) while the bottom portion contains two shift registers (Shift Reg) and mode select logic. A large portion of the RCL logic is dedicated to allow the row-FFs to operate in either of the traditional functional or scan modes of operation. The global (chip-wide) scan signals labeled ‘global SCA’ and ‘global SCB’ are used to specify one of the three possible global operational states. When both are low, functional mode is in effect with CLK controlling the launch-capture activity in the row-FFs. Non-overlapping assertion of these signals causes all scan FFs to act as a shift register, implementing scan mode. The timing mode used by REBEL is specified when both of these signals are asserted. This is illustrated by the ‘1’s on global SCA and SCB in Fig. 3.3.

Note that the two shift registers in the RCL block are conditionally inserted into the scan chain during a scan operation and can therefore be configured prior to a REBEL test. The shift registers’ scan clock inputs (SCA/SCB) are also gated to prevent them from entering FD mode, thereby destroying the state information, when both global SCA and SCB signals are asserted. The state of the two shift registers defines the mode of operation for the row when REBEL mode is activated. Two control bits (as opposed to one) are needed to implement the simultaneous functional and FD modes discussed above because there are actually four possible conditions that need to be handled. The three rows of scan FFs in Fig. 3.2 illustrate three of the four conditions. For example, the scan FFs in the top row need to be in functional mode throughout the REBEL test. In contrast, the scan FFs in the bottom row need to be in FD mode to extend the propagation path of the PUT signal captured in the middle row. Last, the middle row contains scan FFs in both of these modes, i.e., the scan FFs to the left of the PUT insertion point are in

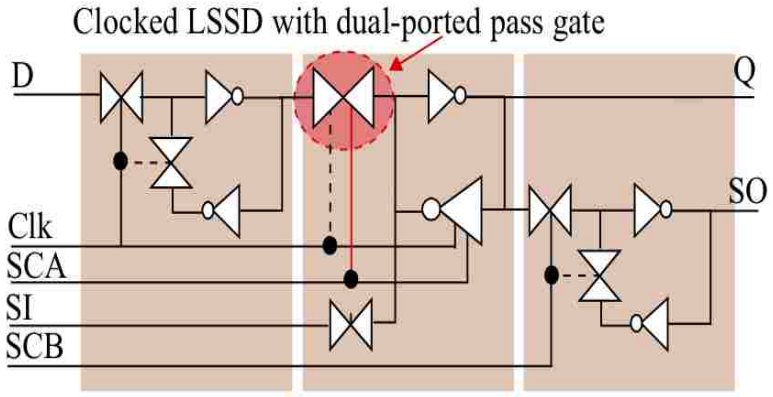


Fig. 3.4: Modified Clocked-LSSD scan FF

function mode while those to the right are in FD mode. The fourth condition is just a special case of this third condition where the insertion point is the left-most scan FF in the row. Table 3.1 identifies the bit configurations that handle these four conditions.

Shift Reg	Functionality
00	All scan FFs in row are in functional mode
01	All scan FFs in row are in FD mode
11/10	Left scan FFs in functional mode, right scan FFs in FD Mode

Table 3.1: Configuration States for Row Control Logic

Before describing the annotations in Fig. 3.3, we turn to the configuration of the scan FFs. Fig. 3.4 shows a clocked LSSD FF (CLSSD), which consists of three latches. The two latches on the left implement the functional path, and are controlled by the system clock. The center latch is dual port and serves both as the slave for the functional path and as the master in the LSSD pair. The right-most latch is the slave latch of the LSSD pair. The top pass-gate of the dual port latch is high-lighted to indicate that it has been modified. In the following paragraphs, it will become apparent that during the REBEL test, both CLK and SCA will be asserted simultaneously during a portion of the

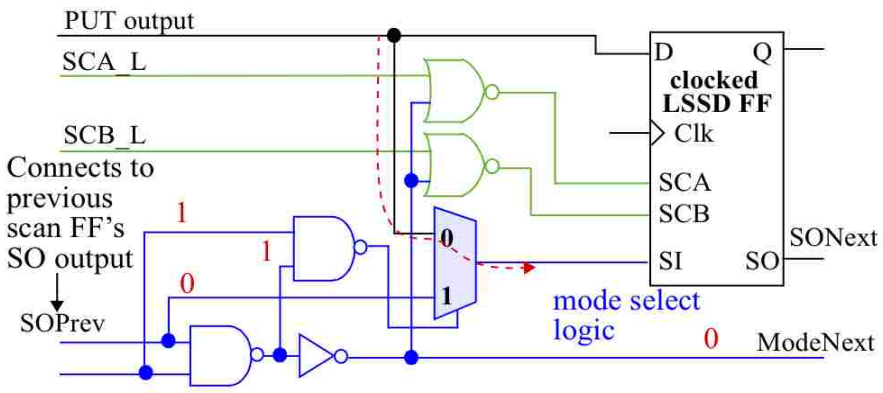


Fig. 3.5: Additional 'Front-End' Logic for CLSSD

test. This creates a potential shorting condition in the dual port latch, i.e., both the master of the functional path and the SI input paths are enabled. To prevent this from happening, we modified the single input pass gate connected to the master's output to include a second input. The second input prevents the master's output from driving the dual port latch when both CLK and SCA are asserted simultaneously.

Fig. 3.5 shows the additional logic required to integrate REBEL into a design with CLSSD-style scan. The functional path's D-input is fanned out to a 2-to-1 MUX. This will allow for the insertion of a macro's PUT into the scan chain during the REBEL test. The local scan signals (SCA_L and SCB_L) are gated by mode select logic shown along the bottom of the figure. The mode select logic incorporates the normal scan path (SOPrev to the SI input), as well as a propagating mode bit (ModePrev to ModeNext). The mode select logic is responsible for selecting the insertion point. This is accomplished by pre-loading the row-FFs with a pattern of all '1's followed by a '0' from left to right along the row-FFs. The '0' in this sequence causes the next scan FF to be configured in a special way, i.e., it allows the PUF output signal to drive the SI pin. The annotation and dotted line in the figure illustrates this case, and assumes the scan FF on the left (not shown) is configured with a '0' bit. Given the scan chain connects the SO

output of each scan FF to the SOPrev of the next scan FF, this arrangement allows the scan chain encoding to specify the PUT insertion point. Moreover, the split mode of operation required for this row is implemented using a propagating mode bit (ModePrev and ModeNext), which is ‘1’ for all scan FFs to the left of the insertion point and ‘0’ to the right. The left-most scan FFs in the middle row of Fig. 3.2 are annotated with a bit configuration that enables the insertion of the PUT at the position shown.

The mode select logic also participates in controlling the local scan signals (SCA_L and SCB_L), and completes the implementing of the four conditions described above in reference to the RCL. The shift registers in Fig. 3.3 are annotated with four states (for the four conditions). The ‘00’ state, which forces functional mode for the row-FFs (row 1 in Fig. 3.2), sets both SCA_L and SCB_L to ‘1’. Given these signal connect to the inputs of the two NOR gates in instances of the scan FFs (as shown in Fig. 3.5), and ‘1’ is the dominant value for a NOR gate, this condition effectively disables FD mode for the entire row. In this case, the ModeNext and SONext output signals of the RCL, which connect to the left-most scan FF’s ModePrev and SOPrev signals, are irrelevant.

The ‘01’ state, as discussed earlier, forces the row-FFs into FD mode (row 3 in Fig. 3.2). This requires both of the SCA_L and SCB_L signals to be set to ‘0’. However, the annotation in Fig. 3.3 indicates the value of SCA_L is ‘Q’, which is the inverted output value of the negative edge triggered FF (N-FF) in the RCL. In the implementation flow for a REBEL test, the initial value of the N-FF is set to ‘1’ by virtue of strobing the SET_B signal low prior to the REBEL test. The REBEL test is defined as a rising edge on CLK (which effectively launches a transition(s) into the macro-under-test), followed by a falling edge on CLK that acts to capture a snap-shot of the PUT’s behavior in the scan

chain. The snapshot is realized by de-asserting the Q output of the N-FF, which occurs when CLK goes low1. This in turn causes the SCA_L output signal from the RCL to transition from ‘0’ (initial value) to ‘1’. From Fig. 3.5, the arrival of the ‘1’ on SCA_L signals of the scan FFs de-asserts the SCA input and turns off FD mode. This action captures the snapshot of the PUT’s voltage behavior in the scan chain.

The ModeNext output signal of the RCL configured in the ‘01’ state is ‘0’. The ‘0’ propagates along the mode select logic of the row and forces all row-FFs to operate in scan mode, i.e., SO to SI to SO, and so on. This condition allows for the propagation of the PUT’s signal along the scan chain. The SONext signal’s value for state ‘01’ is given as ‘SI’ to indicate that this signal is driven from the SI input of the RCL. Therefore, the scan chain by-passes (and preserves the contents of) the state elements in the RCL. The SI input in turn connects to the SO signal from the right-most scan FF of the previous row, effectively extending the scan path across rows (see Fig. 3.2).

Finally, the ‘11’ state in the RCL configures a split mode of operation in the row-FFs and connects a specific PUT output into the scan chain (row 2 in Fig. 3.2). The mode select logic in the scan Ffs work together with the RCL block to implement this split mode of operation. The behavior of the SCA_L and SCB_L outputs are identical to those described above for state ‘01’. The difference lies in the state of the ModeNext and SONext output signals in Fig. 3.3. As noted above, a string of ‘1’s followed by a ‘0’ are pre-loaded into the scan chain to specify the PUT insertion point. The ‘1’ on the ModeNext output propagates along the mode select logic, described earlier in reference to Fig. 3.5 until a ‘0’ is encountered in scan FFs of the row. This causes the next scan FF to be configured as the insertion point. The remaining scan FFs in the row are configured

in FD mode because the mode bit is inverted to a ‘0’ after the insertion point. RCL state ‘10’ behaves identically but allows the insertion point to be the left-most scan FF in the row.

The REBEL support logic is designed such that it minimizes the impact on the functional behavior of the design. There are two components of REBEL that impact the functional operation. The first is the change of the CLSSD as shown in Fig. 3.4, and the second is the fan-out of the D input to the 2-to-1 MUX as shown in Fig. 3.5. Each of these changes adds a small Δt to the functional path.

3.4 REBEL Architecture for MUX-D Style Scan Chain

Although LSSD style scan chains are used in many products, MUX-D based scan designs are the most commonly used in product designs due to the following reasons: 1) compatibility to modern designs using single clock D flip flops, and 2) comprehensive support provided by existing automation tools. In this section, I present the REBEL integration strategy for MUX-D style scan designs. Integration of REBEL into MUX-D scan is easy and even less invasive than it is for CLSSD. The overall operation of REBEL for MUX-D scan is very similar to that described for CLSSD. The main difference is that the launch and capture is accomplished using rising edges of clock (as opposed to a rising and falling edge for CLSSD). Also, an additional primary input is required to specify FD mode. This global signal is routed to the RCL blocks (explained later).

Since MUX-D scan cells are operated by single clock, there is no inherent flush-delay mode to create delay chain. However, at any point of time one of the latches in D flip-flop will be transparent. Therefore, the transparent latch of all the scan cells in a

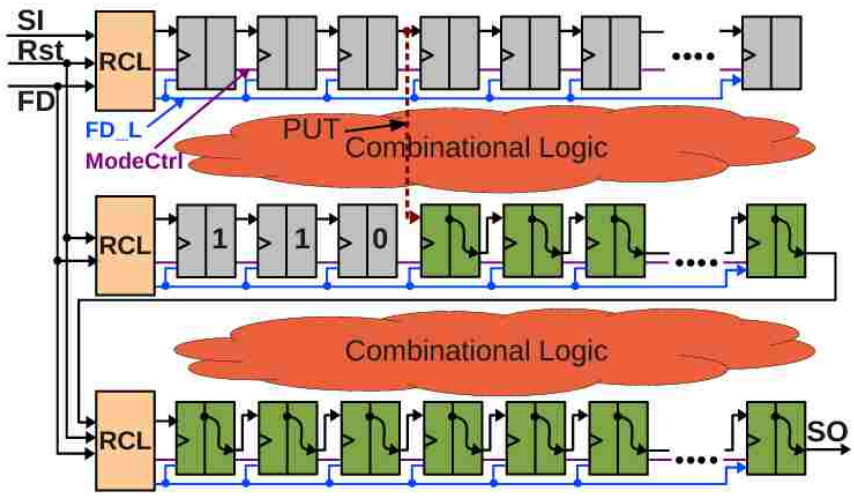


Fig. 3.6: REBEL Integration strategy for MUX-D Scan Chain

specific segment are connected together to create a delay chain.

The key objective in the MUX scan implementation is to implement a FD mode, i.e., a combinational path, using the latches within the MUX scan FFs. This can be achieved by adding a ‘tap-point’ to the master latch, called QM_Next in Fig. 3.7, and routing this signal to a 2-to-1 MUX in the next scan FF of the scan path (labeled QM_Prev in Fig. 3.7). The SE input in Fig. 3.7 refers to the globally routed scan enable signal (already required for MUX scan). SE is set to ‘1’ when we are in scan mode, and ‘0’ when in functional or FD (REBEL test) mode. The remaining logic gates are inserted to implement the four conditions described earlier.

For example, to configure a row in functional mode (row 1 in Fig. 3.2), the RCL block places a ‘0’ on the FD_L and Mode_Prev wires. Support logic (shown in blue) will select D input and hence the scan cell will operate in functional mode. To configure a row in FD mode (row 3 in Fig. 3.2), the RCL block sets FD_L to ‘1’ and ModePrev to ‘0’ which in turn stitches master latch output, QM_Prev, into the scan cell. For a split mode

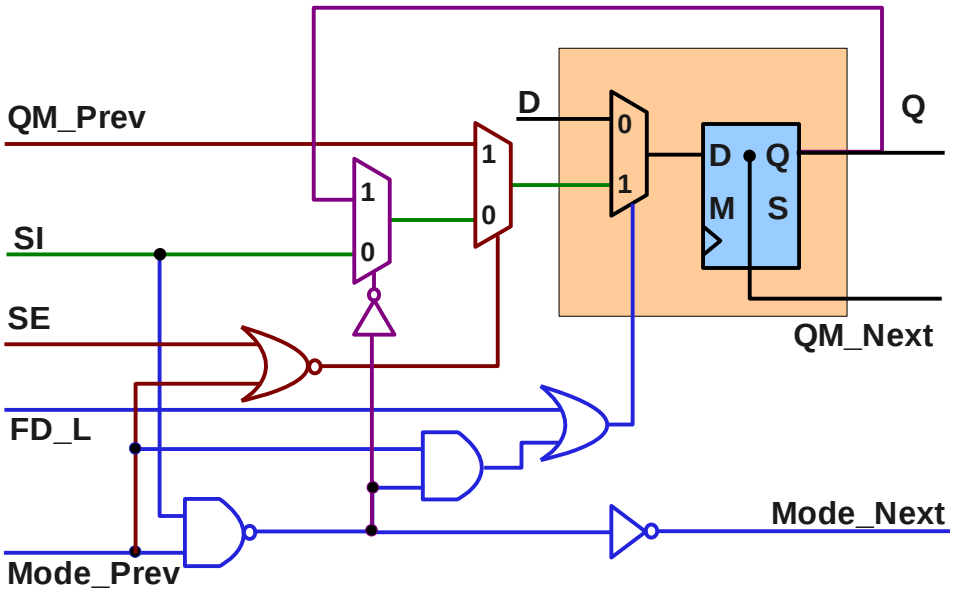


Fig. 3.7: REBEL 'Front-End' Logic for MUX-D Scan Cell

row (row 2 in Fig. 3.2), the same scan FF encoding method described for CLSSD is used. In addition, the RCL block forces a ‘1’ onto FD_L and sets ModePrev to a ‘1’ for insertion points other than the left-most scan FF in the row or ‘0’ otherwise. The annotation in Fig. 3.7 shows the values of the scan FF at the point of PUT insertion for a split mode row. A 2-to-1 MUX (shown in red color in Fig. 3.7) is added to preserve encoding values in the scan FFs to the left of insertion point. Output of the scan FF, Q, is feed back through the MUX when there is a '1' in SI and Mode_Prev to preserve encoding.

The RCL block for MUX scan is similar in function to the CLSSD version, but a lot simpler as all logic in reference to SCA_L and SCB_L of Fig. 3.3 can be eliminated. Fig. 3.8 shows the schematic diagram of the RCL block. As mentioned earlier, REBEL implementation for Mux-D style scan chain requires an additional primary input (FD) (as shown in Fig. 3.8) for creating the combinational delay chain. The shift registers in RCL,

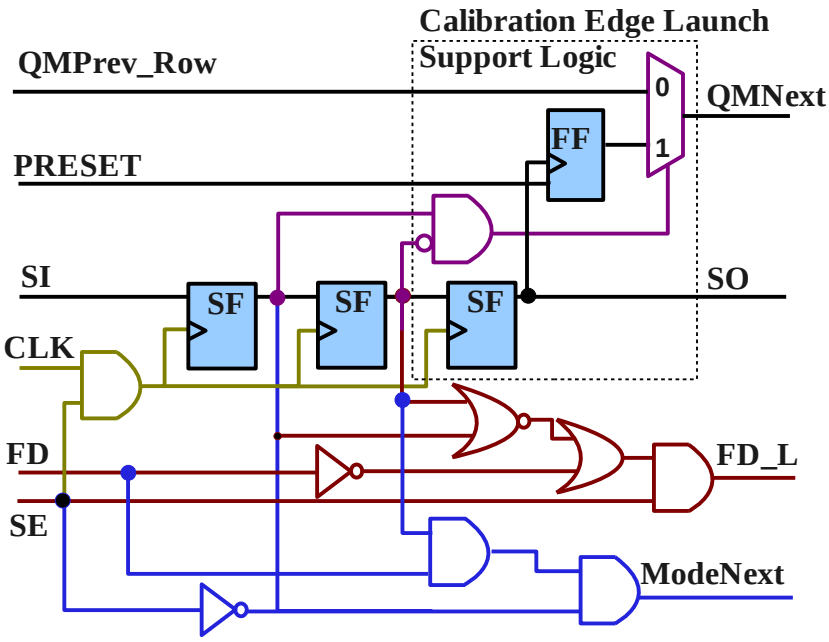


Fig. 3.8: Row Control Logic for MUX-D Style Scan Chain

that are conditionally inserted into the scan chain, can be configured prior to REBEL test, which will in turn configure the scan cells in that segment into different modes. Clock to these registers are gated by an AND gate, that is controlled by scan enable primary input, SE (which is '0' during the REBEL mode and '1' during the shift mode), so that it will be part of scan chain during scan operation and preserve the configuration information during REBEL test.

The state values stored in the two shift registers (out of three) along with the global signal 'FD' controls the operation of each segment/row of the scan chain. The third shift register and the components shown in the dotted box (top of Fig. 3.8) are used for generating on-chip calibration edges for REBEL calibration process. Table 3.2 gives the encoding of the state elements in the RCL block that implement various operating mode of the scan segments.

State Values	Functionality
000	All scan FFs in row are in functional mode
001	All scan FFs in row are in Flush-delay mode
11X	Mixed Mode, Left scan FFs in functional mode, right scan FFs in FD Mode
011/010	Calibration Mode, All the FFs are put in FD mode and rising/falling edge is launched from the RCL for calibration

Table 3.2: Configuration States for Row Control Logic – MUX-D Scan

State values '000' on shift registers forces all the scan cells in that row into functional mode by placing '0' on both FD_L and Mode_Next wires. State '001' forces the scan cells into flush-delay mode by placing '1' on FD_L and '0' on ModeNext. State '11X' configures a mixed mode, all scan cells before insertion point into functional mode and after the insertion point into flush-delay mode, and connects a specific PUT output into the scan chain based on states of scan cells in that row as in CLSSD based design. The state values '011' and '010' are used to launch rising and falling calibration edges respectively.

The third shift register in this RCL design is added for generating on-chip calibration edges. This register along with the components shown in the dotted box (refer Fig. 3.8) is used for calibration process. When a '1' is stored in this register creates a rising edge for calibrating that specific segment and a '0' creates a falling edge. This on-chip calibration edge generation reduces the uncertainty in the calibration data, which in turn reduces the measurement error, and reduces the calibration time by allowing parallelism in calibration process.

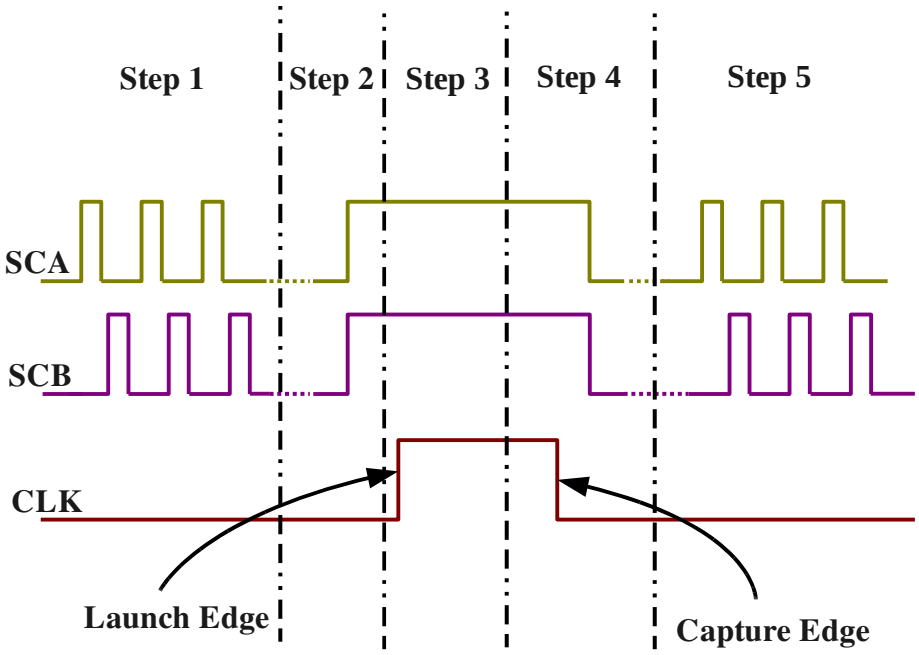


Fig. 3.9: Timing Diagram of REBEL Test for CLSSD Scan Design

3.5 PUT Delay Analysis Process

The Launch/Capture delay in REBEL is controlled by CLK, as described earlier, and therefore REBEL leverages the clock tree for critical timing events.

A REBEL test for CLSSD style implementation is carried out as follows:

1. Configuration data is scanned in.
2. The global SCA and SCB signals are asserted.
3. CLK is asserted to launch a transition into the PUT.
4. CLK is de-asserted after a specific Δt, sufficiently long to allow the transitions on the PUT to propagate along the scan chain.
5. The global SCA/SCB signals are de-asserted, and the values in the scan chain are scanned out.

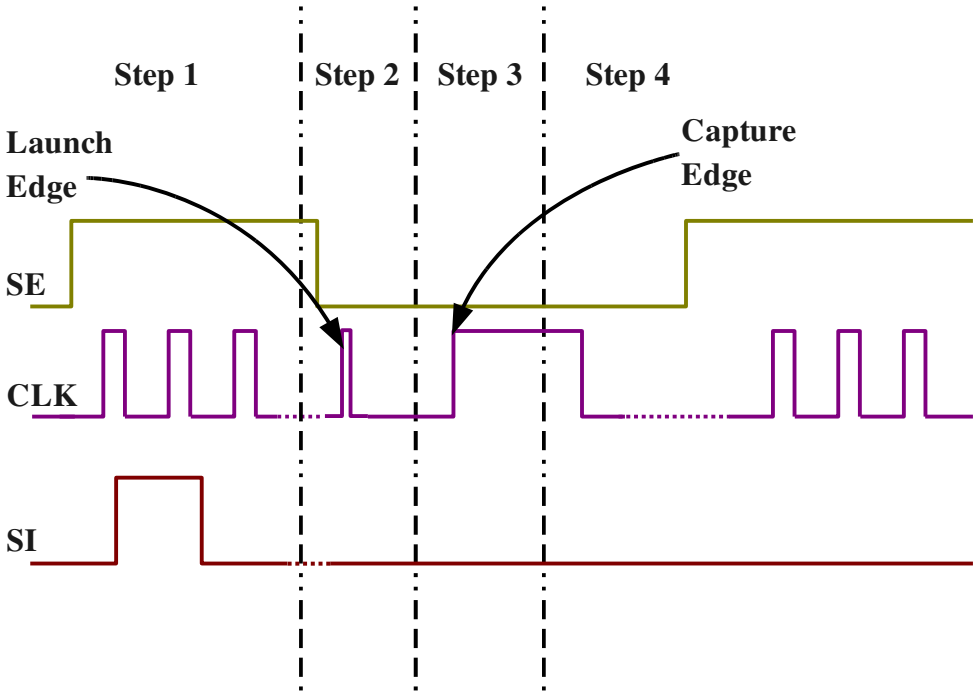


Fig. 3.10: Timing Diagram of REBEL Test for Mux-D Scan Design

REBEL test for MUX-D style implementation is carried out as follows:

1. SE is enabled and configuration data is scanned in.
2. SE is disabled. CLK is asserted which launches the transition into PUT. CLK is de-asserted as quickly as possible (this requires fast tester channels).
3. CLK is asserted again after a specific to Δt, sufficiently long to allow the transitions on the PUT to propagate along the scan chain. This captures the snap-shot in the scan chain.
4. SE is enabled and the values in the scan chain are scanned out.

The delay in the combinational path is computed using eqn. 3.1

$$T_{path} = T_{lc} - T_{sc} \qquad \text{Eqn. 3.1}$$

where, Tpath = Delay in the combinational path

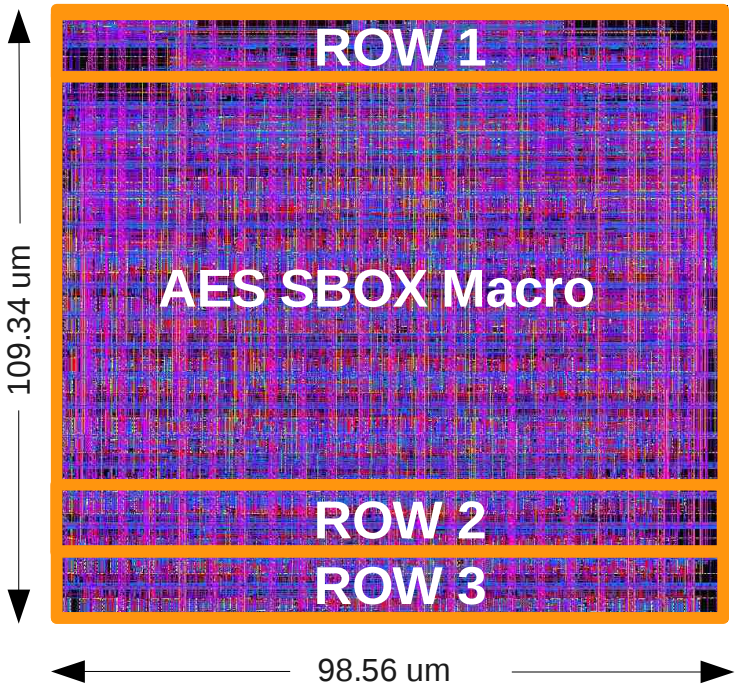


Fig. 3.11: REBEL Layout for AES SBOX

T_{lc} = Launch/Capture Delay

T_{sc} = Delay in the Scan Chain

The scan chain delay, T_{sc}, can be calculated from the number of scan cells that are set by the propagating edge(s), and the data obtained from a set of calibration tests (described in Section 3.6.1).

3.6 Simulation Results

In order to validate REBEL ETS structure, a set of simulation experiments on RC-transistor models of several layouts has been carried out. Fig. 3.11 depicts the layout used in the REBEL simulations. Cadence Encounter Place and Route tool is used to synthesize the layout of an AES SBOX macro1, with the REBEL ETS embedded.

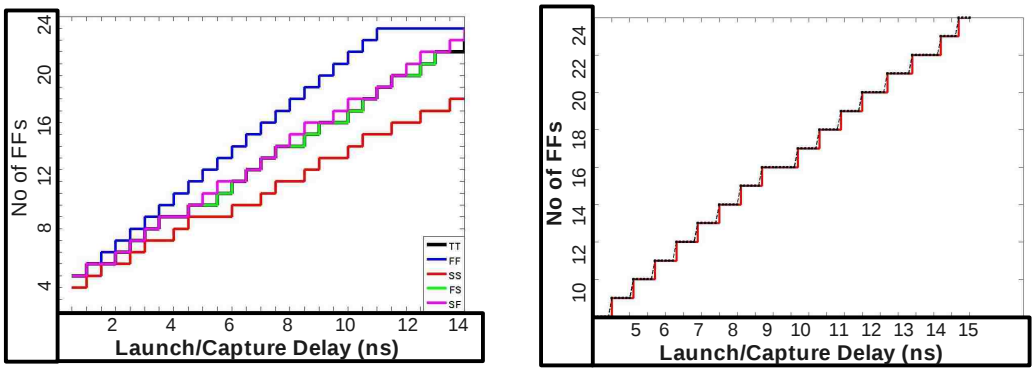
A set of five simulation models, one Nominal (TT) and four process corner models, identified as fast-NMOS, fast-PMOS (FF), slow-NMOS, slow-PMOS (SS), and two mixed variations given as FS, SF, has been created. Also, a separate calibration process was carried out on each ETS to enable the translation of the digital scan data to actual Δt's. The calibration processes also reduce the adverse effects of within-die and die-to-die variations that occur within the REBEL structure themselves.

3.6.1 Calibration

The calibration process for REBEL is designed to enable the delay of a transition propagating along the scan chain to be eliminated from the timing measurement, as specified by Tsc in eqn. 3.1. Once eliminated, the delay of the PUT can be determined.

The calibration process involves placing the entire scan chain into flush delay mode and carrying out a sequence launch/capture experiments. In each successive experiment, the capture timing is increased by a small Δt and the digital values stored in the scan chain are analyzed. This Δt also defines the resolution of the delay measurement along with the process variations. For some of the experiments, the edge is able to propagate to the next scan FF in the chain. The entire sequence of experiments yields results that indicate when this occurs for each scan FF in the chain at the level of timing resolution given by the Δt step size. The Δt value is restricted to the step size to 100 ps in this experiments, although higher resolutions may be attainable in practice.

The graph shown in Fig. 3.12(a) gives the results from applying calibration tests in this fashion to each of the five process models. The step-wise nature of the curves reflects the delay through each of the scan FFs, which is approx. 550 ps. These curves can



(a) (b)

Fig. 3.12: Calibration Data for AES SBOX Macro (a)Low Resolution in Different Process Corners (b) High Resolution for TT Corner

be used to derive PUT delays in the macro. Assume, for example, the PUT is inserted into the scan chain at the 12th scan FF, and the scan chain values indicate that the edge propagated to the 19th scan FF. To compute the path delay, the LC delays for the 12th and 19th scan FF are looked-up on the x-axis using the appropriate curve in Fig. 3.12, and then subtracted. The Δt computed is the delay through that portion of the scan chain. The LC delay to the 19th scan FF (Tlc) and the scan chain delay (Tsc) can be plugged into Eq. 1 to determine the delay of the PUT. Note that the resolution obtained for the PUT delay is limited to the typical propagation delay through a scan FF (approx. 550 ps). However, as discussed later in Section 3.6.2 , the strobing technique described for calibration can also be used in the REBEL tests to improve the timing resolution to 100 ps or less.

3.6.2 REBEL Path Delay Analysis

To evaluate the effectiveness of REBEL for measuring path delays, a set of test vectors that drive transitions through the AES SBOX macro has been derived and analyzed the path delays along seven different paths.

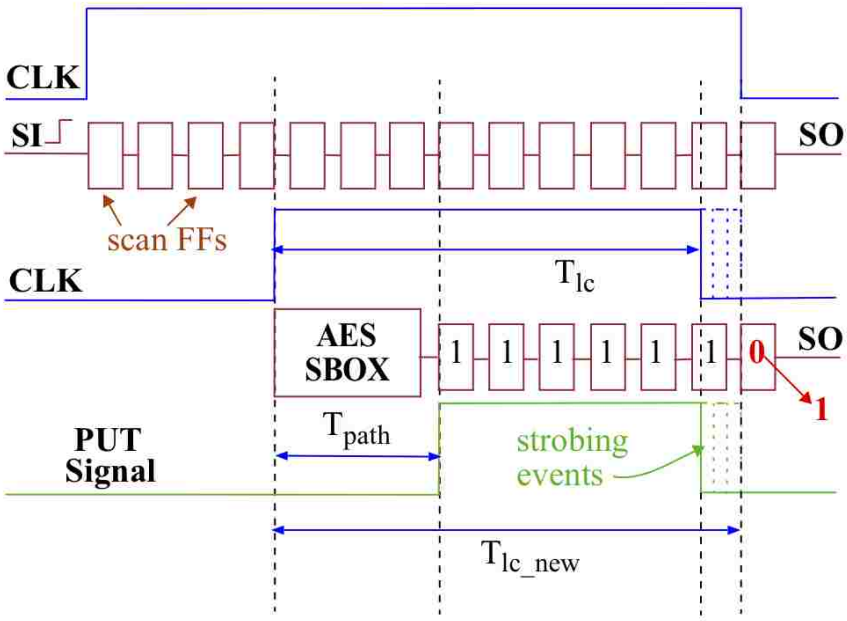


Fig. 3.13: Increasing Measurement Accuracy Through Clock Strobing.

As indicated earlier, the timing resolution of REBEL will be limited to the step size delay of the scan FF (approx. 550 ps) unless a strobing technique is applied in the REBEL tests as well. The timing diagram given in Fig. 3.13 illustrates the concept of strobing as a means of improving timing resolution. The top row of scan FFs depicts the entire scan chain which is tested and characterized during the calibration process. The bottom section shows the AES SBOX and a portion of the same scan chain that connects to its outputs.

The vertical dotted lines on the right side of the diagram illustrate a series of high resolution (100 ps) strobing events, with those applied for calibration shown along the top and those for a REBEL test along the bottom. The goal during strobing is to determine at which 100 ps interval does the next scan FF get set with the propagating transition. This process effectively divides the entire Δt through each scan FF into smaller pieces. The LC interval determined in this way, i.e., that just sets the next scan FF, is used in eqn. 3.1 to

compute the PUT's delay.

In Fig. 3.13, assume a REBEL test is measuring a rising transition on a path through the SBOX macro, and the initial interval is given by T_{lc}. The scan chain results obtained for this test are shown as a sequence of six '1's and one '0'. A sequence of strobes 100 ps apart is now applied and during the last application, the results change to a sequence of seven '1'. This new LC interval, labeled T_{lc_new}, is the target value used in the PUT delay calculation.

Test #	Edge	Digital Snap-Shot	T_{lc_new} (ns)	T_{sc} (ns)	T_{path} (ns)
1	R	000000001101111111100000	5.6	3.5	2.1
2	R	000000001110111111100000	5.5	2.9	2.6
3	F	000000000000000001111111	5.5	3.5	2.0
4	F	000000001111000000001111	5.4	2.9	2.5
5	R	000000001011111110000000	5.1	2.9	2.2
6	F	000000000000000011111111	5.5	3.1	2.4
7	F	000000001111100000000111	5.1	3.0	2.1

Table 3.3: Delay Measured in Different Combinational Paths in AES SBOX Macro

$$Percentage\ Error = (T_{rebel} - T_{actual}) * 100 / T_{actual} \qquad \text{Eqn. 3.2}$$

The application has been simulated with seven transition tests applied to the AES SBOX. Table 3.3 identifies the test number in the first column, whether the signal emerging from the SBOX output is a rising or falling edge in the second column, the scan chain values in the third column, the final LC time interval in the fourth column (after strobing), the computed delay along the scan path in the fifth column and the computed delay of the PUT in the last column. Note that the first eight bits of scan chain are always '0' because

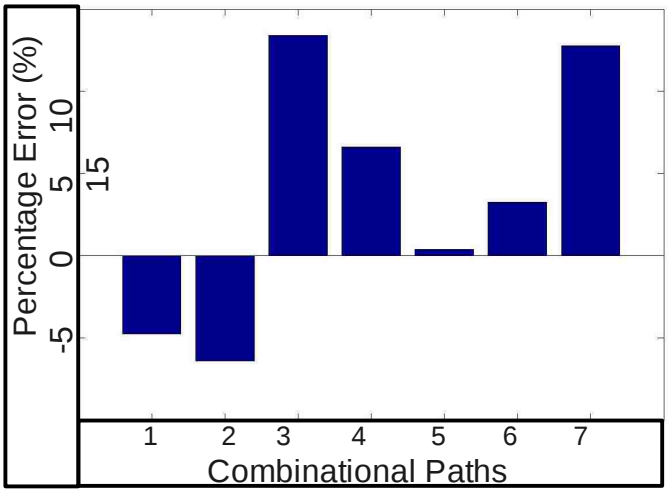


Fig. 3.14: Percentage Error in Delays using REBEL compared with the Actual Delays in AES SBOX macro.

these scan FFs are used to launch the transition into the SBOX. Fig. 3.14 shows the percentage of error (computed using eqn. 3.2) in the delays derived using REBEL, in comparison to the actual delays. The error ranges from -4% to 13%.

3.6.3 Defect Detection Analysis Using REBEL

REBEL is also applicable to the problem of detecting delay defects. To show this, delay defects are emulated by adding various amounts of capacitance to selected paths in the SBOX macro. Transition tests are derived that propagate transitions along the defect path, referred to as P1, and along a second defect-free path, called P2. REBEL is used to measure these path delays, and the results are used as input to a detection process that compares the relative magnitudes of the two path delays.

In addition to the delay defect simulations, a set of defect free simulations are carried out on each of the five process models to determine the expected behavior of P1 versus P2 which is going to be used as a delay signature of defect-free design. For the

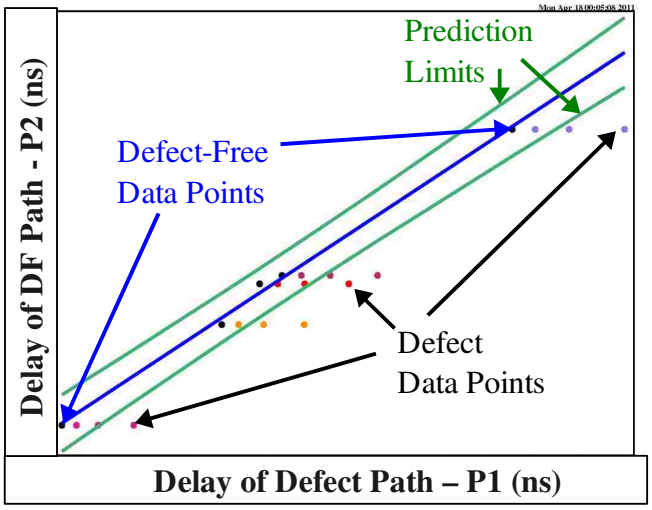


Fig. 3.15: Analysis of Delay Defects in the SBOX macro with the Delays Measured Using REBEL Test Structures

delay defect simulations, the additional capacitive loads introduced on path P1 are 10 fF, 25 fF and 50 fF. The path delays of P1 (x-axis) and P2 (y-axis) are plotted in Fig. 3.15. Regression-based prediction limits are derived using the defect-free data. Data points outside these prediction limits are considered positive detection. The delay defect data points in Fig. 3.15 corresponding to the 10 fF capacitive loads are not detected in any of the process models. However, half of the 25 fF and all of the 50 fF capacitive loads are detected.

3.7 Test Chip Design

In order to verify the capabilities of the proposed embedded test structure in silicon, a new test chip is designed in IBM 90nm CMOS bulk technology that incorporates REBEL in two macros. One of the two macros is an Advanced Encryption Standard (AES), a very commonly used encryption engine. I chose this macro because it is vulnerable to adversarial attacks in which an adversary can introduce a Trojan circuit

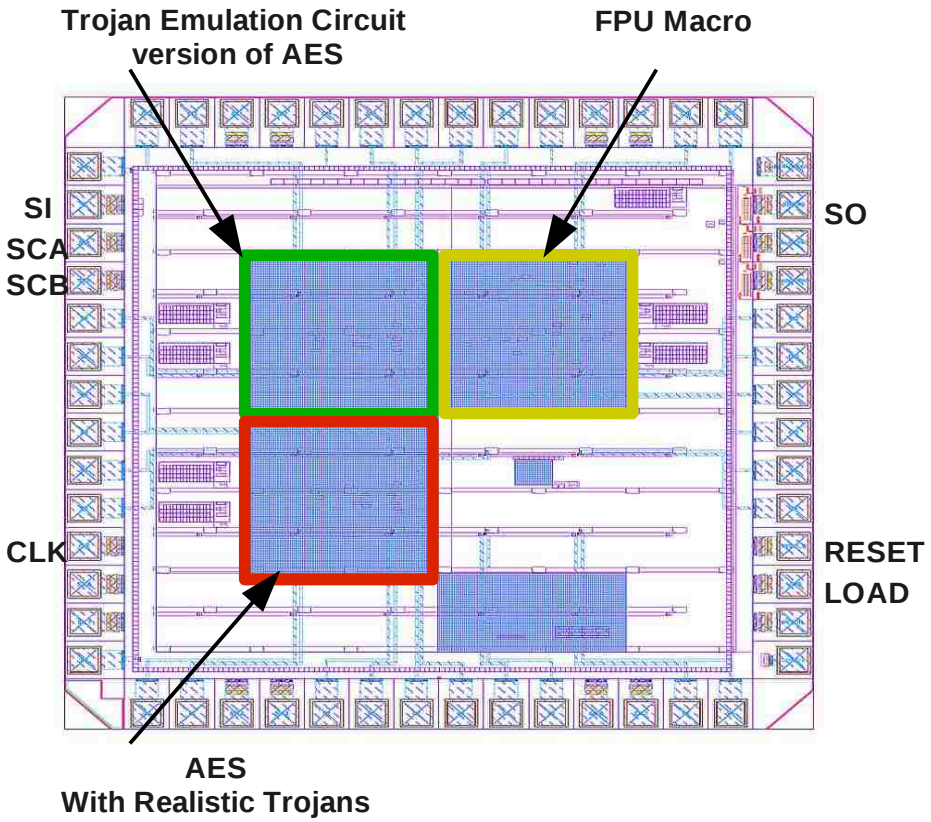


Fig. 3.16: Top Level Chip Layout

that is designed to leak plain text or the encryption key. The second macro is a 32-bit pipelined Floating Point Unit (FPU) that we obtained from colleagues who managed the DARPA IC Trust program. A layout view of these macros is shown in Fig 3.16.

The flowchart in Fig. 3.17 shows the design automation process followed for the test chip design. The behavioral HDL descriptions of AES and FPU macros are first synthesized using Cadence RTL Compiler [46]. The macros are synthesized using a scan insertion DFT flow and the design is converted into a full-scan design, i.e., all the flip-flops are replaced by the scan cells. Then, the REBEL test structure is integrated into this scannable design using an in-house PERL script. As explained in the previous chapter, the scan chain is partitioned into several segments and RCL logic is added to control the

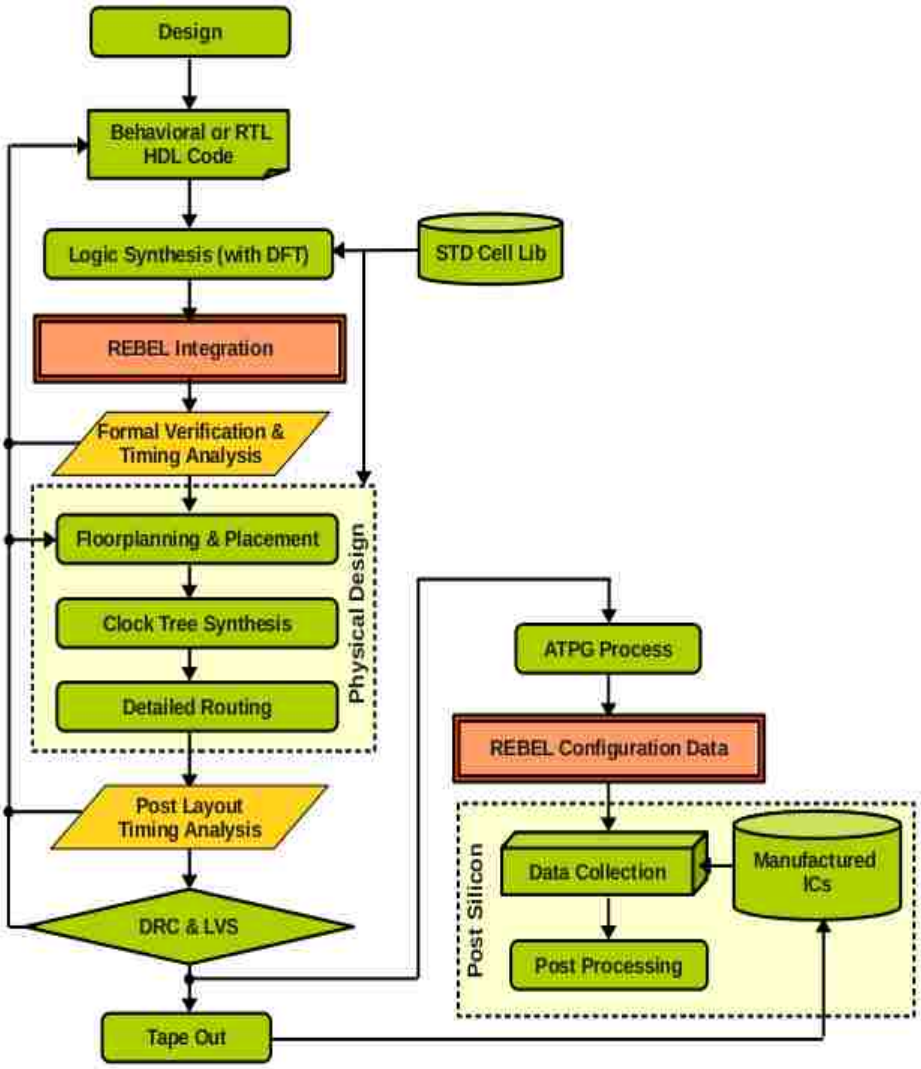


Fig. 3.17: REBEL Integrated Test Chip Design Automation Flow

operation of each segment. Also, the scan cells are replaced with the REBEL-scan cell, which is a scan cell with the 'front-end' logic (explained in section 3.3).

Functional verification is carried out on the REBEL-integrated design to verify that the functionality of the design is preserved. Using Cadence Encounter Place and Route tool [46] the GDSII is generated from the REBEL-integrated structural netlist. The GDSII file is imported into Cadence Virtuoso, and design rule check (DRC) and layout vs schematic (LVS) checks are carried out. The layouts of REBEL-integrated macros are

assembled at the top level chip design manually. The DRC and LVS clean design is then tapped out.

Cadence Encounter Test ATPG tool is used to generate transition/path delay test vectors for various path delay measurements. The output of the ATPG tool (STIL files) are then processed by a PERL script in order to add the REBEL configuration information. The script also converts the test vectors into a tester compatible form. Several path delays that represent various path lengths of the design are measured using the proposed test structure. These path delays are measured in 62 copies of the chip and analyzed for die-to-die and within-die variations.

In these macros several Trojan emulation circuits, that are designed to introduce delay anomalies along the selected paths, are inserted. A control mechanism is incorporated in the Trojan Emulation circuit that enables the controlled insertion of additional capacitive load that models a Trojan. With this mechanism, I intend to derive and apply test vectors that are designed to sensitize transitions along paths that have Trojans inserted, and those that do not. I used the REBEL infrastructure to measure the delays along these paths, which are used in post processing analyses to determine if the Trojan delay anomaly can be detected (explained in Chapter 6).

We received 62 copies of the fabricated chip manufactured in 90nm CMOS technology. A structural tester, the Ocelot ZFP structural Test System donated by Verigy Inc., is used to apply the test vectors and to store the responses. The path delays measured in these 62 chips are used to present REBEL's ability to measure variability in path delays (Chapter 4) and to detect hardware Trojans (Chapter 6).

3.7.1 Area Overhead Analysis

As explained earlier, REBEL is first integrated with a pipelined FPU which has a scan chain length of 671 bits. The scan chain is divided into 28 segments as a strategy to enhance the coverage achievable for path delay Automatic Test Pattern Generation (ATPG). An RCL block is added to control each of these segments as explained in Section 3.3 . REBEL is also integrated with another macro AES encryption engine with a scan chain length of 530 bits that is partitioned into 18 segments.

Macro	Area(μm^2)	No of Scan Cells	No of Segments	Area Overhead
FPU	251763.09	761	28	11.45%
AES	251763.09	530	18	7.7%

Table 3.4: Area Overhead of REBEL

Table 3.4 gives the area overhead associated with REBEL with respect to the logic needed to implement the FPU and in AES macro that is also included on the chip. The area overhead of REBEL depends on: 1) the amount of sequential logic in the design, and 2) the number of segments the scan chain is divided into. From the table, it is clear that the amount of sequential logic and the corresponding number of segments is larger in FPU than in AES which results in higher overhead. A closed form expression for estimating the amount of overhead associated with REBEL is given as follows:

$$REBEL_{Overhead} = N_{sc} * A_{CTRL} + N_{seg} * A_{RCL} \quad \text{Eqn. 3.3}$$

where, N_{SC} = Number of Scan Cells,

A_{CTRL} = Area of additional logic added to scan cell,

N_{seg} = Number of segments,

A_{RCL} = Area of Row Control Logic.

The first term in Eqn. 3.3 is the area overhead introduced by the modifications required to the scan cells. The second term in the Eqn. 3.3 is the area overhead introduced by the RCL logic which depends on the number of segments that are created in the scan chain. The numbers presented in Table 3.4 are the worst case scenario, i.e. the scan chain is divided into several segments to achieve maximum coverage. However, the overhead can be reduced by reducing the number of segments, which in turn may result in reduced coverage. Therefore, the overhead associated with the RCL blocks can be reduced at the expense of reducing path coverage.

CHAPTER 4

REBEL for Delay Variability Characterization

As explained in the previous chapter, REBEL can measure the path delays accurately and therefore it can be used to analyze random and systematic variations in these path delays. In this chapter, I present REBEL's ability to characterize delay variability, in particular, measuring the magnitude of die-to-die and within-die variations. As explained in Chapter 3, REBEL captures the delay behavior in design-context and therefore it can provide accurate characterization of delay variability. Moreover, REBEL can measure delay of any path length (short or long) using at-speed clock. This method measures realistic delay values, as opposed to the techniques that use multiple faster-than-at-speed clocks to measure short path delays causing pessimistic delay value measurements. In addition, the path delays measured using REBEL can be used for identifying systematic variations caused by any particular gate, layout style, etc. using data mining algorithms.

In this chapter, I present the characterization of die-to-die and within-die variations using the hardware data collected from 90nm ASICs and 130nm FPGAs. First, the CLSSD-based REBEL structure, that is integrated into a 32-bit pipelined Floating Point Unit (FPU) and implemented in IBM 90nm CMOS bulk technology, is used to analyze the delay variability in ASICs. Second, the MUX-D based REBEL structure, that is integrated with the SBOX component of the Advanced Encryption Engine (AES)

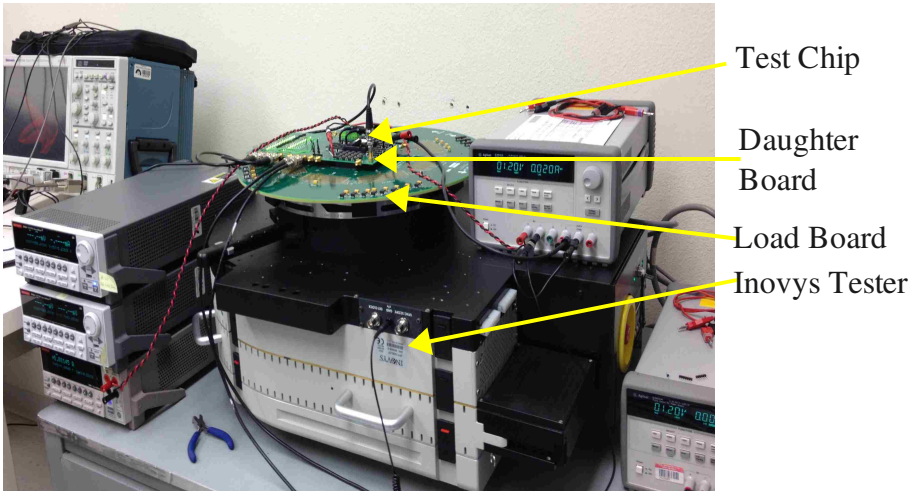


Fig. 4.1: Experimental Setup for ASIC Data Collection

algorithm and implemented in 130nm FPGAs (Virtex2Pro), is used to characterize delay variations in FPGAs. Finally, REBEL is integrated with a built-in-self-test (BIST) architecture, which is used to measure the path delays of a pipelined AES macro, and implemented on FPGAs to characterize delay variability. I also show the improvement in measurement quality while using on-chip clock generation scheme.

4.1 ASIC Delay Variability Characterization

As explained in section 3.7, CLSSD based REBEL structure is integrated into a pipelined floating point unit and implemented in IBM 90nm CMOS bulk technology using the design flow explained in Fig. 3.17. The final design is tapped out to IBM through MOSIS and the received chips are first tested for the functionality of REBEL and later path delay measurements are collected from 62 copies of the chip.

4.1.1 Experimental Design

Fig. 4.1 shows the instrumentation that we used for the REBEL tests. An Inovys

structural tester, donated by Verigy Inc., is used to apply the test vectors and to capture the responses of the delay tests. The High Performance Clock Channels (HPCC) in the tester, that can generate a clock of period up to 635 ns with a launch/capture resolution of 15 ps, are used to generate precisely timed launch/capture edges. An on board line receiver is used to convert the differential signal into a single ended clock at a point very close to the zero-insertion-force socket for the chip. The transition fault/path delay test vectors used in these experiments are created using the Cadence Encounter Test ATPG tool.

4.1.2 Experimental Results

In this section, I present the die-to-die and within-die variations in path delays of FPU macro measured using REBEL on a set of 62 chips. The first step of the process involves applying a set of calibration tests to the FPU macro to characterize the delay along its scan chain segments. The calibration data is then used to compute the actual path delays from a subsequent set of transition tests applied to the core logic. The path delays measured in the set of 62 chips are then used to analyze the process variations in path delays.

4.1.2.1 Calibration

In section 3.6.1 , I proposed a calibration process in which all the scan cells in the scan chain are put in flush-delay mode, an edge is launched from the 'SI' pin and a sequence of launch-capture tests are carried out. However, for the designs that have longer scan chains the uncertainty in calibration data is higher for the scan cells that are at

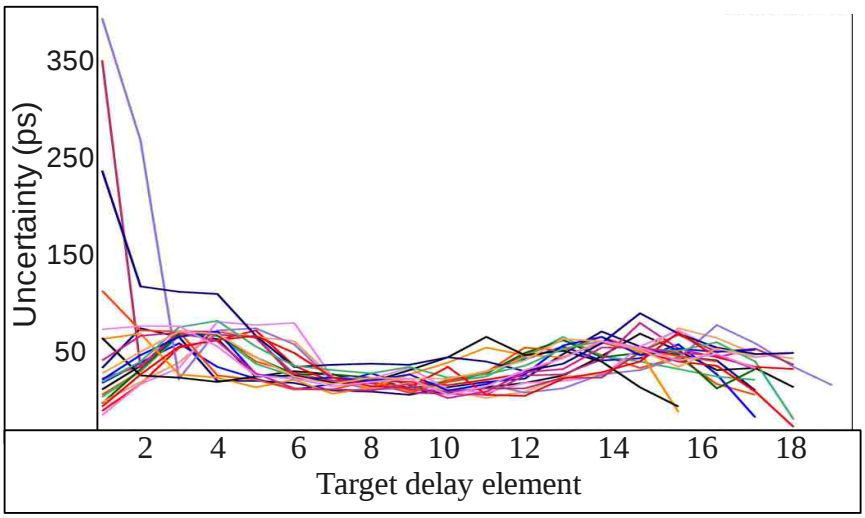


Fig. 4.2: Uncertainty at Various Target Flip-flops

the end of the chain. From the preliminary experiments, I determined that the uncertainty in the delays measured in the calibration process grows as the edge propagates further into the chain. This uncertainty results in increased path delay measurement error. In order to reduce the uncertainty in the calibration data and hence the measurement error, I created a special set of tests that drives a transitions into first (left-most) flip-flop of each segment. This technique reduces the uncertainty in the delays as each segment of the scan chain are calibrated using the edge derived from the nearby logic. Moreover, these special tests can be created using the ATPG tools (in this experiment Cadence Encounter Test) and does not require any additional logic. The different velocities associated with the propagation of rising and falling edge require both rising and falling edge calibration tests.

4.1.2.2 Noise Analysis

As described earlier, clock strobing is applied using incrementally longer LC intervals to determine the delays between elements of the delay chain (referred to as

inter-scan-FF delays earlier). The LC interval in which a delay element is first able to record the transition is the interval used to determine the inter-scan-FF delay. However, this LC interval may vary from one sample to the next. The range over which it varies is referred to as *uncertainty*. Uncertainty reflects the noise level present in the measurements.

In our experiments, we found that the level of uncertainty varies for delay elements at different positions from the PUT's insertion point. Given that REBEL allows a 'target' delay element to be chosen, this characteristic of uncertainty can be leveraged as a means of improving resolution. Fig. 4.2 plots uncertainties along the y-axis for each of the numbered delay elements along the x-axis. The delay elements beginning from the PUT's insertion point (the capture FF) are numbered 1 to 18 for each of the consecutive elements along the delay chain. Each of the curves represents the set of uncertainties associated with one PUT. The graph plots the results for 23 PUTs.

The wide range of uncertainties on the left side of the graph indicate that delay elements near the insertion point work well as a target for some PUTs but very poorly for others. This occurs because the voltage behavior on the outputs of some PUTs glitch, making it difficult to obtain a consistent measurement. On the other hand, the uncertainties for target delay elements in the center region of the graph, e.g., delay elements numbered 8 and 9, provide the lowest overall uncertainty across the PUTs. Uncertainties again increase for delay elements at larger distances from the insertion point because of power supply noise, signal coupling and other types of on-chip noise. Given these characteristics, we use the 9^{th} delay element to compute path delays.

In order to evaluate the uncertainty in the delay measurement, the path delay

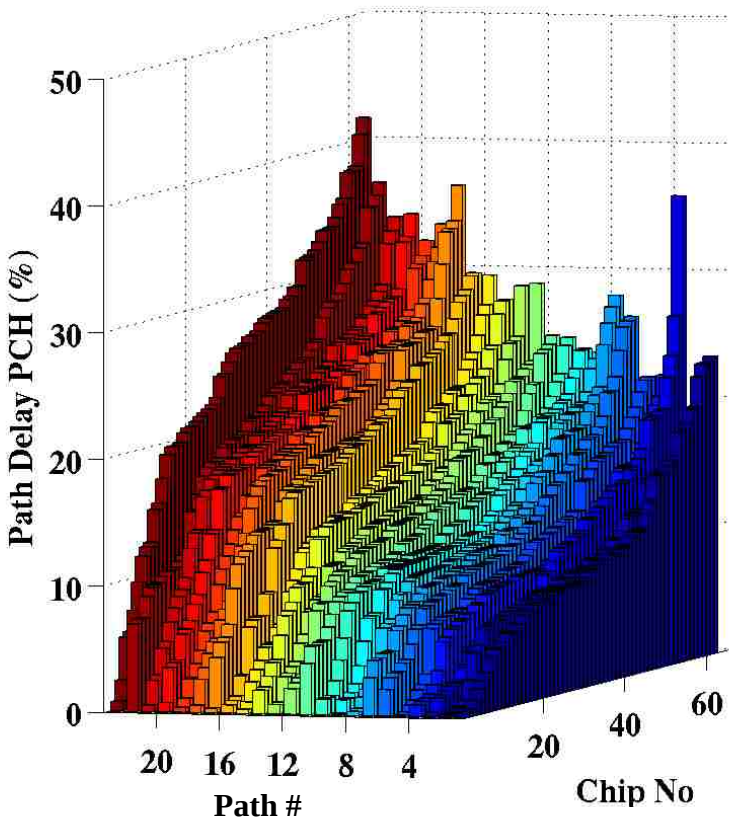


Fig. 4.3: Delay Variations in 23 Representative Paths among 62 copies of the chip

measurements are repeated for 10 times and the standard deviation among these 10 samples are computed. The path delay measurement is carried out for 23 different paths on 62 chips. The average of these standard deviation computed from all path delay measurements in our experiments is approx. 40 ps (3σ of 120 ps), and the worst case standard deviation is approx. 150 ps (3σ of 450 ps). The main sources of this uncertainty are meta-stability in the delay elements, power supply noise, clock glitter and temperature variations. The worst case uncertainty may be smaller for designs that generate the LC timing events on-chip.

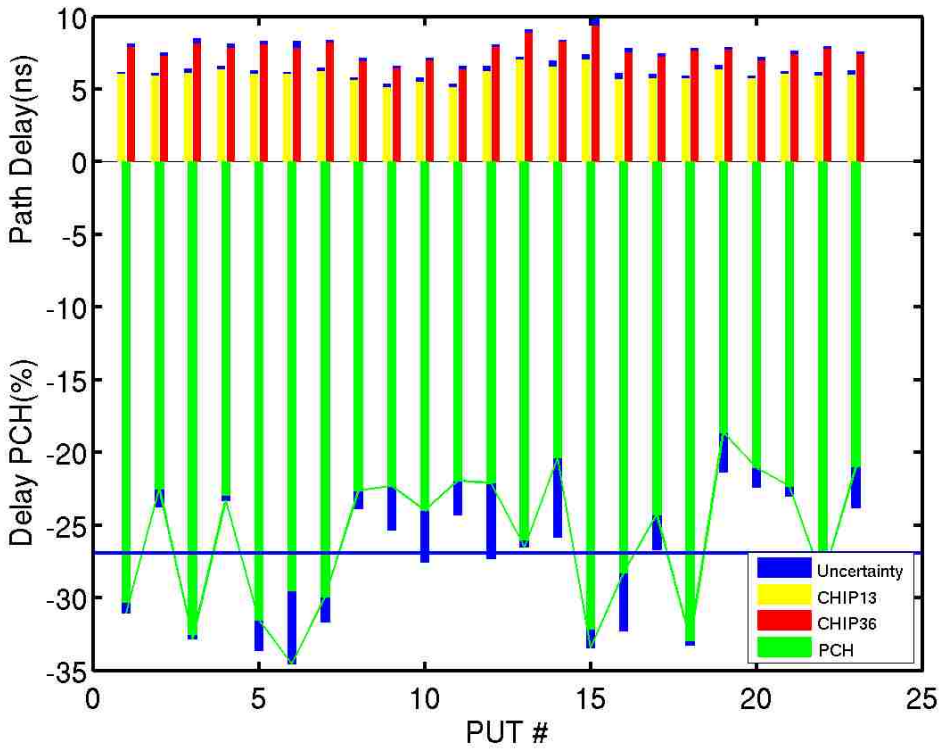


Fig. 4.4: Die-to-Die and Within-Die Variations in 90nm ASIC

4.1.2.3 Die-to-Die and Within-Die Variation Analysis

Fig. 4.3 shows the path delay variations in 23 different paths of FPU macro among 62 chips (sorted along y-axis). The different paths are plotted along x-axis, chip numbers along the y-axis and normalized path delays as percentage change along z-axis. The path delays are normalized with respect to the fastest chip among the population of 62. The die-to-die delay variations are clear along the y-axis and the magnitude ranges up to 40% with respect to the fastest chip. Remember that the uncertainties, caused by the clock jitter, glitches, etc., are included in this data which results in high die-to-die variations. Also, the uncertainty is doubled (uncertainty in measured LC period and calibration data) when the delay along the scan chain is subtracted from the LC period.

Fig. 4.4 portrays the die-to-die and within-die variations more clearly. In Fig. 4.4,

23 different path delays (path and scan delays) of slowest and fastest chip are plotted (top). The percentage change (PCH) in delays with respect to the fastest chip are also plotted (bottom). These PCH values depicts the variations in path delays among these two chips (CHIP 13 and CHIP34). The average of these PCH values is shown as a horizontal line in the figure and it explains the magnitude of die-to-die variations, which is about 27%. The deviations in these PCH values explains the within-die variations (if there is no within-die variations all the percentage values should be equal). The PCH values varies from 22% to 34% showing a 12% within-die variation.

The delay values plotted in this figure also shows the amount of uncertainty (blue tips in the bars) in the measurement among 10 samples. The PCH values also shows the amount of percentage change caused by these uncertainties. In the worst case, about 5% of the change is caused by the uncertainty in the measurement and when the actual path delay values are used it can go up to 10%. This explains the need for reducing the uncertainty in the measurement in order to achieve accurate variation characterization. On-chip clocking schemes can reduce the uncertainty caused by clock jitter significantly. In section 4.3, I will show the improvement in the uncertainty while using the on-chip clocking schemes. The uncertainty can also be reduced reasonably by measuring the path delays in a temperature controlled environment, which eliminates the variations caused by the environmental temperature change.

4.2 FPGA Delay Variability Characterization

Since REBEL test structure is well integrated with any macro and the integration can be done at the HDL level, it can be used for delay variability characterization of

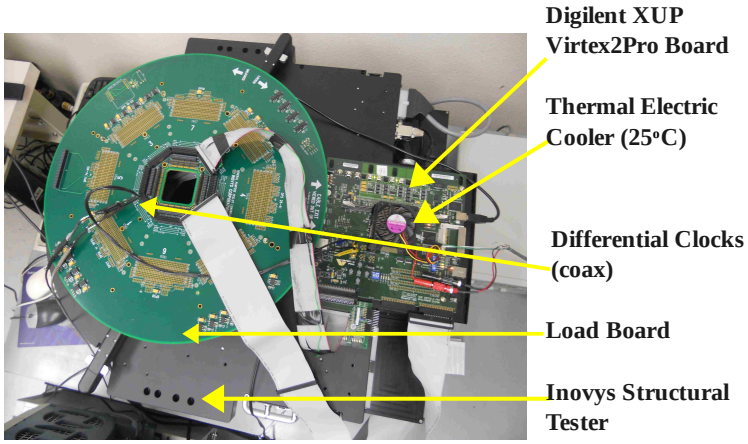


Fig. 4.5: FPGA Experimental Setup

FPGAs also. In this section, I present die-to-die and within-die variation characterization of Xilinx Virtex2Pro FPGAs using a MUX-D based REBEL integrated with a SBOX component of the Advance Encryption Standard (AES) engine.

4.2.1 Experimental Design

In order to explain the capabilities of REBEL, it is integrated into a VHDL description of the SBOX component of the AES algorithm [47], and implemented in Xilinx Virtex2Pro FPGAs which are fabricated in 130nm technology. Although this is an older technology, REBEL is scalable to advanced technologies and the default resolution (with out 'clock strobing' mechanism) of REBEL increases with the decrease in the feature size.

Fig. 4.5 shows the instrumentation used for this experiment. The structural tester, an Inovys Ocelot ZFP, is used to apply the test vectors and to store the test results of the REBEL tests. High speed differential clocks generated in the High Performance Clock Channels (HPCC) of the tester (refer to Fig. 4.5) are used to generate high-speed launch

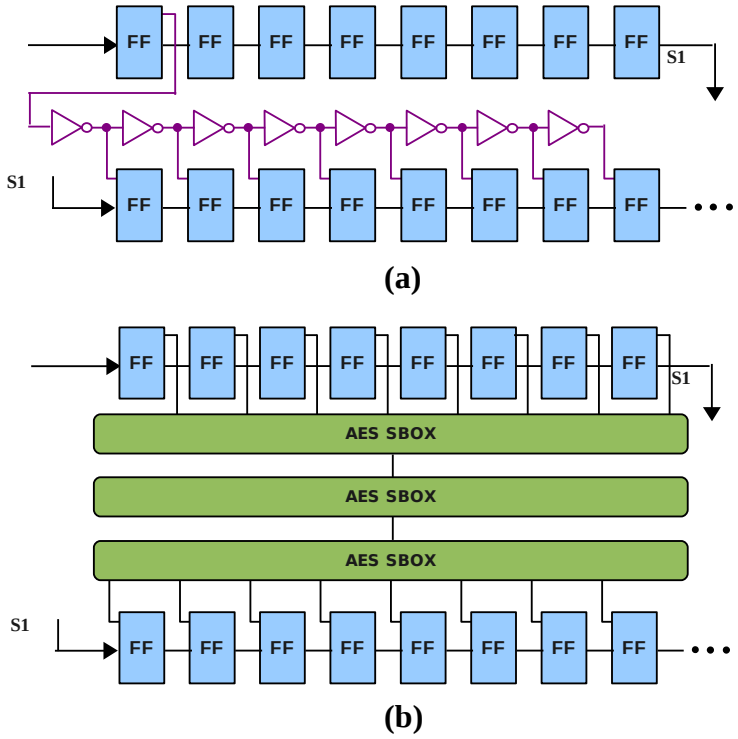


Fig. 4.6: Experimental Design – (a) Inverter Chain (b) AES SBOX

and capture edges. The Xilinx Virtex2Pro FPGAs have internal buffers that creates a fast single-ended clock from the differential clocks and connects to the global clock tree of the FPGA. In order to eliminate the variations due to temperature during the delay measurement, a thermoelectric cooler is used to maintain a constant temperature (25^0C).

As shown in Fig. 4.6(a), the REBEL is first integrated with a simple inverter chain and the output of each inverter is connected to the inputs of the registers in the next segment. This eventually creates 8 different combinational path with different number of inverters. The path delays in these 8 paths are measured using MUX-D based REBEL to analyze the variations in the small combinational paths. Then, as shown in Fig. 4.6(b), the integration of REBEL with a more complex macro, AES SBOX, to show the various capabilities of REBEL including path delay variation measurement and as an on-chip logic analyzer.

4.2.2 Experimental Results

This section explains the capabilities of REBEL with the hardware experimental results. As explained in the previous section, the REBEL integrated macros, inverter chain and AES SBOX, are implemented in Xilinx Virtex2Pro FPGAs and the experiments are carried out in 30 copies of XUP-V2Pro boards. In the inverter chain experiment, the delay along inverter chain and each inverter are analyzed. In the AES SBOX macro, total of 26 different paths are analyzed to measure the die-to-die and within-die variations. In this experiment, the measurement resolution is set to 25 ps.

4.2.2.1 Calibration

As mentioned in section 3.6.1 , the calibration process for REBEL is designed to eliminate transition propagation along the scan chain, as specified by T_{sc} in eqn 3.1. Once the delay along scan chain is eliminated the delay along PUT can be determined. In the previous work [48], I proposed that in calibration process, all the scan cells in the chain are put in flush-delay mode, an edge is launched from the SI pin and a sequence of launch-capture tests is carried out. In each successive experiment the capture timing is incremented in small steps (that defines the measurement resolution) and the digital values stored in the scan chain are analyzed. But I found that the uncertainty in the calibration data grows for the scan cells towards the end of the chain while launching the edge from the SI pin. In section 4.1.2.1 an alternative calibration process is explained that can eliminate the higher uncertainty in the calibration data. However, this method is highly design dependent and if the neighborhood logic is unable to generate a hazard-free, robust transition at the first (left most) flip-flops of each segment, deriving a good

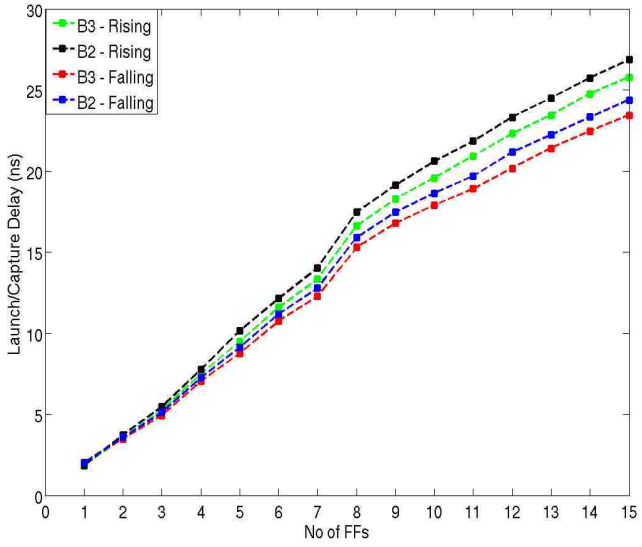
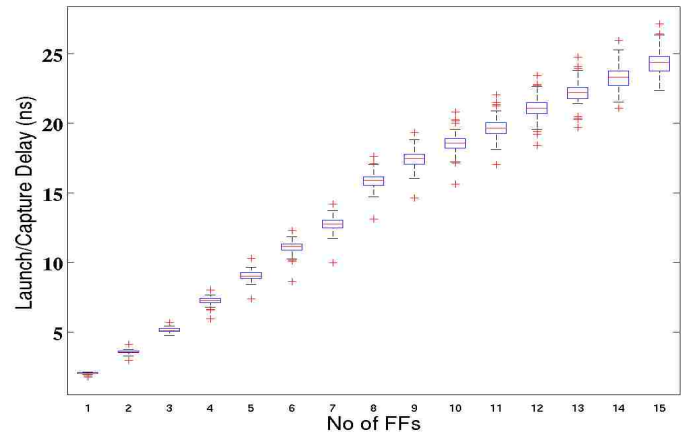


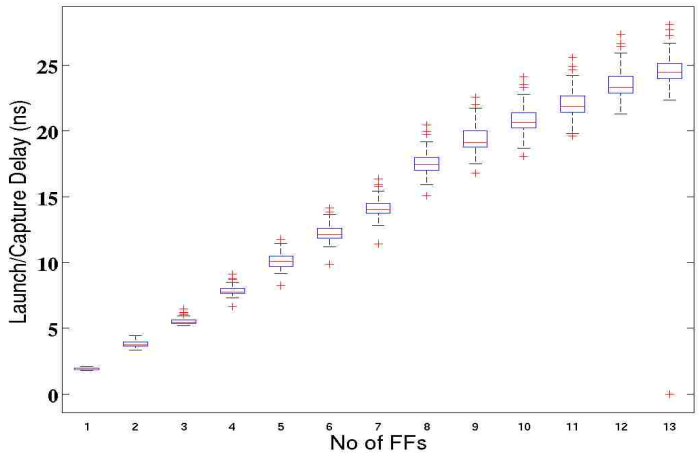
Fig. 4.7: Rising and Falling Edge Calibration Curve for Chips B2 and B3

quality calibration data can be difficult or impossible.

In this work, we propose a method in which the calibration edge is launched from the RCL block of each row that eliminates the aggregation of the uncertainty in the measurement. In Fig. 5.2, the dotted box shows the additional logic required to launch an edge from the RCL. This additional logic, which is essentially a preset-clear flip-flip and few gates, along with the third register creates a falling edge when a '0' is stored in that register and a rising edge when a '1' is stored. This facilitates an on-chip edge creation for the calibration process and also allows calibration of each row individually. This on-chip calibration edge creation provides the following benefits: uncertainty in the calibration data is reduced significantly and results in lower measurement errors, and, since all the segments can be calibrated in parallel, test time can be reduced significantly. The different velocities of rising and falling edge in the scan chain require calibration data for both rising and falling edge.



(a)



(b)

Fig. 4.8: Box-plot of Propagation Behavior along the Scan Chain (a) Falling Edge and (b) Rising Edge

Fig. 4.7 shows the rising and falling calibration curves of two chips (B2 and B3) in 25ps resolution. It is clear that the rising and falling edges have different propagation velocity along the scan chain. This calibration data is then used to eliminate the delay along the scan chain from the launch-capture period to the actual combinational path delay. Once the calibration tests are performed, the delay along any portion of the delay chain can be computed as given by Eqn. 4.1. The expression gives the delay along the delay chain (T_{SC}) between two scan flip-flops FFe and FFs as the difference in the delays

measured under the calibration tests to these scan FFs, i.e., T_{ffe} and T_{ffs}.

$$T_{sc} = T_{ffe} - T_{ffs} \qquad \text{Eqn. 4.1}$$

where, T_{sc} = Delay along the scan chain

T_{ffe} = LC Period for the calibration edge to reach 'FF_e'.

T_{ffs} = LC Period for the calibration edge to reach 'FF_s'.

The calibration data has to be collected ahead of the path delay measurements for each chip. For high accuracy delay measurement, the high resolution calibration data has to be collected, in which case the calibration process may be long as the calibration time depends on the scan chain length and measurement resolution. Interestingly, Fig 4.7 shows that the calibration curves are linear and scaled for different chips. The calibration time can be reduced in one of the two ways: 1) Calibration data can be collected for the first and last scan cells of each row and the calibration data for the other scan cells can be calculated by linear interpolation, 2) High resolution calibration can be done for one of the chips and it can be scaled for other chips based on the scan chain end points data. Obviously, interpolation and scaling may increase the measurement error due to within-die variations. Hence, there is a trade-off between the calibration time and measurement accuracy, based on the test time and resolution requirements the decision can be made.

In this experiment, I collected calibration data for all the chips at the required measurement resolution. The calibration data can be used to characterize the delay variability along the delay chain created, essentially, in the delay along the scan chain. Since the basic structure of delay elements are similar and spread across the die, the calibration data can be used to analyze the spacial variations. Fig. 4.8 shows the variation in the falling (a) and rising (b) edge calibration data of 30 FPGAs. This curve explains

the variations in the delay chain created along the scan chain among the FPGAs. The delay variation of delay chain measures up to 1ns for both rising and falling edge transitions (4% of the delay along scan chain of length 16 bits).

Path	Digital Snap-Shot	T_{lc} (ns)	T_{sc} (ns)	T_{path} (ns)
P1	**11111**00000000000	18.050	7.800	10.250
P2	10**000000**11111111	17.650	9.125	8.525
P3	11110**11111**000000	18.675	9.325	9.350
P4	**0000000**111111111	18.575	11.925	6.650
P5	1111110**111111**000	18.475	9.425	9.050
P6	110**0000000**111111	18.075	10.150	7.925

Table 4.1: Delay Measured in Different Combinational Path

4.2.2.2 Path Delay Measurement

With the availability of calibration data, the combinational path delay can be calculated from the digital snap-shot. Table 4.1 shows the digital snap-shots of the scan chain segments, where the edges propagated, and corresponding path delay value for 6 different combinational paths of the AES SBOX design. The bits high-lighted in green (underlined) color reflects the propagation of the transition along the scan chain. Using this scan chain digital snap-shot, the delay along the scan chain is calculated from the calibration data. The actual path delay, after subtracting the calibration data, is shown in the last column of the table. The results shown in the table are the path delay measured with high-resolution (25 ps) calibration data and clock strobing mechanism.

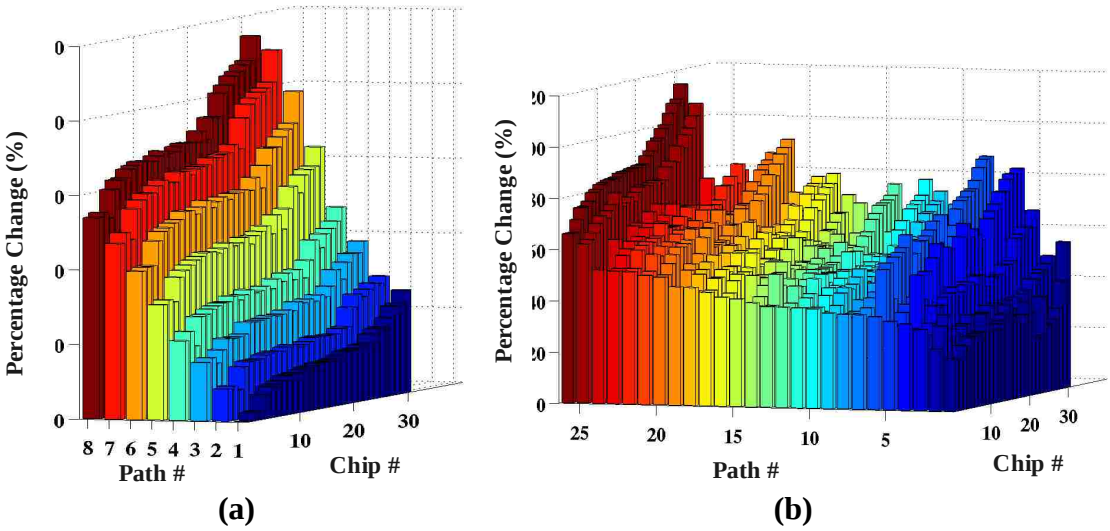


Fig. 4.9: Path Delay Variation across 30 FPGA chips (y-axis) in (a) Inverter Chain and (b) AES SBOX paths

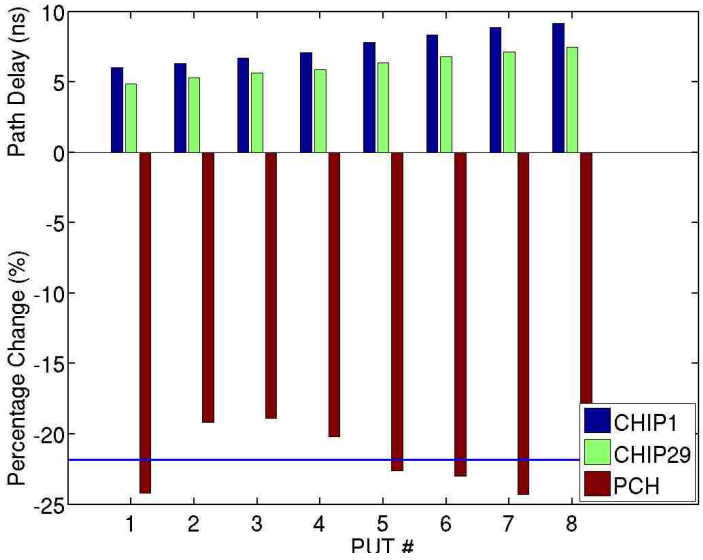


Fig. 4.10: Chip-to-Chip and Within-Chip Variations in Path Delays

4.2.2.3 Regional and Chip-to-Chip Variation Analysis

Fig. 4.9(a) shows the path delay variation in the 8 different paths created by inverter chain among 30 FPGA chips (sorted along y-axis). The different paths are plotted along the x-axis, chip numbers along the y-axis and normalized path delay as percentage

change along the z-axis. The path delays are normalized with respect to the reference component path #1 from chip #1 (lower, right-most element). Although the die-to-die variation is obvious along the y-axis, the within-die variations are also observable. For example, the percentage change values difference between path #2 and path #3 varies from 4.4% up to 9.5% (variation of 5.1%) and difference between path#5 and path#6 varies from 8.6% to 14.8% (variation of 6.2%).

Fig. 4.9(b) shows the path delay variations in 26 different paths in AES SBOX macro. The path delays are normalized with respect to the smallest path of the fastest chip and the normalized delay percentage change is plotted along z-axis. The die-to-die variation in the path delays among the 30 chips is clear along the y-axis. For example, in path #26, the percentage change value ranges from 65.9% to 114.9% (with respect to the smallest path). The figure also explains the within-die variations, for example, the difference between the path #25 and path #26 percentage change values among 30 chips varies from 2.6% up to 9.4%.

Fig. 4.10 explains the die-to-die and within-die variations more clearly. Here, the eight path delays (inverter chain) of slowest and fastest chip are plotted (top). The percentage change (PCH) in path delays with respect to the fastest chip are also plotted (bottom). The average of these PCH values represented as a horizontal line (at the bottom) explains the magnitude of die-to-die variations which is about 21.8%. Moreover, the variations in these PCH values is due to the regional variations and it measures up to 5.4% .

5 ns

Path	FF15	FF14	FF13	FF12	FF11	FF10	FF9	FF8	FF7	FF6	FF5	FF4	FF3	FF2	FF1	FF0
PO7	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PO6	X	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
PO5	X	X	0	1	1	1	1	1	1	1	1	1	1	1	1	1
PO4	X	X	X	1	0	0	0	0	0	0	0	0	0	0	0	0
PO3	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0
PO2	X	X	X	X	X	0	0	1	1	1	1	1	1	1	1	1
PO1	X	X	X	X	X	X	0	0	1	1	1	1	1	1	1	1
PO0	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0

10 ns

Path	FF15	FF14	FF13	FF12	FF11	FF10	FF9	FF8	FF7	FF6	FF5	FF4	FF3	FF2	FF1	FF0
PO7	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
PO6	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PO5	X	X	0	0	0	0	1	1	1	1	1	1	1	1	1	1
PO4	X	X	X	1	1	1	0	0	0	0	0	0	0	0	0	0
PO3	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0
PO2	X	X	X	X	X	1	1	1	0	1	1	1	1	1	1	1
PO1	X	X	X	X	X	X	0	0	0	0	1	1	1	1	1	1
PO0	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0

15 ns

Path	FF15	FF14	FF13	FF12	FF11	FF10	FF9	FF8	FF7	FF6	FF5	FF4	FF3	FF2	FF1	FF0
PO7	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
PO6	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PO5	X	X	0	0	0	0	0	0	1	1	1	1	1	1	1	1
PO4	X	X	X	1	1	1	1	1	0	0	0	0	0	0	0	0
PO3	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0
PO2	X	X	X	X	X	1	1	1	1	1	0	0	0	0	0	0
PO1	X	X	X	X	X	X	0	0	0	0	0	0	0	0	1	1
PO0	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0

Fig. 4.11: Digital Snap-shots at 5ns, 10ns and 15ns

4.2.3 REBEL as an On-Chip Logic Analyzer

In addition to the path delay measurement, the REBEL test structure models the capability of a logic analyzer, where the static and dynamic hazards can also be captured. The digital snap-shot captures all the activity at the output of the PUT including the static and dynamic hazards that reveals additional information about the delays to the off-path inputs or segments. Fig. 4.11 shows the digital snap-shot of a segment of the scan chain at 5 ns, 10 ns and 15 ns launch-capture periods. The blue circles in the figure highlights the hazards captured in the scan chain. Remember that the positive pulses in the hazard will disappear after propagating through few registers due to 'pulse-shrinking' effect, where as the negative edges will expand. For example, in path PO6, a positive pulse is registered at 5 ns launch-capture period, which is disappeared at 10 ns and 15 ns launch-capture periods. However, the negative pulse in path PO2 is expanding. In low resolution measurements, hazards with smaller pulse widths may go undetected. But, the high-

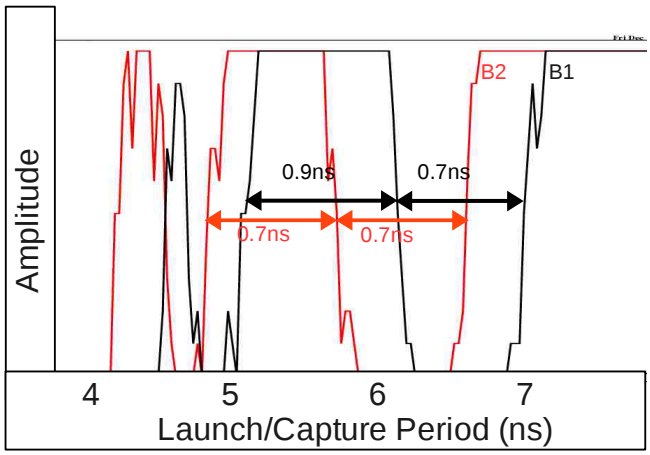


Fig. 4.12: Off-Path Inputs Delay Behavior Analysis

resolution measurements (using clock strobing technique) will capture almost all the activities at the output of the combinational logic.

Fig. 4.12 shows the static (a) and dynamic hazards (b) captured in a specific combinational path output in the high-resolution test. From the digital snap-shots captured in several REBEL tests, a waveform with a resolution 25 ps is generated. Fig. 4.12 shows the hazard occurred in two different chips (B1 and B2). This particular PUT output had a positive and a negative pulse before the final transition. The positive pulse widths of the hazard is different in two chips, 0.7 ns and 0.9 ns for chip B1 and B2 respectively, where as the negative pulse width is 0.7 ns for both the chips. This explains that the delay behavior variation of the off-path segment in those two chips. Analyzing the hazards in all the logic tests will give more information about the off-path segments delay behaviors. Moreover, the information about the occurrence of static and dynamic hazards are useful in post-silicon debug.

4.3 REBEL-Integrated BIST Architecture for Path Delay Variability Characterization

REBEL can be integrated into various types of scan-based DFT structures, e.g. built-in-self-test (BIST), Illinois scan design, etc. For BIST architectures, REBEL can be integrated into the circuit-under-test and deterministic-delay-test vectors or random vectors can be used to measure path delays. However, the BIST controller has to be modified in such a way that it can append the REBEL configuration information with the deterministic or random vectors used for delay testing. For Illinois scan architectures, the basic principle on which it operates, i.e., exploiting the independent nature of the modules in the design, can be leveraged for integrating REBEL also. The scan chain is already partitioned into several segments based on dependency for the purpose of broadcasting the test vectors. However, in order to integrate REBEL, it has to be further divided, which is fairly simple.

In this section, I present the integration of REBEL with BIST architecture and its application for analyzing the process variation in path delays. The presented architecture combines the benefits of BIST and REBEL. First, as mentioned in the previous section, one of the main sources of the uncertainty or measurement error is the clock jitter, this can be reduced significantly by using an on-chip clock scheme. Second, the on-chip test vector generation eliminates the time associated with transferring test data from/to the test equipment, significantly reducing the overall test time. Finally, the two on-chip capabilities mentioned above eliminate the need for costly test equipment, greatly reducing the test cost.

4.3.1 Experimental Design

Fig. 4.13 shows the architecture of the REBEL-integrated BIST system used for path delay measurement. The circuit-under-test (CUT) is a REBEL-integrated, pipelined AES encryption engine (1 round). MUX-D based REBEL is integrated into the HDL description of the AES macro. The test pattern generator (TPG) automatically generates random test vectors using a linear feedback shift register (LFSR). The output response analyzer (ORA) analyzes the output of each test for validity, i.e., whether there is a transition at a specific insertion point for a given random test vector, and returns the response of a valid test to the logic BIST controller. The logic BIST controller coordinates TPG, CUT, clocking scheme, and ORA.

The BIST controller first sends the control signals to the TPG which creates a random test vector and shifts it into the launch segment of the CUT. Then, it configures the REBEL structure for a specific path delay measurement using the integrated REBEL controller. Finally, it carries out launch-capture tests by sweeping the LC period until the transition propagation reaches a specific target flip-flop and returns the corresponding LC period value. It also invalidates the insertion points that do not have any transition for a specific random vector, returning the data for the valid transitions only. In some applications, like delay testing, the 'golden signature' is also embedded into the BIST logic controller, and often contains diagnostic logic for fault diagnosis. In this experiment, we used serial communication to send the test responses to the host computer and process data for analyzing process variations in path delays.

The digital clock managers (DCMs) in Virtex2Pro FPGAs are used to generate the

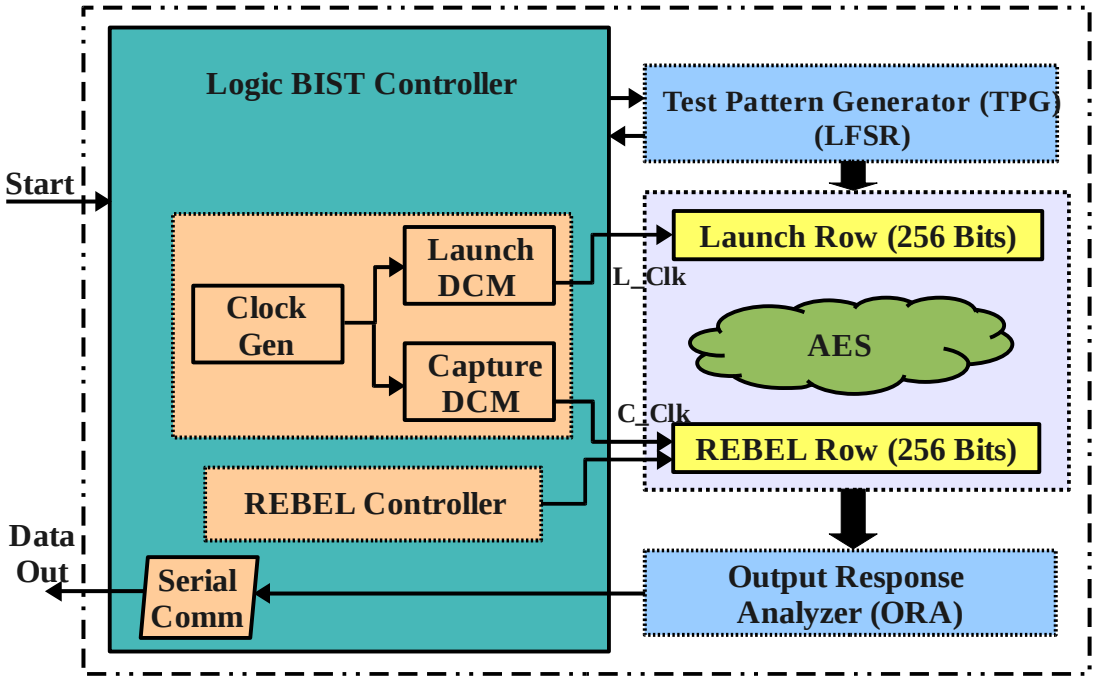


Fig. 4.13: REBEL-integrated Logic BIST Architecture

launch and capture clocks for REBEL launch-capture tests. In addition to the clock-divider feature, DCMs allow for creating specific phase shift to the clocks. The launch clock is generated without a phase shift and the capture clock is generated with a phase shift tuned based on the required LC period. The BIST controller tunes the LC period for each LC test based on the response of the previous test and stops the path delay measurement when the transition reaches a specific target flip-flop.

Fig. 4.14 demonstrates the launch and capture clocks generated using the on-chip DCMs. The launch clock is derived from the system clock without a phase shift and the capture clock is derived with a phase shift. In this experiment, the system clock period is 10 ns (100 MHz) and the launch-capture period sweep range is set to 5 ns – 15 ns. By controlling the phase shift, the capture clock can be tuned to achieve any LC period within this range. Fig. 4.14 also shows the smallest and largest LC periods (bottom two

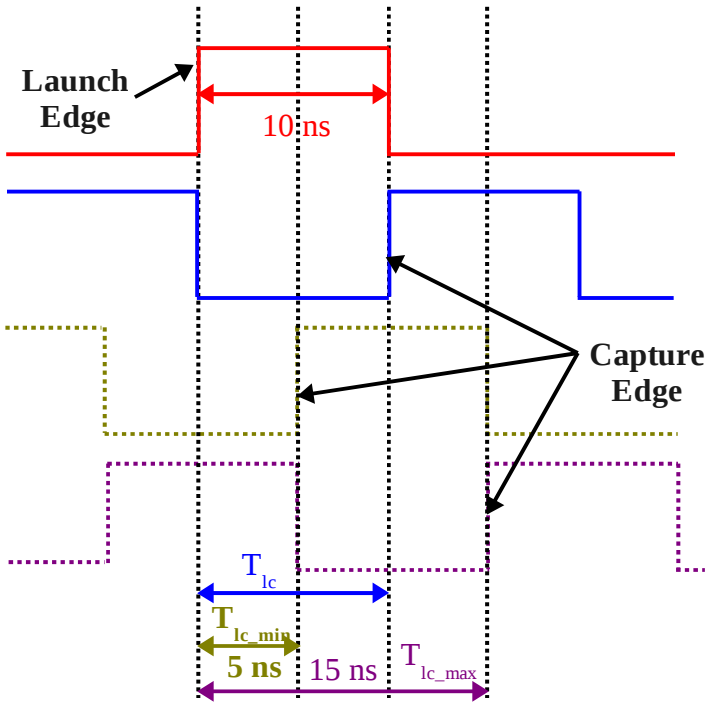


Fig. 4.14: Launch-Capture Clocking Scheme using DCMs

waveforms in the figure) used in this experiment. Obviously, the shortest path that can be measured using this scheme is 5 ns in length. For the CUT that is used in this experiment, AES encryption engine, most of the path lengths are above 5 ns and hence it does not affect any path delay measurement (the delay values that are used in this analysis include path delay and the delay along the 4 flip-flops from the insertion point). However, the Xilinx Virtex2Pro FPGAs allow for tuning LC period to any required value by coarse and fine tuning the phase shift of the capture clock.

4.3.2 Experimental Results

In this section, I present the path delays measured using the BIST engine explained in the previous section and the die-to-die and within-die variations in these delay values. Using the random test vectors generated by the LFSR a total of 4657 valid

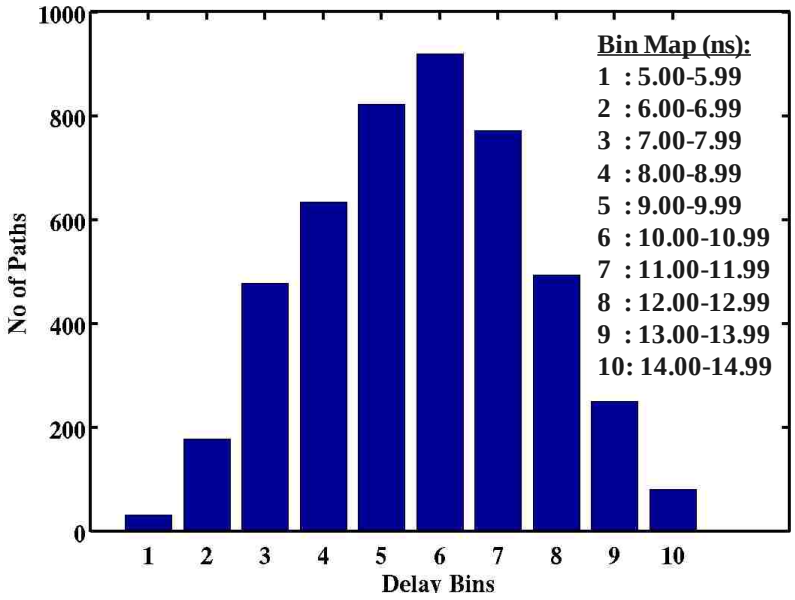


Fig. 4.15: Path length Distribution in Pipelined AES Encryption Design

path delays are measured in 30 copies of the FPGA. Fig. 4.15 shows the distribution of path lengths of the AES macro implemented in one of the chips (B1). In this figure, the bins are plotted along the x-axis and the number of paths that falls into that bin are plotted along the y-axis. Each bin in this figure corresponds to 1 ns delay range. The delay values used in this analysis are measured at the 4th target flip-flop and include path delay as well as the delay along the first four flip-flops from the insertion point. In this experiment, the clock is swept from a launch-capture period of 5 ns to 15 ns. It is clear from the Fig. 4.15 that the path length distribution is close to a Gaussian distribution with the median at the 10 ns – 11 ns range.

4.3.2.1 Noise Analysis

In order to analyze the measurement noise or uncertainty, the path delay measurement is carried out 10 times (samples) and the standard deviations among these measured values are analyzed. The delays of all 4657 paths are measured 10 times on all

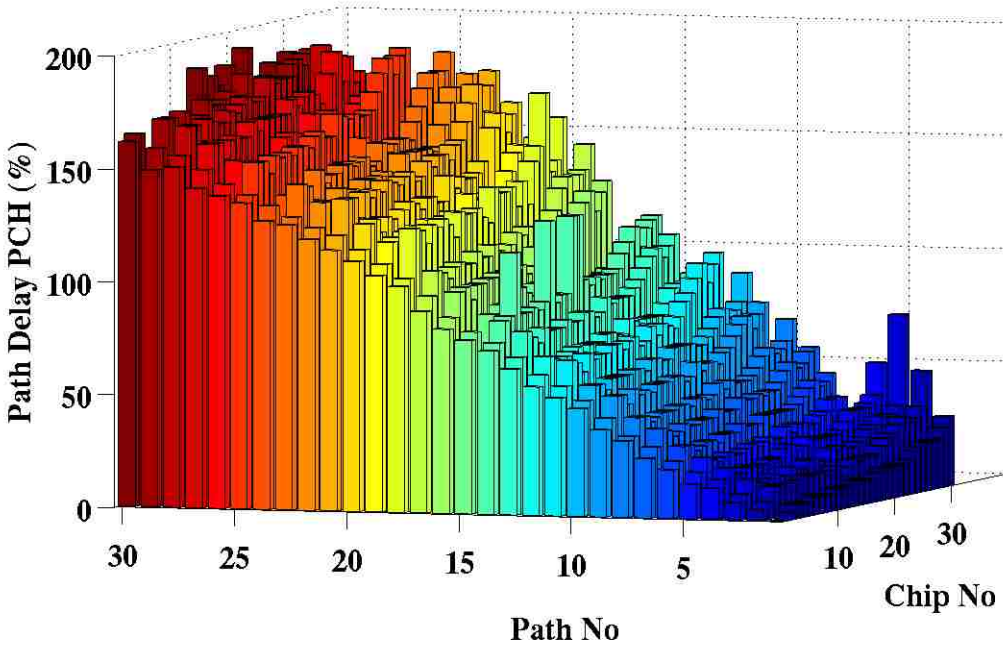


Fig. 4.16: Process Variations in delays of 30 Representative Paths among 30 copies of 130nm FPGAs

30 boards and standard deviations are computed for each of these measurements. The mean of all the standard deviations is approx. 31 ps and a 3σ of 93 ps. This is about 5.6x improvement in the quality of the measurement when compared with the data collected using the clocks generated from the structural tester (524 ps). The main reason for this improvement in the measurement quality is the quality of the on-chip clocking scheme. The on-chip clocking schemes have small clock jitter and results in less uncertainty/noise in the measurement.

4.3.2.2 Die-to-Die and Within-Die Variation Analysis

In order to analyze the process variation in these path delays, the experiment is carried out in a set of 30 Xilinx Virtex2Pro FPGAs. For simplicity of analysis, a set of 30 paths representing each bin in Fig. 4.15 (3 paths from each bin) are chosen. Fig. 4.16 shows the delay variations in these representative paths among 30 FPGAs (sorted along

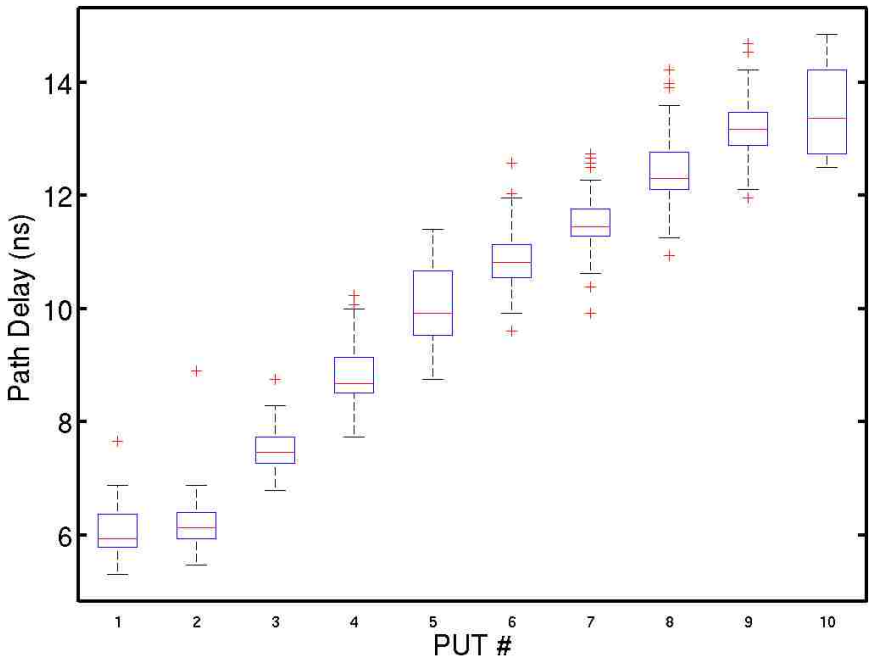


Fig. 4.17: Die-to-Die Variations: Box-plot of 10 Representative Paths Delays (from 10 Different Bins) among 30 FPGAs

y-axis). Different paths are plotted along the x-axis, FPGA chip numbers along the y-axis and normalized path delays as percentage change along z-axis. The delay values are normalized with respect to the shortest path in the fastest chip (path #1 from chip #1). The die-to-die variations are obvious along the y-axis and subtle within-die variations are also visible.

Fig. 4.17 displays the boxplot analysis of 10 representative path delays (one path selected from each bin of Fig. 4.15) from 30 FPGAs. The figure plots the path numbers along the x-axis against the measured delays on the y-axis. The boxplot associated with each path identifies the median, upper and lower quartiles as horizontal lines. A set of dotted lines extend to smallest and largest delays and a '+' indicator identifies outliers in the distribution. From the Fig. 4.17 it is clear that the die-to-die variation ranges from 600 ps up to 1.6 ns. The magnitude of variation in these path delays depends on path length

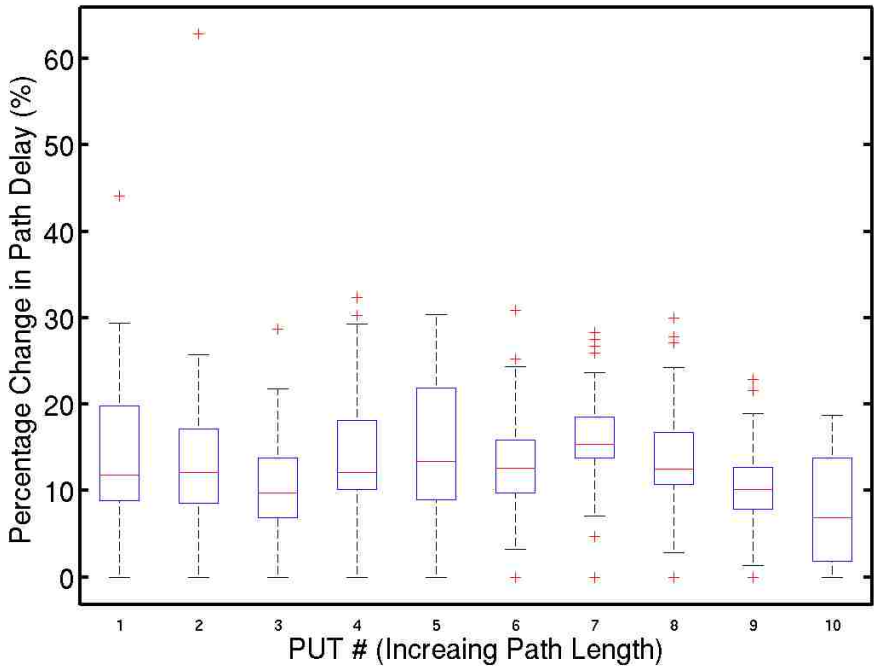


Fig. 4.18: Die-to-Die Variations: Box-plot of Normalized Delay Values (Percentage Change with respect to Fastest Chip) among 30 FPGAs.

and implementation of the specific path on the FPGA. In order to analyze the variation magnitude change with respect to the path lengths, the path delays are first normalized with respect to the delay of fastest chip in the population.

Fig. 4.18 shows the boxplot analysis of the normalized delay values (percentage change with respect to the fastest chip) from 30 copies of the FPGA. The path numbers are plotted along the x-axis (increasing path length) against the percentage change in path delays. It is clear from the figure that the short paths have significant variations (the center horizontal median line) and as the path length increases beyond certain length the magnitude of variation reduces due to 'averaging-effect' (right most paths). In this experiment, the launch-capture period is limited to 15 ns and the delay values include combination path delay and delay along four flip-flops from the insertion point. Although, the long paths are excluded in this analysis, if they were included we would

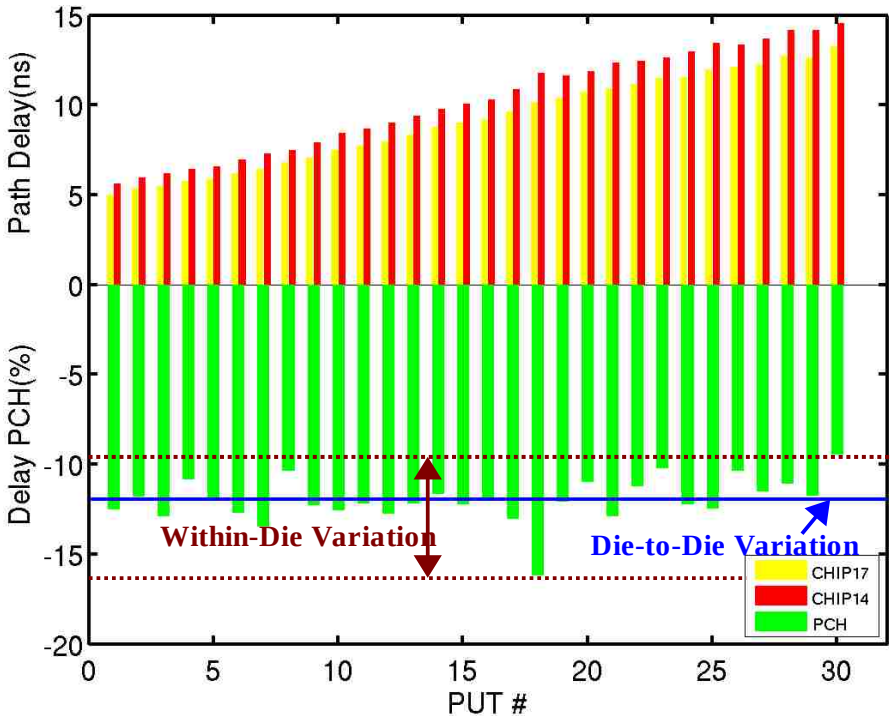


Fig. 4.19: Die-to-Die Variations and Within-die variation Analysis using the Path Delays from AES Macro

see the reduction in process variation magnitude even more clearly.

In Fig. 4.19, the die-to-die and within-die variations are captured very clearly. In this case, the representative path delays from two of the chips (slowest and fastest among the population of 30 chips) are analyzed. The path delays of these two chips are plotted at the top of the figure and the percentage change in the delay with respect to the fastest chip is plotted at the bottom. These percentage changes values explains the variation in these path delays from one chip to the other. The mean of these PCH values is about 12% and explains the die-to-die variations in these chips (plotted as horizontal line in Fig. 4.19). The deviations in the PCH values explains the within-die or regional variation in these chips, the PCH values ranges from 9.5% to 16.5% resulting in a within-die variation of 7%.

CHAPTER 5

Trojan Detection Sensitivity Analysis

Before explaining REBEL's ability to detect hardware Trojans, I present a preliminary work that analyzes the sensitivity requirement for two side-channel parameters, namely delay and power, on an emulation platform. In this chapter, the impact of specific Trojan implementation characteristics on these two parametric parameters is first investigated using FPGAs as a validation platform. The additional logic inserted by an adversary, which represents the Trojan, can be classified into two general forms, the “trigger” and the “payload”. In order to realize the trigger portion, the inputs of the Trojan must connect to existing logic nodes in the IC[2]. These Trojan gate connections increase the capacitive load on these nodes. The payload portion requires change(s) to, or the insertion of a gate in series with, the existing logic of the IC. Both of these modifications change the delay and power characteristics of the IC.

Given the difficulty of designing and fabricating ASICs for this study, we choose to carry out the hardware experiments using the FPGA as a surrogate. Although the implementation of logic paths in a FPGA and a typical ASIC are significantly different, we believe the analysis is meaningful to ASICs because of commonalities in the two platforms, namely, FPGAs, like ASICs, are subject to the same types of process variations, and many of the primitive components of FPGAs, e.g., FFs, lookup-tables, multiplexers, etc. are also found in ASICs. More importantly, our analysis is focused on

2 The exception is fully autonomous Trojan implementations, which we do not consider in this work.

evaluating the *relative* magnitudes of delay and power variations introduced by noise verses those introduced from Trojan gates, and is therefore less sensitive to the actual circuit implementation characteristics. In fact, the subtle delay variations introduced in the FPGAs to model Trojan trigger connections are relatively smaller than would occur in ASICs. This is true because logic gates implementations in FPGAs have larger overheads in area, power and delay.

The objective is to evaluate the sensitivity of power and delay analysis to small signal anomalies introduced by the invasive characteristics of Trojan circuit components. The signal anomalies are introduced in a manner that realistically represents the stealthy nature of a Trojan. To simplify the measurement of power and delay and, more importantly, to allow a high degree of control over switching activity, we choose to use a set of ring oscillators (ROs) to represent the core logic of the digital chip under test. The simple structure of an RO also facilitates alternative configurations designed to produce only very small signal anomalies when compared with the default, Trojan-free configuration of the RO. Although this strategy does not emulate an actual Trojan scenario, it is well suited to serve our objective of evaluating sensitivity. More importantly, it decouples our analysis from any specific core logic or Trojan implementation, and therefore, serves as a general evaluation platform to determine the sensitivity requirements of power and delay analysis for Trojan detection.

In the hardware experiments, the FPGAs are configured with 32 copies of a 9-stage ring oscillator (RO), designed as a hard macro and distributed uniformly across the re-configurable fabric of the FPGAs. A select MUX allows each of the RO outputs to be routed, one at a time, to a high-speed external connector, where an oscilloscope is used to

measure its frequency. A high resolution source-meter is used to provide voltage and measure the supply current of, the FPGA's core logic.

A noise analysis, as well as within-die and chip-to-chip process variation analysis, are carried out on several copies of the FPGA to determine the level of process and environmental (PE) noise. The placement strategy of the 32 ROs is designed to enable a within-die variation profile to be determined. The analysis shows predominately random variations in RO current and frequency but a strong correlation between the two, suggesting that using the two together may be more effective than using either one alone.

The noise and PE variation analysis are carried out using a specific configuration of the ROs that represents a Trojan-free condition. Once the statistical limits of the noise and PE variation are determined, a variety of invasive Trojan circuit components are investigated, hereafter referred to as 'emulated Trojans'. The emulated Trojans are added to the RO hard macro and are designed to emulate additional capacitive loading and increased delay that a Trojan is likely to add to nodes and paths in the Trojan-free design of an ASIC chip. The additional loading (introduced by the Trojan's trigger connections) is implemented by increasing the fan-out of the inverters inside the RO. The fan-out gates reduce the RO frequency and increase the power consumption of the RO. The payload portion of a Trojan is modeled as series inserted gates. The 'odd # of inverters' requirement of the RO forces the addition of two inverters, which extends the RO to 11-stages.

Several statistical outlier techniques are used to determine the sensitivity of current and delay analysis for detecting the emulated Trojans. Although it is possible to use each of these parametric parameters to detect Trojans by themselves, we found that

using them together is more effective.

A calibration technique is proposed that is able to further enhance the sensitivity of our proposed statistical methods. The proposed calibration technique leverages a set of existing embedded ring oscillators (EROs). EROs are inserted by designers into ASIC product chips for the purpose of monitoring process parameters. Given that our experiments are already based on ROs, the calibration technique is easily implemented by selecting a 'calibration RO' from each region (described below). We show that Trojan detection sensitivity can be increased by an order of magnitude over power or delay analysis alone, by using calibration and regression analysis of the measured current and delay parameters.

5.1 Experimental Setup

The Virtex-II Pro[3] FPGAs under investigation in this work are fabricated in an 130 nm technology. Although process variations in this technology node are not as severe as they are in cutting-edge technology nodes, e.g., 45nm, our regional analysis approach and calibration methods are designed to scale, and therefore, will remain effective in more advanced technologies.

Figure 5.1 shows one of the XUP-V2Pro boards, along with a Keithley 2400 and a Tektronix TDS 7254 digitizing oscilloscope. The XUP-V2Pro provides a connector bank and jumpers to allow several components of the board, including the V2Pro chip, to be powered up using alternative external source meters, i.e., the default board power supply can be disconnected. We use a Keithley 2400 high-precision source meter as an

3 Mounted on Digilent development boards

Fig. 5.1: XUP V2-Pro board and experimental setup with Keithley 2400 source meter and Tektronix TDS 7254 digitizing oscilloscope.

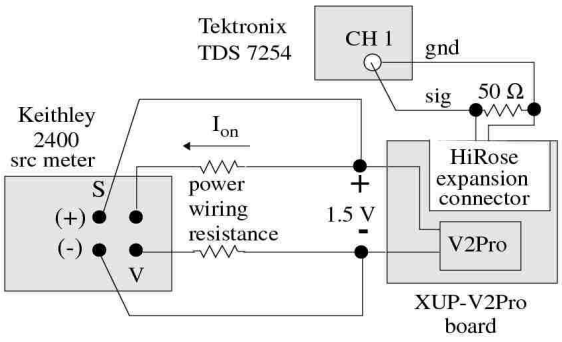


Fig. 5.2: Custom connections to the XUP-V2Pro board for Current and Frequency Measurements

alternative power source for the V2Pro chip.

The wiring configuration for the source meter is shown in Figure 5.2. The Keithley is setup in 4-wire (sense) mode which allows the power terminals close to the board to be maintained at 1.5 V, the core supply voltage for the V2Pro. This mode effectively eliminates the IR drop across the power wiring from the Keithley to the board.

A MUX is configured into the re-configurable fabric of the FPGA to enable one of the RO outputs to be routed to a pin on the V2Pro, and then off the XUP board through a high-speed HiRose expansion connector. A 50 Ohm termination resistor is connected between signal and ground on this connector pin as shown in Figure 5.2, and an active probe with an input bandwidth of 1.5 GHz (Tek P6245) is used to measure the frequency of the pulse train generated by the RO on the oscilloscope.

5.1.1 Experimental Design

A 9-stage RO is used as the primary design component in the hardware experiments, and is shown in Figure 5.3. The RO is configured with an ‘enable’ pin to

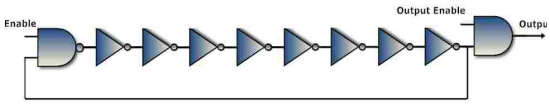


Fig. 5.3: Schematic of 9-Stage RO

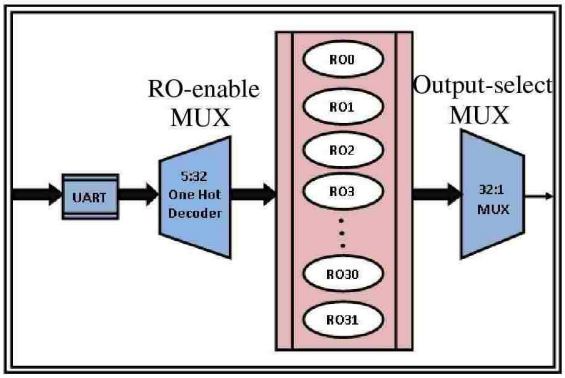


Fig. 5.4: FPGA Design Components Added as Support Logic to Control ROs

allow any one or more of the 32 ROs on a given chip to be enabled. An 'output enable' pin is also included to prevent the RO pulse train from driving the output routing support circuitry. This is required to obtain accurate current measurements for the ROs because the output routing support circuitry impacts the power and on-chip temperature significantly when enabled.

Figure 5.4 shows the support circuit components configured into the FPGA. We use a serial port (UART) to set the select inputs of the *RO-enable MUX*. The 32 outputs of the RO-enable MUX are connected to the *enable* inputs of the NAND gates of the 32 ROs (see Figure). The select MUX is designed such that more than one RO can be enabled simultaneously. The *output-select MUX* routes exactly one of the RO outputs to the output pin as shown on the HiRose connector in Figure 5.1. LABVIEW software was used to completely automate the data collection process.

Figure 5.5(a) shows the placement of the 32 ROs in a block level diagram of the V2Pro. The ROs are placed on 4 diagonals from the corners of the chip to the center. The RO in the upper left corner is designed as the **reference RO** (to be discussed). The

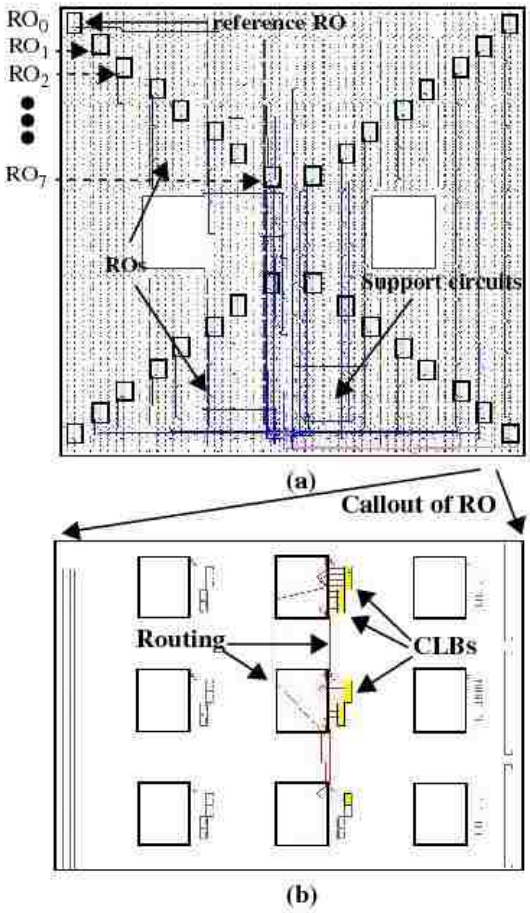


Fig. 5.5: FPGA architecture with embedded hard-macro Ring Oscillators (ROs) and support circuits, (b) Ring Oscillator (RO) hard macro design, showing the 5 CLBs and connections that implement the 9 stages.

support circuits, e.g., UART and MUXes, are shown along the bottom of the figure. Figure 5.5(b) shows a callout of the RO hard-macro, which includes a set of 5 CLBs and associated routing.

5.1.2 Emulated Trojans

The Trojan scenarios that are investigated fall into two broad categories: 1) Trojans that add capacitive load and therefore increase delay and power consumption, called '**trigger Trojans**' and 2) those that add series inserted gates, which increase delay

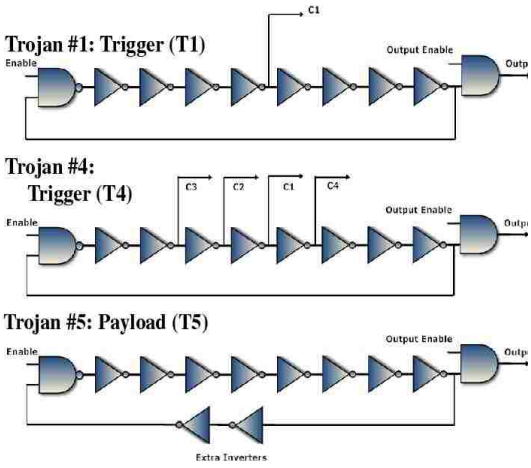


Fig. 5.6: Modified RO Schematic showing implementation of two 'trigger' and the payload components of the emulated Trojans.

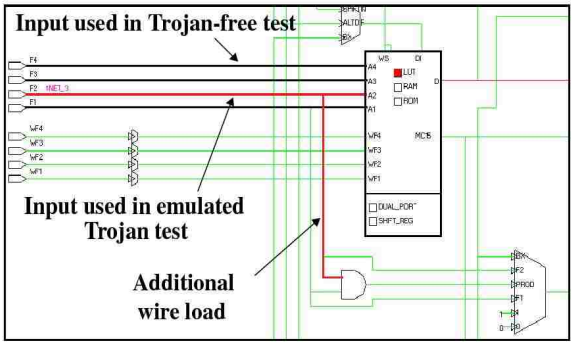


Fig. 5.7: Xilinx Fedit view of CLB showing 'trigger' Trojan Implementation.

but have a smaller impact on power, called '**payload Trojans**'. Each of these models the nature of actual Trojans in ASIC chips, where the trigger portion of the Trojan is connected to existing circuit nodes for the purpose of activation and the payout portion is inserted into logic paths for the purpose of modifying logic functions.

Figure 5.6 shows the modifications made to the base RO shown in Figure 5.4 to emulate the two Trojan types. The trigger Trojan adds additional capacitive load by leveraging the **intrinsic fanout** that already exists in the CLB blocks. Figure 5.7 shows a screen snapshot using Xilinx's FEdit of a CLB with labels on two of the input wires. For the Trojan-free tests, the upper input is used in the implementation of the RO, and for the trigger Trojan tests, the lower input is used. Larger capacitance loads are implemented by repeating this implementation for other CLBs in the RO, e.g., four CLBs are modified this way to implement Trojan #4 shown in Figure 5.6. The payload Trojan (Trojan #5) shown in Figure 5.6 is implemented by adding two additional CLBs (inverters) to the base RO implementation.

The initial attempt to emulate a trigger Trojan added capacitance load by connecting additional CLBs to nodes in the RO using the switch fabric of the FPGA. Although this strategy increased the level of measured current, frequency was largely unaffected. Closer examination of the FPGA revealed that the switch resistance effectively isolates the additional capacitive load from the driving inverter in the RO, and is not a good model for fan-out in an ASIC. Our choice to leverage the natural fan-out already present in the wiring configuration of the CLB, on the other hand, is a close match to an actual loading condition introduced by a Trojan in an ASIC, and therefore, is a better Trojan emulation strategy.

Each of these Trojans is implemented in a separate RO and is used to replace the Trojan-free version of RO_0 (the reference RO) in 5 different FPGA designs. Therefore, a total of 6 configurations are implemented, one Trojan-free (TF) and 5 Trojan (T1, T2, etc.) implementations.

5.2 Experimental Results

The experiments were carried out on 20 copies of the XUP-V2Pro board. The first set of experiments are designed to determine the impact of noise and process variations on RO power and delay as a means of establishing statistical limits for the subsequent emulated Trojan experiments. Both noise and process variations (within-die and chip-to-chip) work to reduce the sensitivity of delay- and power-based Trojan detection methods, and are increasing in magnitude in nanometer technologies. Therefore, it is critical to measure and evaluate them. The results of this analysis demonstrate that methods designed to calibrate for these sources of variations are essential in achieving meaningful

sensitivity.

5.2.1 Noise and Within-Die Variation Analysis

In order to get a handle on the level of uncertainty in the experimental evaluation process, we carried out a ‘noise’ analysis on several copies of our chips. For each chip, a sequence of 32 *null* current and RO current measurements were made using the Keithley. In order to isolate the current in a RO from that of the support circuitry, the *output enable* in Figure 5.4 was set to ‘0’. Therefore, the support circuitry remains idle (not switching) during all of the current measurements. None of the ROs were enabled for the null current measurements. Immediately following each null current measurement, a RO was enabled and the current was again measured. For each RO, the **net current** is computed by subtracting the *null* current value from the value measured with the RO enabled. This process effectively eliminates the chip-wide DC leakage currents and minimizes temperature variation effects.

We also measured the frequencies of the ROs in a separate process that followed the global current measurement phase. Separating global current measurements from frequency measurements minimized the adverse effects of on-chip temperature fluctuations introduced by the output support circuitry. These temperature fluctuations added significant levels of noise to the global current measurements. For the frequency measurements, the *output enable* was set to ‘1’ and the oscilloscope was set to average 128 samples of the output waveform. This entire sequence of experiments was repeated 12 times on several boards for the noise analysis.

The box-plots in Figures 5.8 and 5.9 summarize the statistics computed using the

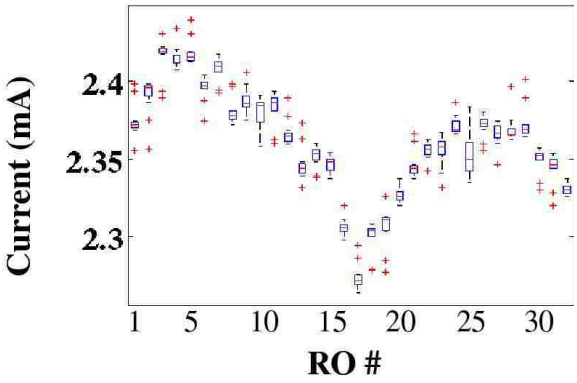


Fig. 5.8: Noise and Intra-Chip Variation: Box-plot of Current Behavior from Chip B1 for 32 ROs.

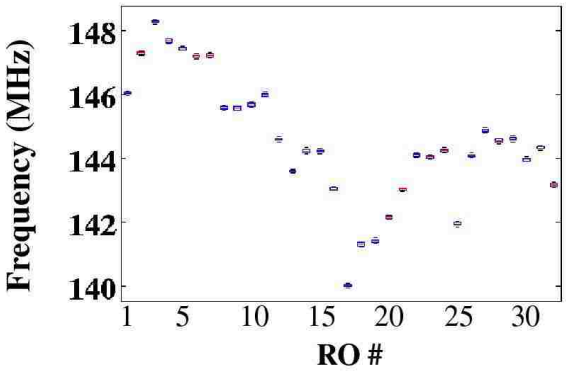


Fig. 5.9: Noise and Intra-Chip Variation: Box-plot of Frequency Behavior from Chip B1 for 32 ROs.

12 data sets of current and frequency measurements, respectively, for chip B_1 for each of the 32 ROs. The ROs are listed along the x-axis. The noise statistics are summarized by 6 values on the y-axis in each box plot: the medium, the upper and lower fence limits (for largest and smallest observations, respectively), upper and lower quartiles, and outliers. The first important observation is that the magnitude of noise is smaller than the within-die variation, particularly for frequency. For example, the set of standard deviations associated with the box-plots reflect the **measurement noise** for each RO, and their average can be used as a measure of noise across the entire analysis. The average standard deviation for current is approx. 9 uA and for frequency, it is 50 KHz. Similarly, **within-die variation** is reflected in the standard deviation of the means across the 32 box-plots in each figure. For current, the standard deviation is 34 uA and for frequency, it is 2 MHz.

A second important observation in the data is the correlation between current and frequency, which can be seen by comparing corresponding ROs in each figure. The Pearson correlation coefficient measures the level of correlation between two data sets.

The correlation coefficient computed using the 32 mean currents from Figure 5.8 against the 32 mean frequencies from Figure 5.9 is 94.4%, where 100% represents perfect correlation.

Average case analysis of the noise works well to show trends. However, worst case analysis is needed to properly determine the sensitivity of power and frequency for detecting Trojans. The worst case 3σ values of the noise are 53 uA and 332 KHz, and as **percentage change**, are approx. 2% and 0.2%, respectively. Likewise, the worst case 3σ values for within-die variation are approx. 4-5% for both current and frequency. These worst case values have significant bearing on defining the statistical limits of the noise, which we will show in the Trojan analysis section below.

5.2.2 Chip-to-Chip Variation Analysis

The behavior of the mean values shown in the box-plots of Figures 5.8 and 5.9 for one chip is different for each of the other chips. In this section, we analyze the chip-to-chip variation in the 20 FPGAs and show that it is predominately random with a large variance. Unlike the noise analysis, only one sample of current and frequency from each chip is used in the chip-to-chip analysis. The chip-to-chip variations in current are presented and those for frequency are summarized.

A simple strategy for analyzing chip-to-chip variations is to construct a 20x32 matrix of the RO currents (or frequencies) from the chips, with the rows representing chips and the columns representing ROs. From this matrix, the mean values of the currents in the 32 columns are then computed. Since the mean is computed across the chips for each RO, any systematic trend that is common to all chips will produce a

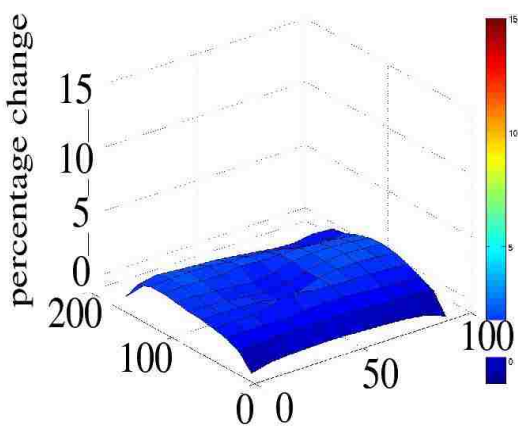


Fig. 5.10: Chip-to-Chip Variation: Mean Current Percentage Change (PC_{mx}) w.r.t Reference RO_0.

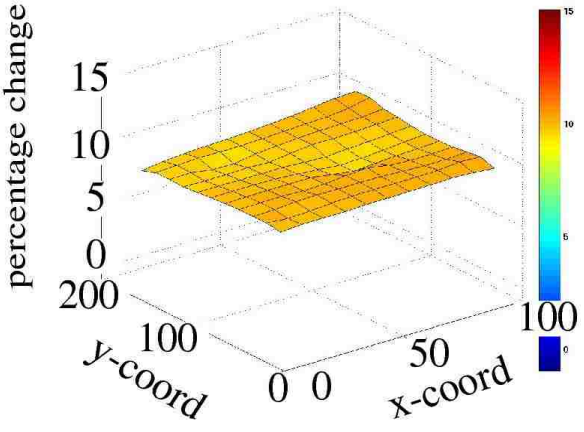


Fig. 5.11: Chip-to-Chip Variation: Three σ Percentage Change in Current (PC_{mx}) w.r.t Reference RO_0.

pattern across the set of mean values. On the other hand, if within-die variation is random across chips, the mean values will be very similar (no pattern).

To show this and to enable comparisons between the current and frequency analyses, the mean values of the columns of the matrix are first converted to percentage change (PC_{mx}) values. PC measures the relative difference in current with respect to a *reference value*. The mean value computed for RO_0 (first column) is used as the reference in this analysis. Eq. 1 gives the expression for PC using mean currents.

$$PC_{mx} = \frac{(I_x - I_0)}{I_0} \times 100 \qquad \text{Eqn. 5.1}$$

Here, I_x represents the mean current for RO_x, and I_0 is the mean current from RO_0.

Figure 5.10 plots the PC_{mx} values on the z-axis of a 3-D surface plot. The x-y plane represents the x-y plane of the chip, with the coordinates given in units of CLBs. The distribution of the 32 ROs, as shown in Figure 3.5 provide a ‘sampling’ of the values in the x-y plane, with the remaining values linearly interpolated from these measured

values. The fact that the entire surface is close to 0% indicates that chip-to-chip variations are primarily random. A small systematic trend is visible as a 'dimple' at (x,y) = (50,100), but the magnitude of PC_{mx} values remains small over the entire (x,y) region of the chip with bounds of -1.0% to 2.5%.

A similar procedure is used to measure the second component of chip-to-chip variation, i.e., the standard deviation. Using the matrix described above, the standard deviations are computed across the columns, as we did for the means. Eq. 2 is

$$PC_{sx} = \frac{3\sigma_x}{I_0} \times 100 \qquad \text{Eqn. 5.2}$$

used to compute a PC_{sx} for each of the standard deviations s_x. s_x is multiplied by 3 to scaleDrawing the standard deviation so that it includes 99.73% of the population.

The PC_{sx} values for the 20 FPGAs are plotted in Figure 5.11, using the same format as that described for Figure 5.10. The entire surface 'hovers' around 10%, again with small deviations between 9% and 11%. The featureless characteristic of the surface supports the random nature of chip-to-chip variations and illustrates that within-die current variations are fairly uniform in magnitude across the chips at approx. 10%. The frequency analysis yielded similar results with bounds of -1% to 2% for PC_{mx} and 11% to 12.5% for PC_{sx}.

5.2.3 Analysis of Delay Variations in ASICs

In order to validate the need for calibration methods further, we analyze delay variations in a set of eighteen 65 nm test chips to illustrate the trend of increasing within-die variations. A block diagram of the test chip is shown in Figure 5.12. The test-chip

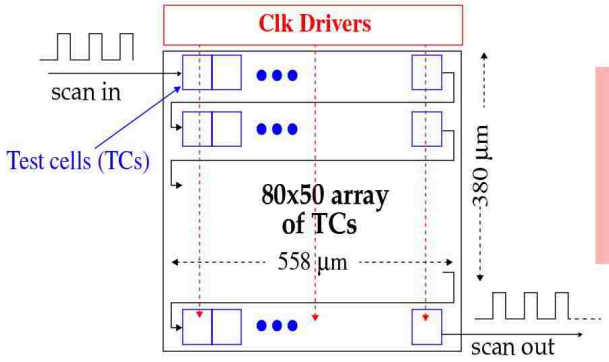


Fig. 5.12: Block Diagram of Test Structure's Scan Path

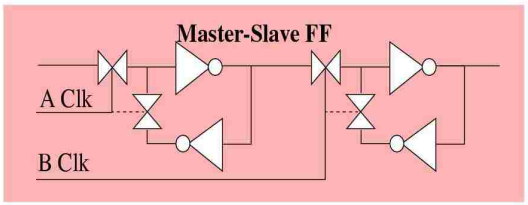


Fig. 5.13: Scan FF with A/B Clocks to Enable Flush Delay Mode

consists of an 80x50 array of test circuits (TCs) connected together through a scan chain. Each of the 4,000 TCs contains three master-slave FFs for a total of 12,000 FFs. The master-slave FFs are designed in an LSSD fashion, with separate clocks driving the master and slave latches as shown in Figure 5.13. The dual clock configuration allows a long delay chain to be created by setting both clocks high. Since each FF has two pass-gates and two inverters in series, the delay chain is effectively 48,000 gates long (12,000 FFs x 4 gates/FF).

A sequence of experiments were carried out in which a rising edge is launched into the scan chain input at time t_0 = 0 ns and at a time t_x, the A clk is de-asserted to stop the propagating edge. With the scan chain initialized to all 0s, the number of 1's captured in the scan chain indicates how far the edge propagated over the t_x time interval. We conducted a sequence of experiments on each of the chips in which the launch/capture delay t_x was varied from 500 ps to approx. 1200 ns in 5 ns intervals, i.e., approx. 240 experiments were carried out per chip.

The analysis shows that the chip-to-chip variation in delay along the entire chain is approx. 15% (data not shown). However, **within-die** variations are much larger,

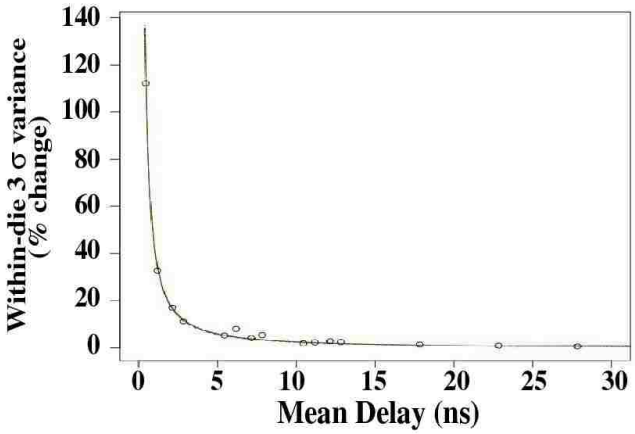


Fig. 5.14: Within-Die Variation Expressed as % change against Mean Path Delay(ns) for a set of 65nm Test Chips

particularly along shorter segments of the delay chain, as shown in Figure 5.14. Here, the mean delays along various segments of the delay chain are computed using the 18 chips and are plotted along the x-axis. The 3σ variations in the delays of these path segments are plotted along the y-axis as percentage change. For example, the within-die variations for the 5 ns segment are approx. 10%, i.e., delays in the range of 4.5 to 5.5 were measured across the 18 chips. The trend in the data is captured by the superimposed exponential curve. For path delays of 500 ps (approx. 6 master-slave FFs as shown in Figure 5.13), the % change increases to approx. 100%. Although the chip-to-chip variation is only slightly larger than the 11-12% observed in the 130 nm FPGA technology, this analysis shows that the regional, within-die variation is significantly larger and needs to be accounted for in Trojan detection methods.

5.3 Trojan Analysis

A second important objective of this work is to determine the impact that Trojans have on power and delay, and to investigate analysis methods that are sensitive to small

anomalies in these parameters. As discussed earlier, an FPGA platform is used to model an ASIC because of the flexibility it provides in re-configuring the logic to model different Trojan scenarios. We recognize that the current and frequency characteristics of equivalent ROs in an ASIC will be different, i.e., smaller power consumption and faster in frequency (as demonstrated in the previous section for delay), but so would the power and frequency anomalies that are introduced by the modeled Trojans. Therefore, we believe our experimental approach and analysis, although an approximation, will scale to ASICs.

The within-die and chip-to-chip variation analysis carried out in the previous section is essential for determining the types of analysis methods that may be effective in detecting Trojan anomalies. Methods based on power and delay analysis are inherently statistical in nature. The bounds of the statistical limits that separate Trojan-free and Trojan-implanted chips must be derived from the level of measurement noise and process variations present in the chips.

5.3.1 Statistical Analysis Methods

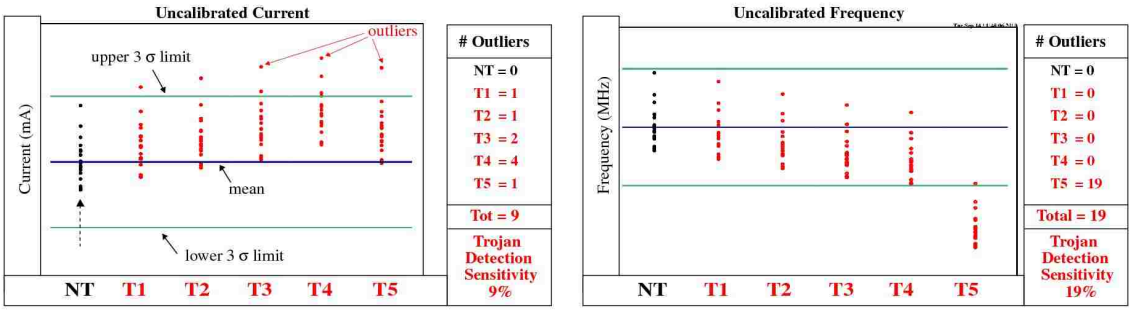
In this section, we describe several statistical methods designed to detect the emulated Trojans. A simple 1-dimensional (1-D) statistical method is used to analyze RO current and frequency data in isolation. A 2-D method, called regression analysis, is used to analyze current and frequency together, to determine the advantage, if any, of combining both parameters for detecting Trojans. Last, a calibration method is proposed that is designed to further improve the detection sensitivity of the statistical methods.

5.3.1.1 1-D Analysis

A 1-D statistical analysis is carried out using the currents and frequencies from RO_0. Recall that RO_0 is used to emulate each of the 5 Trojans in separate hard-macro designs of the RO, as shown in Figure 5.6. A set of 3σ statistical limits are computed from the current and frequency data measured with each of the 20 FPGAs configured with the Trojan-free version of RO_0. The plots in Figure 5.15 show the Trojan-free data on the left side as a series of 'vertically stacked' data points, called a line graph, for current (a) and frequency (b). The line graph of the Trojan-free data is labeled with 'NT' on the x-axis. The y-axis plots current and frequency, respectively. The 20 data points in the NT line graph are used to define three lines labeled *mean*, *upper 3σ limit* and *lower 3σ limit*. The factor of 3 is used to scale the standard deviation (s) such that 99.73% of the sample population is included between the limits. The value of 3σ is commonly used in industry as a statistical limit.

The line graphs labeled 'T1', 'T2', etc. depict the data from the Trojan experiments. Several trends are notable. The currents are larger in the (a) plot for all Trojans, in comparison to the NT currents. Also, there is an incremental increase in current for the trigger Trojans T2, T3 and T4 over T1, as expected given the increase in capacitive load added to the RO by these Trojans. Payload Trojan T5, implemented as 2 additional series-inserted inverters in the RO, also increases the current over the NT case. The frequency plot (b) shows the opposite trend, i.e., the trigger and payload Trojans increase delays and reduce the frequencies of the ROs. It is also clear that the delay added by the payload Trojan is more significant than that added by the trigger Trojans.

Given the large dispersion in the Trojan-free data points, only some of the Trojans



(a) (b)

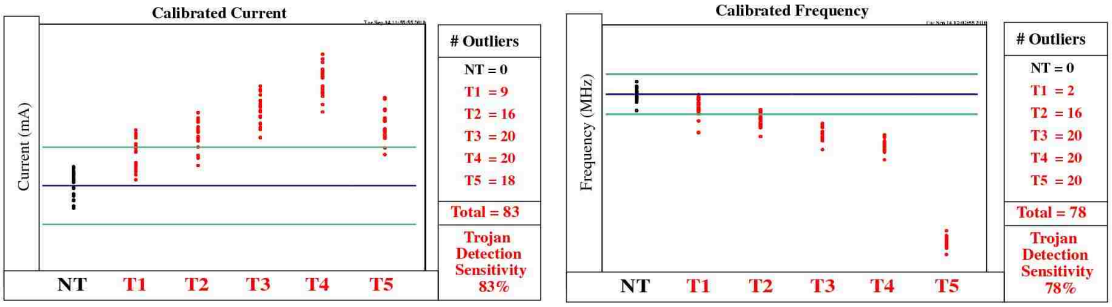
Fig. 5.15: Uncalibrated current and (b) frequency 1-D statistical analysis using 20 Trojan-Free RO0 values (NT), and 20 RO values from each of the 5 Trojans (Tx).

are actually detected. We consider a Trojan detected if its data point falls above or below the 3σ limit lines. The number of detections for each Trojan are tabulated on the right side of the plots. For example, none of the trigger Trojans, T1 through T4, are detected in the frequency analysis.

5.3.1.2 Calibration

The large dispersion in the Trojan-free (NT) data points in Figure 5.15 is caused by measurement noise and chip-to-chip process variation effects. The overall effect of these noise sources is to reduce the sensitivity of statistical methods to Trojan current and frequency anomalies. Calibration can be used to reduce chip-to-chip (and within-die) process variation effects.

We propose a ‘regional’ calibration technique that uses the current and frequency measured from a ‘calibration RO’ to help compensate for process variation effects. This type of calibration assumes that a distributed set of ROs are embedded into the product chip. As discussed earlier, embedded ROs are increasingly inserted into ASIC designs for monitoring process variations and wear-out effects, and therefore, our proposed scheme



(a) (b)

Fig. 5.16: (a) Calibrated current and (b) frequency 1-D statistical analysis using 20 Trojan-Free RO0 values (NT), and 20 RO values from each of the 5 Trojans (Tx).

simply leverages these existing resources. The basic idea is to compute a current or frequency ratio using, in our case, the reference RO_0 and a 'calibration RO' that is located in the same region. In an ASIC, the measured current or path delay from an applied logic test would be used instead of the values measured from the reference RO. In either case, the ratio is used to calibrate for within-die and die-to-die variations in current and frequency.

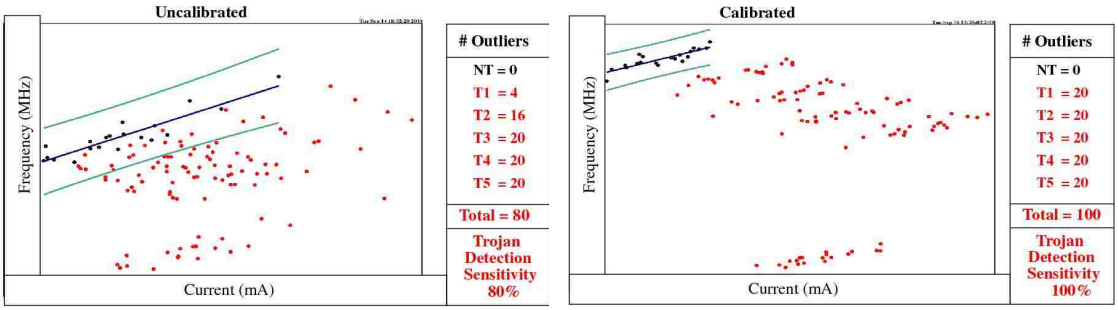
In our experiments, we partition the FPGA layout as shown in Figure 5.5 into 4 regions, each of which contains 8 of the 32 ROs. Our regional calibration method uses one of the ROs in each region for calibration. RO_3 is chosen as the calibration RO for calibrating current and frequencies for circuit activity that occurs in the upper left region of the chip, which is the region in which the emulated Trojans are inserted. The calibration process involves computing the ratio of the current (or frequency) generated by RO_3 on a reference chip (B_1 is used in our experiments), to the values measured from RO_3 on each of the remaining chips. These ratios are used to multiply the currents (or frequencies) measured from other 'regional' RO experiments on these chips, in particular, those from RO_0.

Figure 5.16 shows the data from Figure 5.15 after the calibration process is performed. It is clear that the dispersion in the Trojan-free data (and the corresponding limits) is significantly reduced, i.e., the 3σ limit lines are closer together. The number of detection also increases significantly over the uncalibrated data analysis, as given by the tabulated results on the right side of the plots. For example, total detections increase to 83 and 78 for current and frequency analysis, respectively, over the uncalibrated results given in Figure 5.15 as 9 and 19.

5.3.2 Regression Analysis

A second statistical analysis method that we investigate is regression. Regression analysis is carried out on a scatter-plot that contains data from two chip parameters. For our experiments, we plot frequency against current, as shown in Figure 5.17, by pairing the current and frequency values from RO_0 across the 20 chips. Since current and frequency are correlated, the data points from the Trojan-free experiments cluster around a line, called the regression line. A regression line is derived from the Trojan-free data points and represents the best fit line through them. A set of statistical limits can also be defined in an analogous fashion to the line plot analysis described earlier. The regression line and 3σ parabolic limits using uncalibrated and calibrated data are shown in the (a) and (b) plots of Figure 5.17, respectively. Trojan detection is also defined in an analogous manner as data points that fall above or below the limits.

Observations similar to those given earlier can be made regarding the dispersion of the data points for both plots. The main point to this analysis is to determine if Trojan detection sensitivity is better when both current and frequency information is used



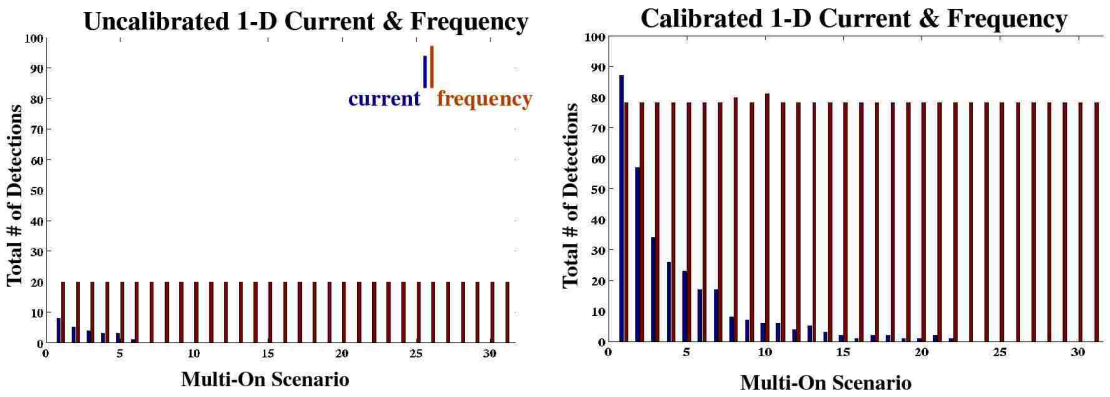
(a) (b)

Fig. 5.17: (a) Uncalibrated and (b) calibrated regression analysis using 20 Trojan-Free RO_0 values (NT), and 20 RO values for each of the 5 Trojans (Tx).

together (as opposed to being used alone as in the 1-D analysis). This is clearly true in the uncalibrated data analysis where the number of total detection increases from 19 (the larger of the two values in Figure 5.15, i.e., from the frequency data) to 80, as given in the (a) plot of Figure 5.17. Using calibrated data, the detection sensitivity increases further, from 83 as given in the (a) plot of Figure 5.17 to 100, i.e., all Trojans detected, in the (b) plot of Figure 5.17.

5.3.3 Multi-On RO Experiments

The results from the current analysis of the RO_0 experiments described in previous sections are best case in the sense that only one RO is enabled. In an actual ASIC application, it is extremely difficult or impossible to control logic activity such that only a single path is sensitized (propagates a edge). It is far more common to apply a test that causes multiple paths to propagate edges. Therefore, the single RO experiments are not representative of an actual testing scenario. In this section, we investigate Trojan sensitivity using 1-D and regression analysis when more than one RO is enabled. We call these scenarios Multi-On.

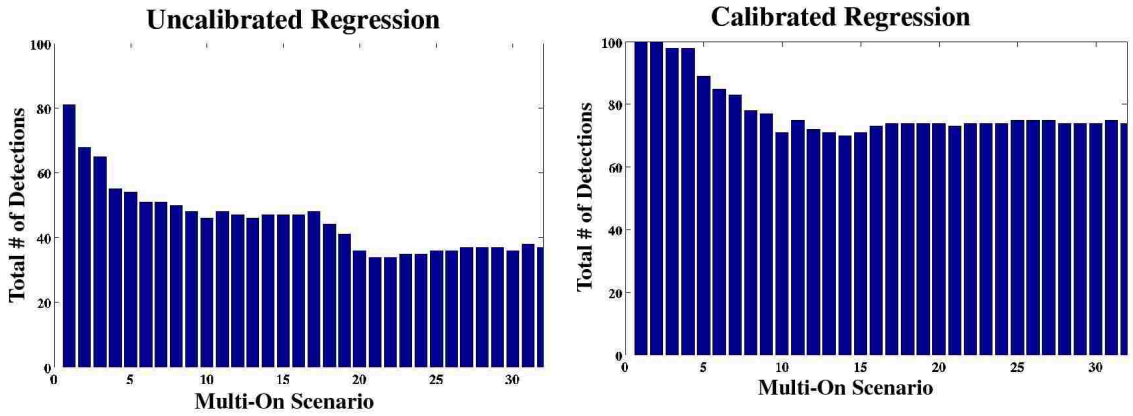


(a) (b)

Fig. 5.18: (a) Total Trojan detections using uncalibrated and (b) calibrated current and frequency data for Multi-On experiments.

The Multi-On experiments are numbered from 1 to 32 to indicate the number of simultaneously enabled ROs. RO_0, which contains the Trojan in the Trojan experiments, is always enabled. Multi-On scenario 2 enables RO_0 and RO_1 (see Figure 5.5 for RO positions). Multi-On scenario 3 enables RO_0 through RO_2, etc. As more ROs are enabled, the level of 'background' current increases. We define background current as the current generated from the additional ROs, i.e., RO_1, RO_2, etc.

Figures 5.18 gives the total number of detection in a bar-graph using uncalibrated (a) and calibrated (b) data for the 1-D analysis method. The Multi-On scenario is given along the x-axis. The results for current and frequency are shown as adjacent bars under each scenario. The number of detection goes to 0 using uncalibrated current for Multi-On scenarios larger than 6, which illustrates that current sensitivity drops dramatically as the background current increases. The same behavior is observable under the calibrated current analysis (Figure 5.18(b)), except that Trojans continue to be detected up through Multi-On scenario 22. Intuitively, the Multi-On scenarios have a much smaller impact on the frequency analysis because the frequency anomalies introduced by the Trojan remain



(a) (b)

Fig. 5.19: (a) Total Trojan detection using uncalibrated and (b) calibrated data under regression analysis for Multi-On experiments.

distinct and relatively independent of the amount of additional switching activity.

The regression results are shown in Figure 5.19. Only one bar is present for each Multi-On scenario because regression uses both current and frequency data as described earlier. The total number of detection are larger than zero across all Multi-On scenarios using the uncalibrated data (plot (a)) because frequency remains effective under the Multi-On scenarios. A similar trend is observable in the (b) plot in Figure 5.19 which shows the regression results using calibrated data. The average height of the bars is larger, however, confirming again that calibration has a significant impact on detection sensitivity. Interestingly, the regression analysis using calibrated data (Figure 5.19(b)) is slightly worse than the calibrated frequency data (Figure 5.18(b)) for several of the Multi-On scenarios in the range of 10-15. The difference is small but illustrates that the additive effect of noise from two data sets, as is true for the regression analysis, can reduce sensitivity.

5.4 Discussion

It is clear from the results presented in Section 5.3 that a high resolution analysis of current and delay, combined with a calibration method to attenuate the adverse effects of within-die and die-to-die process variation effects, has the potential of providing high sensitivity to the small current and delay anomalies introduced by Trojans. Our experiments were designed to determine the advantages of such a strategy under highly controlled conditions. Unfortunately, in practice, it is not possible to obtain high resolution measurements of current and delay without some type of support infrastructure on-chip and/or in the manufacturing test environment. In this section, we describe test strategies and an on-chip support infrastructure that can be used to accomplish these goals.

In a previous work, we proposed a multiple supply port method (MSP) as a means of obtaining high resolution transient current measurements [33]. Subsequently, we also demonstrated in hardware experiments on a 65nm test chip a similar strategy using leakage current (I_{DDQ}) [34]. However, as shown in this work, transient current analysis becomes increasing less effective as the number of simultaneously exercised paths increases. This is particularly evident for global current measurement methods, but MSP will be adversely impacted as well. Therefore, maintaining the effectiveness of transient current methods requires a special form of automatic test pattern generation (ATPG) that is designed to target specific paths and minimize transient ‘background’ noise, i.e., the total number of simultaneously exercised paths. We describe a method for achieving this in a recent work [49].

Also as demonstrated in this work for some of the Multi-On scenarios, the highest

level of Trojan detection sensitivity is achieved by combining current and delay analysis. Unfortunately, conventional delay test methods used in manufacturing test cannot provide high resolution measurements of delay of individual core logic paths. This is true because the launch/capture timing used in delay tests is fixed throughout the test application process and therefore, only an ‘upper’ bound on the delay of **all** tested paths is determined. Although modifications of the test application process have been proposed where the launch/capture timing is changed dynamically and repeatedly for each test pattern until tight upper bounds are determined for **each** of the tested paths, such methods are difficult implement in practice because of cost constraints imposed in manufacturing test, the limitations of automatic test equipment (ATE), masking of invalid bits, etc. We believe the same problems will exist for the process of detecting Trojans using delay tests.

The test generation (ATPG) strategy that we propose is based on the transition fault model commonly used by the manufacturing test community for delay faults. The transition fault model assumes a defect introduces a ‘lumped’ delay, as opposed to the path delay fault model which assumes a defect, and its corresponding delay, is distributed along the entire path. The lumped delay model is a better match for the capacitive loading and inserted gate delays introduced by Trojans.

There are several major benefits of the transition fault model. First, the lumped delay assumption allows ATPG to target tests for nodes, instead of paths, reducing the required number of tests to just 2 times the number of nodes, i.e., a rising and falling transition test for each node. Moreover, given that the number of paths is much larger (exponential to the number of nodes), each node can be tested along multiple different

paths, making it more difficult for adversaries to ‘hide’ the additional delay by reducing delays in upstream and downstream gates.

Given these limitations, a better solution to obtaining high resolution delay information is to incorporate an on-chip infrastructure to support it. The authors in [50] describe one such infrastructure, which makes use of shadow registers to capture subtle delay variations in paths introduced by Trojans, and analyze its effectiveness using simulations on a variety of process models. We describe here a novel infrastructure designed to accomplish this goal, and which attempts to do so in a more cost effective manner, using an infrastructure with a smaller footprint and compatibility with existing ATE capabilities.

CHAPTER 6

REBEL for Hardware Trojan Detection

The experimental results presented in Chapter 5 shows that Trojans, that manifest themselves as additional load capacitance or additional series gates, can be detected using the path delays signature of the chip and the detection sensitivity can be improved by statistical techniques. Also the simulation results in Chapter 3 and hardware results in Chapter 4 shows that the proposed Embedded Test Structure, REBEL, can be used to measure these path delays accurately. These path delay measurements can be compared with the delay signature of the "Golden Model" to detect the presence of Trojans in product chips.

In this chapter, I present REBEL's capability to detect the subtle delay anomalies introduced by hardware Trojans. As explained in section 3.7, REBEL is integrated with a 32-bit pipelined floating point unit (FPU) which has a scan chain length of 671 bits. The scan chain is divided into 28 segments in order to enhance the coverage achievable for path delay ATPG. In addition, several copies of the Trojan emulation circuit, which are designed to introduce delay anomalies, are inserted into the FPU macro at the structural description level. As explained in section 3.7, the structural description is synthesized to a layout using the Cadence Encounter Place and Route tool. The final design is fabricated in IBM's 90nm CMOS bulk process. The data collected from 62 copies of the chip is used to show REBEL's ability to detect hardware Trojans.

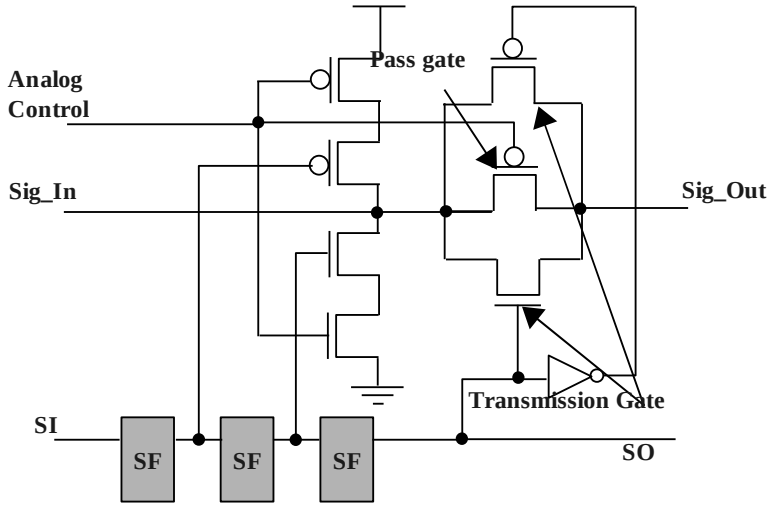


Fig. 6.1: Trojan Emulation Circuit

6.1 Trojan Emulation Circuit

The Trojan emulation circuit is designed to introduce delay anomalies in the path that model the additional capacitive loading imposed by Trojan circuits. Fig. 6.1 shows the internal structure of the Trojan emulation circuit. The operation of the Trojan emulation circuit is controlled by three configuration registers. Setting the configuration registers to '101' disables the Trojan emulation circuit and represents the Trojan-free design. In this scenario, both the NMOS and PMOS transistors of the transmission gate are enabled. In order to emulate a Trojan, the transmission gate is disabled by setting the state of the configuration registers to '100'. With the transmission gate disabled, a resistive connection can be configured between Sig_In and Sig_Out by tuning the voltage on the analog control pin to a value above 0 V. The analog control pin connects to the gate of the PMOS pass transistor, so values above 0 V leave it only partially conducting. For example, setting the analog control pin to 0.6 V (one half of the supply voltage) will introduce a delay anomaly for a rising or falling edge that propagates though the cell. The

stacked PMOS and NMOS transistors in the left portion of the Trojan emulation cell can also be used in a similar fashion to create delay anomalies by setting the state of the configuration resisters to '111'.

6.2 Experimental Setup and Results

In this section, we present the results of applying our technique to the FPU macro on a set of 62 chips. The instrumentation shown in section 4.1.1 is used for this experiment also. The path delay or transition test vectors that sensitize the paths where the Trojan emulation circuits are generated using Cadence Encounter Test ATPG tool. These test vectors are post-processed to append the REBEL configuration information using a perl script.

The first step of the process involves applying a set of calibration tests to the FPU scan chain as a means of characterizing delays along its segments. The calibration data is used to compute the actual path delays from a subsequent set of transition tests applied to the core logic of the FPU. We apply two types of transition tests to the FPU, one set in which all Trojan emulation circuits are disabled and a second set that enables exactly one Trojan emulation circuit at a time. The delays measured under the Trojan-free tests are used to derive statistical limits. These limits are later used to classify the delays measured under the HT tests.

6.2.1 Calibration

As mentioned in section II.A, the calibration process for REBEL is designed to eliminate delay of the PUT's transition along the delay chain, as specified by T_{sc} in Eqn.

1. From preliminary experiments, we determined that the uncertainty in the delays measured in the calibration process grows as the edge propagates further along the delay chain. This uncertainty increases the measurement error in the calculated path delays. In order to minimize these measurement errors, we created a special set of transitions in the combinational logic for calibration. These special logic tests are designed to drive a transition into the first (left-most) flip-flop of each segment. This technique reduces the uncertainty in the delays by introducing a set of transitions that are derived from nearby logic. The different velocities associated with the propagation of a rising and falling edge in the scan chain require both rising and falling edge calibration tests.

In our experiments, we set the LC resolution to 25 ps. Once the calibration tests are performed, the delay along any portion of the delay chain can be computed as given by Eqn. 6.1. The expression gives the delay along the delay chain (T_{SC}) between two scan flip-flops FFe and FFs as the difference in the delays measured under the calibration tests to these scan FFs, i.e., T_{ffe} and T_{ffs}.

$$T_{sc} = T_{ffe} - T_{ffs} \qquad \text{Eqn. 6.1}$$

where, T_{sc} = Delay along the delay chain

T_{ffe} = LC period for the calibration edge to reach 'FFe'.

T_{ffs} = LC period for the calibration edge to reach 'FFs'.

The calibration process needs to be carried out first, and at the same resolution that is used for the logic tests. The time required to carry out calibration depends on the ength of the delay chain and the desired resolution, but in any case, it represents overhead that we would like to minimize. In the analysis of our chip data, we found that the calibration curves are similar in shape across the chips and that most of the differences

between them can be eliminated by multiplying them by a scaling factor. In other words, regional, within-chip variations are insignificant compared to the chip-to-chip variations in this technology. We realize that this may not hold true in more advanced technologies, and calibration may be needed for every chip. However, in cases where scaling is possible, this short-cut can be used to significantly reduce overhead.

We also investigate the elimination of the calibration process altogether. Technically, we do not need to know the actual path delays in order to detect anomalies introduced by HTs. However, including the delay along the delay chain in the statistical detection method (described below) acts to 'water-down' the anomaly introduced by the HT, and reduces detection sensitivity. We refer to 'uncalibrated' delays as delays that include the delay chain component in the following sections.

6.2.2 Noise Analysis

As described earlier, clock strobing is applied using incrementally longer LC intervals to determine the delays between elements of the delay chain (referred to as inter-scan-FF delays earlier). The LC interval in which a delay element is first able to record the transition is the interval used to determine the inter-scan-FF delay. However, this LC interval may vary from one sample to the next. The range over which it varies is referred to as *uncertainty*. Uncertainty reflects the noise level present in the measurements.

In our experiments, we found that the level of uncertainty varies for delay elements at different positions from the PUT's insertion point. Given that REBEL allows a 'target' delay element to be chosen, this characteristic of uncertainty can be leveraged as a means

of improving resolution. Fig. 4.2 plots uncertainties along the y-axis for each of the numbered delay elements along the x-axis. The delay elements beginning from the PUT's insertion point (the capture FF) are numbered 1 to 18 for each of the consecutive elements along the delay chain. Each of the curves represents the set of uncertainties associated with one PUT. The graph plots the results for 23 PUTs.

The wide range of uncertainties on the left side of the graph indicate that delay elements near the insertion point work well as a target for some PUTs but very poorly for others. This occurs because the voltage behavior on the outputs of some PUTs glitch, making it difficult to obtain a consistent measurement. On the other hand, the uncertainties for target delay elements in the center region of the graph, e.g., delay elements numbered 8 and 9, provide the lowest overall uncertainty across the PUTs. Uncertainties again increase for delay elements at larger distances from the insertion point because of power supply noise, signal coupling and other types of on-chip noise. Given these characteristics, we use the 9^{th} delay element to compute path delays.

The average uncertainty computed from all path delay measurements in our experiments is approx. 40 ps, and the worst case uncertainty is approx. 150 ps. The main sources of this uncertainty are meta-stability in the delay elements, power supply noise, clock glitter and temperature variations. The worst case uncertainty may be smaller for designs that generate the LC timing events on-chip.

6.2.3 Analysis of Process Variations

The path delays from a set of Trojan-free transition tests are used to analyze delay variations caused by within-chip and chip-to-chip process variations. Fig. 6.2 gives a

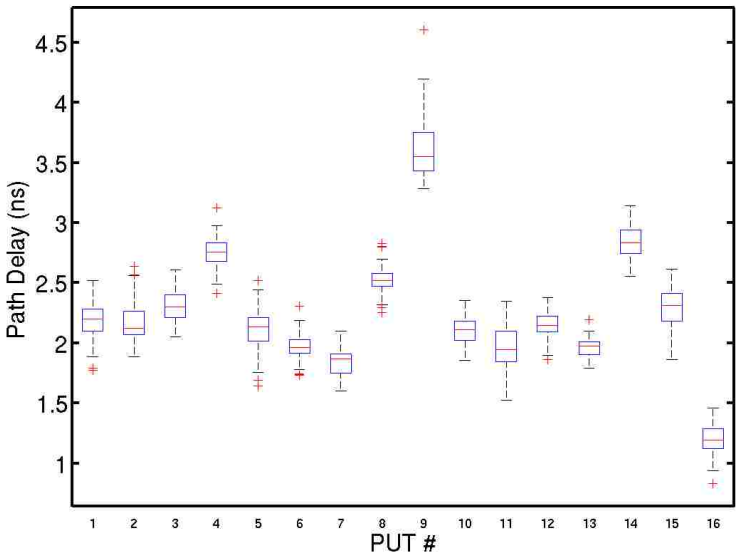


Fig. 6.2: Noise and Process Variations: Box Plot of 16 Different Path Delays among 62 Chips

boxplot analysis of PUT delays from 62 chips and 16 transition tests. The figure plots the PUT number on the x-axis against the measured delays on the y-axis. The boxplot associated with each PUT identifies the median, upper and lower quartiles as horizontal lines. A set of dotted lines extend to the largest and smallest delays and a '+' indicator identifies outliers in the distribution. As expected, chip-to-chip variation increases as the length of the path increases, e.g., the variation portrayed by the boxplot for PUT #9 is larger than that for PUT #10.

The curves shown in Fig. 6.3 plot PUT # against delay in the same manner as in Fig. 6.2, except this time, only 2 chips are shown and the delays for a given chip are connected together in a curve. The close match of the two curves indicates that these two chips have similar overall performance characteristics. Crossings that occur between the curves illustrate cases where within-chip variation and/or measurement noise changes the relative delay values of the PUTs. Although this occurs frequently in the curves, the magnitudes of the vertical deviations are very small, indicating that within-die variations

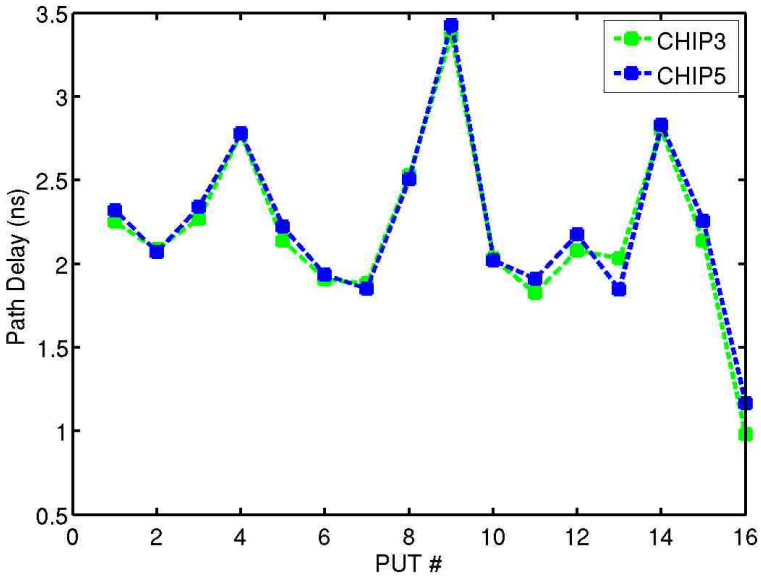


Fig. 6.3: Within-Chip Variation: CHIP3 and CHIP5

and noise are insignificant relative to the chip-to-chip variations depicted in Fig. 6.3. The statistical technique that we describe in the next section can deal effectively with chip-to-chip variations. However, within-chip and variations and noise act to reduce its sensitivity. The fact that they are small in this technology allows very small delay anomalies to be detected.

6.2.4 Emulated Trojan Characteristics

As indicated earlier, delay anomalies are created using HT emulation circuits, which are designed to model the capacitive loads HTs introduce on existing wires in the design. The magnitude of the delay anomaly is controlled using an analog voltage pin that connects to a PMOS passgate. Analog voltages above 0 V increase the on-resistance (R_{ONP}) of the PMOS passgate and increase delay according to Eqn. 6.2:

$$\Delta T \approx 0.7 * \Delta R_{ONP} * C_{Load} \qquad \text{Eqn. 6.2}$$

where, ΔR_{ONP} is the change in the PMOS on-resistance due to change in analog

voltage, and C_{Load} is the load capacitance.

We solve this equation using simulation experiments carried out on a extracted netlist representing a 'nominal' model of the FPU. The results shown in Table II give the additional resistance added to the path for the set of analog voltages listed in the left-most column. An approximation of the equivalent additional load capacitances can be obtained by plugging in the ΔR_{ONP} values from the table into Eqn. 6.2. The results are given in right-most column of Table 6.1. The small values of $\mathbf{\Delta C_L}$ in the top-most rows indicate that the Trojan emulation circuit is capable of introducing very subtle changes in path delays.

Analog Voltage Change (Δvg) (V)	**ΔR_{ONP} (kΩ) (appx.)**	**ΔC_L (fF) (appx.)**
0.1	1.4	1.0
0.2	2.1	3.0
0.3	3.0	5.0
0.4	4.5	7.0
0.5	7.0	10.0
0.6	18.0	17.0
0.7	162.0	25.0

Table 6.1: Analog Voltage Change and Corresponding Additional Load Capacitance

Fig. 6.4 shows the change in path delays due to various Trojans. Here, the capacitive load added by the Trojans is plotted along the x-axis and the percentage change in path delay due to the additional load is plotted along the y-axis. For this analysis, the average of path delays measured in 62 copies of the chip is used. From Fig. 6.4 it is clear that small additional loads cause little change in path delay. For example, in most paths with an additional load of 1 fF the percentage change in path delay is less than

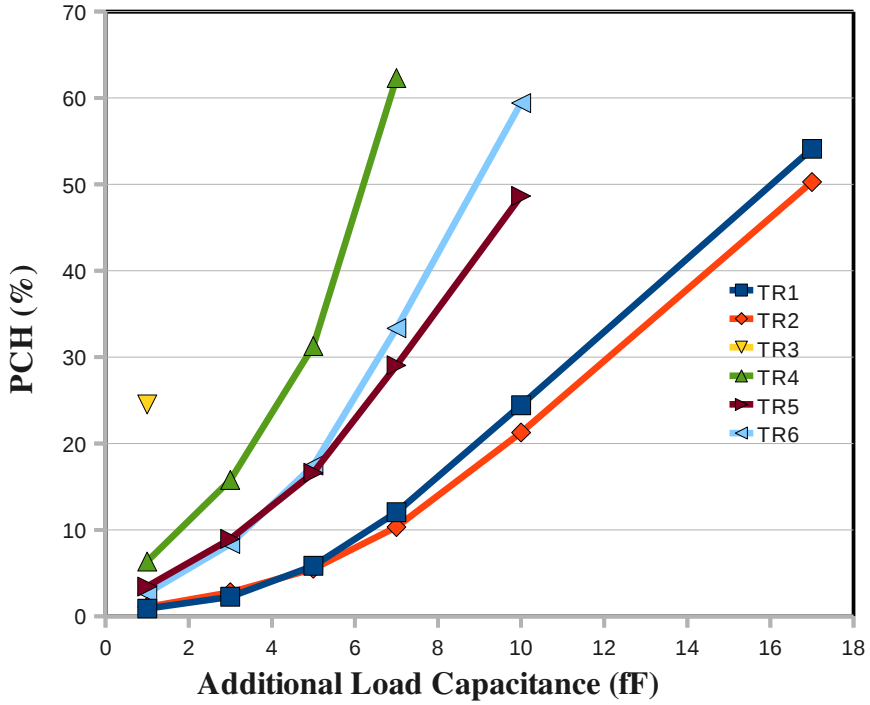


Fig. 6.4: Change in Path Delays for Various Additional Capacitive Load in 6 Different Paths

7%, while it is more than 20% for additional load of 10 fF. Moreover, the change in path delay varies for different paths due to different load capacitance of the trigger-nodes (fan-outs). The larger the trigger-node load capacitance, the smaller the change in path delay. For example, the percentage change in delay for the path TR5 (7 fan-outs) is smaller (9%) than that of TR4 (1 fan-out) (16%).

6.2.5 Regression Analysis

From section 6.2.3 it is obvious that the die-to-die variations are significant when compared with the noise and within-die variations. In this section, I propose a statistical technique, called regression analysis [51], that deals with the die-to-die variations effectively. Regression analysis operates on scatterplots where the delays from one PUT are plotted against the delays from a second PUT. Fig. 6.5 shows the regression

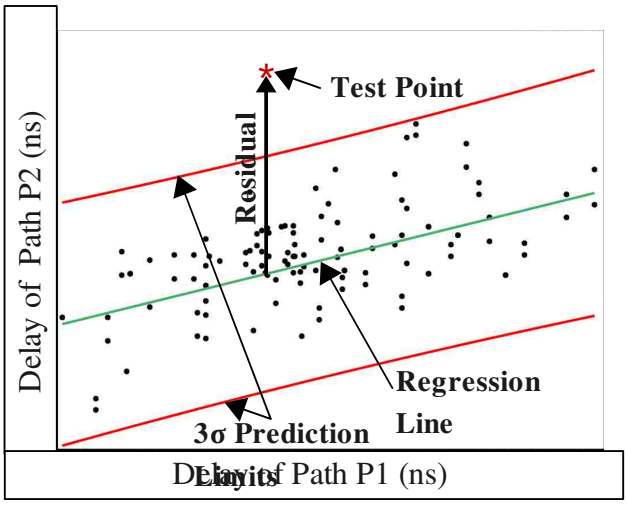


Fig. 6.5: Regression Analysis using Path Delay Values

analysis carried out on two path delays. The middle line is the 'regression line' or 'best-fit line' and the two lines on each side of the regression line are the 'prediction limits'.

Fig. 6.5 shows the regression line and prediction limits derived from two different path delays collected from 62 chips. The prediction limits are defined by eqn 6.3.

$$Y = B_0 + B_1 X \pm W * \sqrt{MSE} * \sqrt{1 + \frac{1}{n} + \frac{(X - \bar{X})^2}{\sum_i (X_i - \bar{X})^2}} \qquad \text{Eqn. 6.3}$$

where $W = \sqrt{2F(1-\alpha);2;(n-2)}$

$$\sqrt{MSE} = \sqrt{\frac{\sum_i (Y_i - \hat{Y}_i)}{n-2}}$$

The prediction limits are sensitive to both number of samples (1/n in eqn 6.3) and the amount of dispersion of the data points around the regression line (mean square error or MSE). These control limits are derived from control data, in this case Trojan-free data and the test data are tested against these limits and if the test data point falls outside the limits, it is classified as an anomaly. Fig. 6.5 shows one of the test point that falls outside

the control limits.

6.2.6 Calibrated vs. Uncalibrated Delays

We indicated earlier that calibrating path delays, i.e., removing the delay chain component, is optional but can improve detection sensitivity. Figs. 6.6 and 6.7 are used to illustrate this point using regression analysis.

The scatterplots of Figs. 6.6 and 6.7 plot the delays measured from two paths in the chips. The path plotted along the x-axis is a Trojan-free path while the path plotted along the y-axis includes an emulated HT circuit. Fig. 6.6 uses delays that are a sum of the delays from the PUT and delay chain ('uncalibrated') while Fig. 6.7 uses delays with the delay chain component subtracted (calibrated). Regression lines or 'best-fit' lines are derived using the Trojan-free data points in both figures. The delays representing the path plotted along the y-axis in this case are measured with the emulated Trojan circuit disabled. The curves labeled “3σ prediction limits” represent the expected variation in these paths. The expected variation accounts for within-chip process variations and noise. The delays from Trojan-free chips are expected to fall within the region delineated by these curves. Chip-to-chip process variations, on the other hand, are tracked along the length of the regression line and therefore, are effectively eliminated as detractors to HT detection sensitivity.

The plots of Figs. 6.6 and 6.7 also include data points with the emulated Trojan circuit activated using two different analog voltages (labeled as T1 and T2). The analog voltages used are equivalent to load capacitances of 10 fF (T1) and 17 fF (T2). The HT data points are depicted as shaded points in the figures, with the darker shaded points

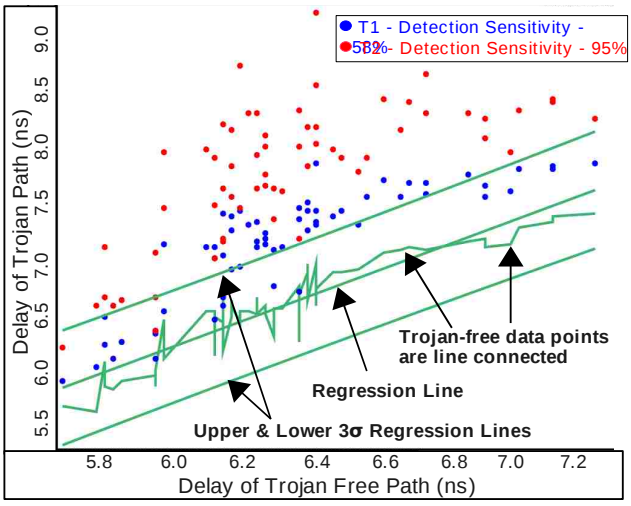


Fig. 6.6: Regression Analysis using Uncalibrated delays – 62 Trojan-free data points and 62 Trojan data points for each of the 2 Trojans

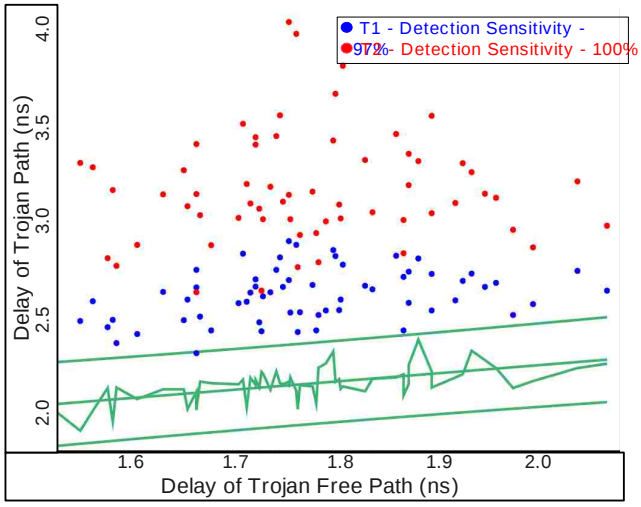


Fig. 6.7: Regression Analysis using Calibrated delays – 62 Trojan-free data points and 62 Trojan data points for each of the 2 Trojans

corresponding to T1. The impact of including the delay chain component can be evaluated by counting the number of HT data points that fall outside of the prediction limits in Fig. 6.6 and Fig. 6.7. The number of detections is larger in Fig. 6.7 than in Fig. 6.6, indicating that removing the delay chain component improves detection sensitivity. In particular, 97% of the T1 data points and 100% of the T2 data points are detected in Fig. 6.7, while only 58% and 95% are detected in Fig. 6.6.

Calibrated data improves sensitivity because the anomaly introduced by the HT is relatively larger with the delay chain component eliminated. Moreover, any delay variations that occurs within the delay chain due to within-chip process variations are also eliminated. The drawback of using calibrated delays is that the uncertainty in the measurements is effectively doubled because two measurements are used. For example, in our experiments, the single measurement uncertainty of 150 ps doubles to 300 ps.

6.2.7 Hardware Trojan Results

Path delay ATPG is used to derive tests for 6 of the embedded HT emulation circuits. Delays measured from these paths, with and without the HT enabled, are used in regression analysis, and detection statistics are computed using the set of test chips.

Trojan Detection Percentage (%)								
Path Name	Path Delay (ns)	ΔC_L (fF)						
		1	3	5	7	10	17	25
TP1	2.175	0	4	10	52	97	100	100
TP2	1.900	0	0	2	8	40	94	100
TP3	4.925	51	100	100	100	100	100	100
TP4	3.500	4	42	93	100	100	100	100
TP5	1.200	0	0	7	48	86	100	100
TP6	1.575	0	1	19	77	100	100	100

Table 6.2: Trojan Detectability For Various Trojans

The results are summarized in Table 6.2 for each of the HTs identified by path names TP1 through TP6. The second column gives the average delay of the path across the 62 chips. The remaining columns give the number of positive detections as a

percentage. The percentages are computed by counting the number of data points that fall outside the 3σ prediction limits in each of the six scatterplots. These numbers are then divided by the number of chips and multiplied by 100. Each emulated HT is tested at 7 different analog voltages. The analog voltages are translated into equivalent additional load capacitances using Eqn. 6.2, and are given in the header row of the table.

As expected, smaller capacitive loads are harder to detect. For example, a HT modeled as adding 7 fF to a node along the path named TP1 is detected approx. half of the time (52%), but increases to 97% when the additional capacitance increases to 10 fF.

6.2.8 Sensitivity Analysis

From Table 6.2, a relationship appears to exist between the smallest capacitive load that first produces positive detections and path length. For example, two of the shortest paths, TP5 and TP2, are not able to detect anomalies until the capacitive loads reach 5 fF while the longest paths, TP4 and TP3, are able to detect capacitive loads as small as 1 fF. This at first seems counter-intuitive because one would expect that short paths would be changed more dramatically by smaller capacitive loads than longer paths, under the condition that the delta increase in delay introduced by the capacitive load is constant and independent of path length. For example, if the capacitive load of a HT increases the delay of a path by 1 ns, then the delay of a short path with nominal delay of 2 ns is increased by 50%, whereas the delay of a longer path with nominal delay 4 ns is increased by only 25%.

The reason the opposite appears to hold in our data is rooted in the "averaging" effect that occurs for longer paths. Random with-chip variations in the gate delays of

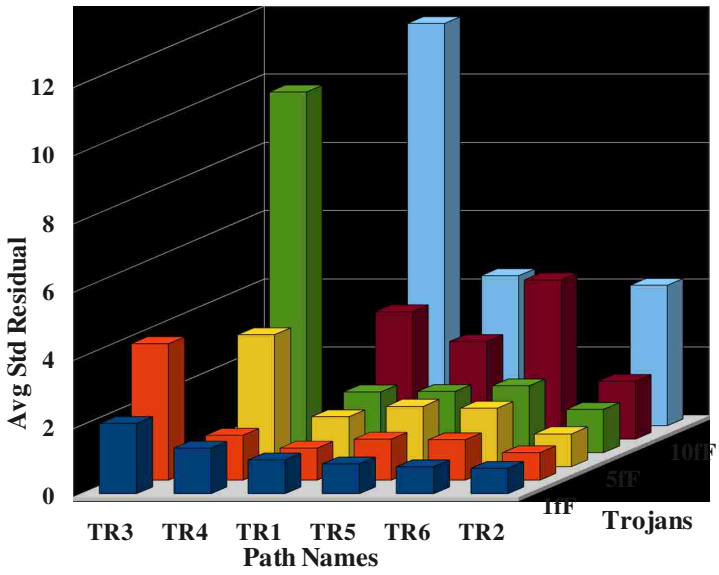


Fig. 6.8: Detectability Analysis using Standardized Residual Values for Various Trojans

longer paths tend to average out and exhibit lower levels of overall path delay variation in the chip population. This is reflected by the position of the statistical limits in our analysis. In particular, we found that the distance of the statistical limits from the regression lines for longer paths is smaller than it is for shorter paths.

The metric based on which Trojan detection decision is made is 'residual' as shown in Fig. 6.5. A residual is defined as the vertical distance from the data point to the regression line. For the Trojan-infected devices the data points are expected to have residual larger than the distance between the regression line and a chosen prediction limit. Also larger the residual values greater is the confidence that the device is Trojan-infected.

Although we count the number of points that fall outside the prediction limits to analyze the detectability (as shown in Table III), it is also meaningful to examine the magnitudes of the residuals for various Trojan scenarios. In order to make these residual values more meaningful for comparisons with other Trojan-infected paths and to analyze

the various factors that affect the detectability, the magnitude of the residuals are normalized or standardized using eqn 6.4.

$$ZRES = \frac{residual}{\sqrt{MSE}} \qquad \text{Eqn. 6.4}$$

Here, MSE is the variance of the Trojan-free data point residuals.

This standardized residual value is computed for both control data points (Trojan-free data) and the test data points (Trojan data points) and compared. A larger value of standardized residual for the Trojan test points indicates that some or all of the residuals in the set of scatter plots is large, as the one shown in Fig. 6.8, and the presence of Trojan can be easily identified.

6.2.9 Test Time Analysis

In this work, we test only a small number of paths in the FPU. In any practical application of this technique, a much larger number of paths would need to be tested to gain sufficient confidence that the chips are free of HTs. Test time then becomes an important issue to consider because it relates to the cost effectiveness of the technique. The clock strobing technique discussed in the previous section can be very expensive in terms of test time if it is applied in a step-wise linear fashion. The temporal visibility that REBEL provides allows a much faster search process to be carried out to find the LC interval that achieves the goal of propagating the edge to a target delay element (as we discussed in Section III). The LC interval can be 'intelligently tuned' based on the results of the previous application of the test sequence, which can dramatically reduce the number of repeated applications of the test sequence.

A second cost cutting measure is to eliminate the calibration process. However, as we discussed in the previous section, calibration improves detection sensitivity. Therefore, a trade-off exists in this case. Bear in mind that the HT detection problem, unlike the defect detection problem, can leverage parallelism by applying different subsets of the test sequences to separate chips simultaneously. This can be very effective at reducing test costs but only works if we can assume that every chip is identical, i.e., the HT is not selectively inserted into only a subset of the chip population.

6.3 REBEL-Integrated BIST System For Trojan Detection

In order to analyze the test effort of the proposed Trojan detection technique, it is integrated with a BIST system. The BIST system incorporates a pseudo random test pattern generator (PR-TPG). In this section, I present the Trojan detectability of REBEL integrated with a BIST system and analyze the test effort required to achieve it. In the previous section, Trojans that add capacitive loading to the existing wires are investigated. In this experiment, I investigate hardware Trojans that manifest as additional series gate(s) to the existing gates, called 'payload' Trojans.

6.3.1 Trojan Emulation Circuit

In order to evaluate the payload-Trojan scenario, a different kind of Trojan emulation circuit is designed. Fig. 6.9 shows the 'payload' Trojan emulation circuit. This design is fairly simple which uses a multiplexer to select between a Trojan-free path (original design) and a Trojan path (adds a buffer to the path). The select signal of this multiplexer, 'TrEn', that is connected to a primary input of the design, allows for creating a Trojan-free scenario (when set to '0') and a Trojan scenario (when set to '1'). In addition

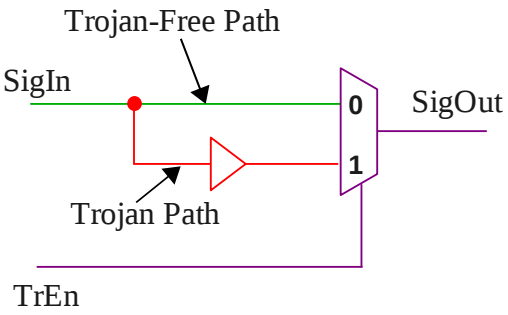


Fig. 6.9: Payload Trojan Emulation Circuit

to the buffer in the Trojan path, there are some routing changes and the delay change due to this is fairly small. This Trojan emulation circuit is designed as a hard-macro using the FPGA editor tool and is inserted at the HDL description of the design.

6.3.2 Experimental Design

In order to analyze the detection capability of the proposed technique in a real application, it is integrated into a pseudo random built-in-self-test (PR-BIST) system. Fig. 6.10 shows the block diagram of the REBEL integrated PR-BIST system. This is similar to the system used in Section 4.3.1 . The circuit-under-test (CUT) is a REBEL-integrated, pipelined AES encryption engine (1 round). The MUX-D-based REBEL structure is integrated into the HDL description of the AES macro. In addition, a Trojan emulation hard-macro that incorporates a 'payload' Trojan (explained in previous section) is inserted into the AES design. The Trojan-free and Trojan scenarios can be created by setting the Trojan enable pin to '0' and '1' respectively.

The test pattern generator (TPG) automatically generates pseudo-random test vectors using a linear feedback shift register (LFSR). The output response analyzer (ORA) analyzes the output of each test for validity, i.e., whether there is a transition at a specific insertion point for a given random test vector, and returns the response of a valid

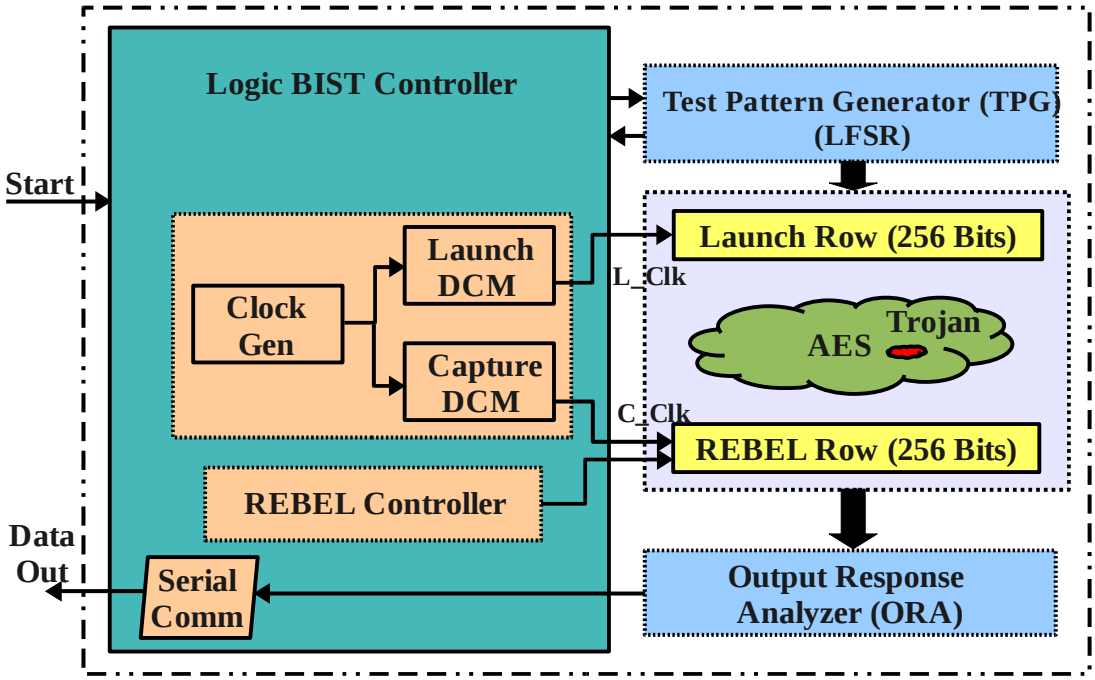


Fig. 6.10: REBEL-Integrated BIST System for Trojan Detection

test to the logic BIST controller. The logic BIST controller coordinates TPG, CUT, clocking scheme, and ORA.

The BIST controller first sends the control signals to the TPG which creates a random test vector and shifts it into the launch segment of the CUT. Then, it configures the REBEL structure for a specific path delay measurement using the integrated REBEL controller. Finally, it carries out launch-capture tests by sweeping the LC period until the transition propagation reaches a specific target flip-flop and returns the corresponding LC period value. It also invalidates the insertion points that do not have any transition for a specific random vector, returning the data for the valid transitions only. In this work, we used serial communication to send the test responses to the host computer and analyze the data for presence of Trojans using statistical methods.

The digital clock managers (DCMs) in Virtex2Pro FPGAs are used to generate the

launch and capture clocks for REBEL launch-capture tests. In addition to the clock-divider feature, DCMs allow for creating specific phase shift to the clocks. The launch clock is generated without a phase shift and the capture clock is generated with a phase shift tuned based on the required LC period. The BIST controller tunes the LC period for each LC test based on the response of the previous test and stops the path delay measurement when the transition reaches a specific target flip-flop.

6.3.3 Experimental Results

In this section, we present the results of applying our technique to the AES macro, that is integrated with a PR-BIST system, on a set of 29 FPGAs. As explained in Section 6.2.6 , the uncalibrated delay values can be used to detect Trojans. It eliminates the calibration process altogether resulting in significant reduction in test time. For all the analysis presented in this section, the uncalibrated delay values are used.

Similar to the noise analysis explained in Section 4.1.2.2 , the delay values are measured 10 times (samples) and the uncertainty among the samples are analyzed. In this experiment, the paths that have hazards are invalidated by the ORA resulting in measurement of uncertainty caused by the clock-jitter and meta-stability. A total of 2967 path delays are analyzed and the standard deviation of delay among the 10 samples are computed. The mean of these standard deviations results in 31 ps and a 3σ of 93 ps. Therefore, it is obvious that the uncertainty (or measurement noise) is reduced significantly while using the on-chip clock schemes.

6.3.3.1 Trojan Detection Analysis

As indicated earlier, the Trojan emulation circuit allows for creating Trojan-free scenario by setting the Trojan enable signal (TrEn) to '0'. Applying the pseudo-random vectors, several paths delays are measured to derive the statistical limits. Linear regression analysis is used to deal with the die-to-die variations. Regression analysis is carried out on the Trojan-free data points and the '3σ prediction limits', that represent the expected variations, are derived.

Parameter	Value
No. of Test Vectors Applied	50
No. of Valid Path Delays Measured	2967
No. of Trojan-infected Paths	60
No. of Paths that shows 100% Detection Sensitivity	34

Table 6.3: Trojan Detection Test Results

The 'payload' Trojans are enabled by setting Trojan enable pin to '1', which effectively inserts an additional buffer into the path. Using the same seed, the same set of random test vectors are generated and the path delay measurements are carried out with the Trojan enabled. The Trojan data points are plotted against the prediction limits and the detection decisions are made. Table 6.3 shows the test results. A total of 50 random test vectors are applied and 2967 valid path delays are measured. Among these delays, about 60 path delays shows the delay anomaly due to the Trojan. About 34 paths show 100% Trojan detection sensitivity, i.e., all Trojan data points fall outside the prediction limits. This explains that the proposed technique is effective for detecting 'payload' Trojans also. The main reason for the reflection of delay anomalies in several paths is the nature of SBOX component of the AES macro. However, the number of Trojan-infected

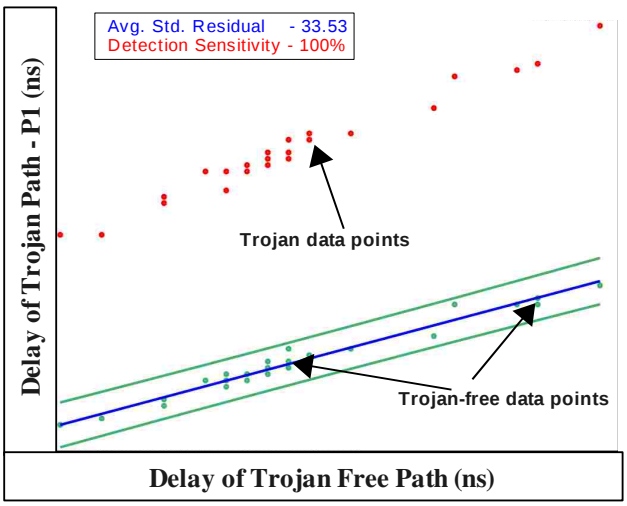


Fig. 6.11: Regression Analysis on a Trojan-Infected Path that shows Higher Detection Sensitivity

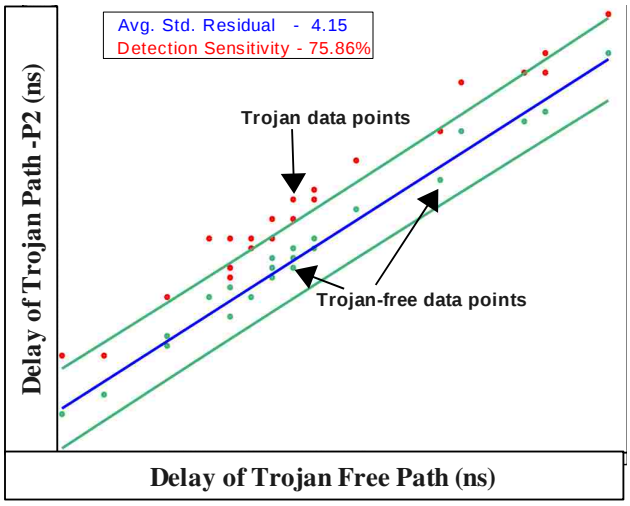


Fig. 6.12: Regression Analysis on a Trojan-Infected Path that shows Lower Detection Sensitivity

paths may be fewer for other designs.

Fig. 6.11 and Fig. 6.12 shows the regression analysis on Trojan-infected paths that show higher detection sensitivity (P1) and lower detection sensitivity (P2) respectively. The reason for this difference in detection sensitivity for different paths is due to the path length and the variation in these paths. Longer paths tend to have lower variations due to 'averaging-effect' and result in tighter prediction limits, which effectively result in higher detection sensitivity.

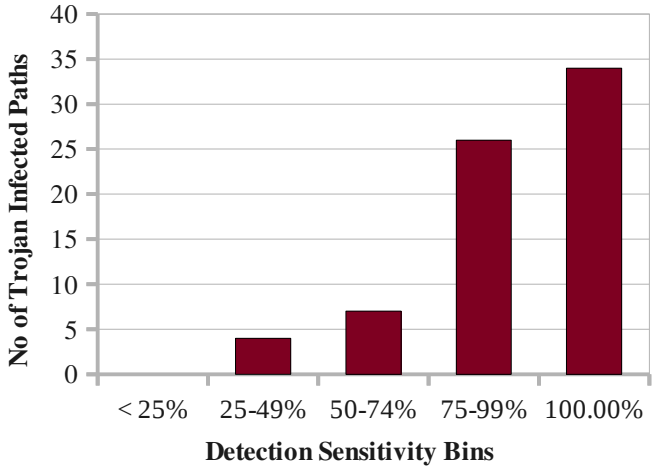


Fig. 6.13: Trojan Detection Sensitivity Distribution

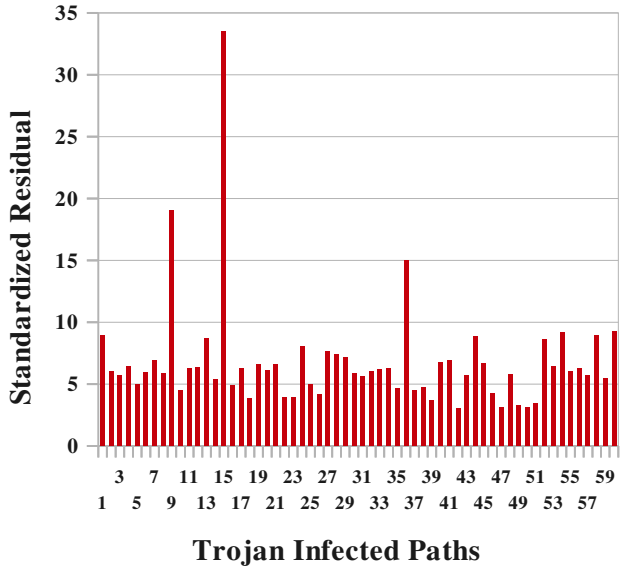


Fig. 6.14: Residual Analysis of Trojan Infected Paths

The Trojan-infected paths are binned based on the detection sensitivity. Fig. 6.13 shows the detection sensitivity distribution of the Trojan-infected paths. From this figure, it is clear that the number of paths that show 100% detection sensitivity is higher than the number of paths in any other bin.

6.3.3.2 Residual Analysis

The residual analysis explained in Section 6.3.3.2 is carried out for all the Trojan-

infected paths. The mean of the standardized/normalized residual values of all the Trojan data point is computed and shown in Fig. 6.14. The Trojan-infected path numbers are plotted along the x-axis and the average standardized residual values are plotted along the y-axis. Although the detection sensitivity is not 100% for all the paths, almost all of the paths show a average standardized residual value greater than 3 (σ). This provides higher Trojan detection confidence.

CHAPTER 7

Future Work

7.1 Clock Scheme Improvements

From the results presented in section 4.3.2.1 , it is clear that the on-chip clocking scheme improves the measurement quality by reducing the clock jitter. In the next generation of ASICs an on-chip clock scheme, like PLL, can be implemented to improve the measurement quality. As a result, the accuracy of process variability characterization can be improved.

For all the path delay measurements presented in this work, the linear clock strobing mechanism is used, which is expensive in terms of test time. Intelligent clock strobing mechanisms can be implemented to significantly reduce the test time. For example, after the first LC test, a binary search based clock strobing mechanism can be used to identify the LC period when the transition reaches the next flip-flop. Considering a worst-case inter-scan-FF delay of about 400 ps and a measurement resolution of 25 ps, linear clock strobing may require 16 tests for each path delay measurement, where as the binary search based clock strobing will be able measure with only 4 tests resulting in fourfold improvement in the test time. The on-chip clock schemes should be designed in such a way that these kinds of intelligent clock strobing mechanisms are incorporated. These intelligent clock strobing mechanisms will result in significant reduction in test time, an important parameter for the complexity of modern designs.

7.2 Systematic Variation Analysis and Model-to-Hardware Correlation Improvement

Although I used the path delay data for analyzing the magnitude of die-to-die and within-die variations, the silicon path delay data can also be used to analyze the systematic variations, including variations imposed by a specific gate, layout style, etc. Visual analysis of these systematic variations may be very difficult or almost impossible, in which case data mining algorithms can be developed to analyze the systematic variations. The data from this analysis can be fed back into the design flow and corrective measures, such as changing the standard cell design, changing layout style, etc., can be taken to reduce the variation magnitudes.

Moreover, the silicon data can be used to tune the variation models used by designers to reduce the gap between models and hardware. In order to use the silicon path delay data for tuning the models, a data-mining based system has to be developed to analyze the data. This system can improve the model-to-hardware correlation and improve the yield ramp.

7.3 Hardware Trojan Detection

In this work, the golden model is derived from the hardware itself by disabling the Trojan emulation circuit. However, in real scenario, it is hard or impossible to identify the Trojan-free hardware, in this case the golden models should be developed from the simulation experiments. While using the simulation data as a golden model, there may be false alarms due to the mismatch between the models and actual silicon. The proposed Trojan detection methodology should be applied to the above explained scenario to

analyze the number of false alarms involved in it.

In the experiments presented in this work, the location of hardware Trojans are known. However, in a real scenario, we do not know the location of the hardware Trojans. In that case, we have to come up with a set of delay tests that would cover almost all of the nodes in the design. In future work, the algorithms that can give this test set will be analyzed.

7.4 Other Applications

REBEL can be used in many other contexts including small delay defect (SDD) detection, post-silicon design debug and physical unclonable function. The effectiveness of REBEL in these contexts can be analyzed and improvements can be made to the proposed embedded test structure based on the analysis.

CHAPTER 8

Conclusion

In this work, I proposed a Truly Embedded Test Structure, called REBEL, for accurate measurement of path delays in product designs and verified it on the silicon. As REBEL leverages the existing scan structures, it is minimally invasive and adds small overhead to the design. The design overhead is also negligible because the integration can be completely automated within the DFT synthesis flow. Moreover, the measurement resolution can be scaled down to any required values. In addition, REBEL allows for delay measurement of any path, both short and long, using functional or at-speed clock eliminating the need for costly test equipments and power supply noise due to high-speed clocks. These capabilities of REBEL allow it to be used in several applications including design-for-manufacturability, hardware security, defect detection and post-silicon design debug.

REBEL integration strategy for both LSSD and MUX-D style scan chains are proposed and verified in hardware. The CLSSD based REBEL structure is implemented in 90nm IBM CMOS technology and used for two applications: delay variability characterization and hardware Trojan detection. The MUX-D based REBEL is integrated into a BIST architecture and used to characterize the delay variability in 130nm FPGAs.

Since REBEL can measure the process variations in design-context and uses at-speed clock for measuring any path delays, it can provide accurate die-to-die and within-

die characterization. The proposed embedded test structure is used to measure the process variation in path delays of a FPU macro among 62 ASICs manufactured in 90nm technology. The results show that the die-to-die and within-die variations can be as large as 27% and 12% respectively. REBEL is compatible with various scan-based DFT structures and in this work it is integrated with random BIST architecture and used to measure the delay variability in combinational paths of an AES macro. The digital snap-shots obtained using REBEL are similar to the information provided by a logic analyzer, which can be used to analyze the static and dynamic hazards and useful for post-silicon debug.

As a second application, REBEL is used for hardware Trojan detection. REBEL can measure the subtle delay anomalies introduced by hardware Trojans. Trojan emulation circuits are used to model the delay anomalies introduced by hardware Trojans. The path delays measured in 62 copies of the chip fabricated in IBM's 90nm CMOS technology are used to present REBEL's ability to detect hardware Trojans. Linear regression analysis is applied to the data sets and is shown to reduce chip-to-chip process variation effects and significantly improve detection sensitivity. The results show that REBEL can detect delay anomalies introduced by subtle changes in the capacitive loads introduced by HTs.

Appendix A

Code Statistics

A large part of my time during this work was invested on automating the chip design process and data processing. I used several programming languages (including Perl, C, Tcl, Skill and Matlab) to achieve this automation. The table below lists the statistics of codes.

Module	Language	Files	Lines
REBEL-Integrated chip design automation	Perl	5	1816
	Tcl	2	523
	Skill	2	107
ASIC Data Processing	C	1	2009
	Perl	4	892
	Matlab	4	402
FPGA Data Processing	Perl	3	1132
	Matlab	3	207

References

[1] R. Raina, "What is DFM & DFY and Why Should I Care?", *Proc. ITC*, pp. 1-9, 2006.

[2] D. Burek, "True Design-for Manufacturability Critical to 65-nm Design Success", http://www.eetimes.com/showArticle.jhtml?arti-cleID=202803596, 2008.

[3] S. R. Nassif, "Modeling and Analysis of Manufacturing Variations", *Conference on Custom Integrated Circuits*, pp. 223-228, 2001.

[4] I. Ahsan et al, "RTA-Driven Intra-Die Variations in Stage Delay and Parametric Sensitivities for 65 nm Technology", *Proc. Symposium on VLSI Technology*, pp. 170-171, 2006.

[5] D.G. Chesebro et al., "Overview of Gate Linewidth Control in the Manufacture of CMOS Logic Chips", *IBM J. of Res. and Dev.*, *Vol.39*, pp. 189-200, 1995.

[6] J.-Y. Lai, N. Saka, J.-H. Chun, " Evolution of Copper-Oxide Dama-scene Structures in Chemical Mechanical Polishing", *J. of Electro-chem. Soc.*, pp. G31-G40, 2002.

[7] C. Hedlund, H. Blom, S. Berg, " Micro loading Effect in Reactive Ion Etching", *J. of Vacuum Science and Tech.*, pp. 1962-65, 1994.

[8] S. Paul, S. Krishnamurthy, H. Mahmoodi, S. Bhunia, "Low-over-head Design Technique for Calibration of Maximum Frequency at Multiple Operating Points", *Proc. ICCAD*, pp. 401-404, 2007.

[9] W. Xiaoxiao, M. Tehranipoor, R. Datta, "Path-RO: A Novel On-Chip Critical Path Delay Measurement under Process Variations", *Proc. ICCAD*, pp. 640-646, 2008.

[10] J. Aarestad, C. Lamech, J. Plusquellic, D. Acharyya and K. Agarwal, "Characterizing Within-Die and Die-to-Die Delay Variations Introduced by Process

Variations and SOI History Effect", *Proc. DAC*, pp. 534 - 539, 2011.

[11] D. Acharyya, K. Agarwal, J. Plusquellic, "Leveraging Existing Power Control Circuits and Power Delivery Architecture for Variability Measurement", *Proc. ITC*, pp. 1-9, 2010.

[12] Y. Jin; Y. Makris, "Hardware Trojan Detection using Path Delay Fingerprint", *Proc. International Workshop on Hardware-Oriented Security and Trust*, pp. 50-57, 2008.

[13] Haihua Yan; Singh, A.D., "On the effectiveness of detecting small delay defects in the slack interval", *Proc. IEEE International Workshop on Current and Defect Based Testing*, pp. 49-53, 2004.

[14] A.D.Singh, "A Self-Timed Structural Test Methodology for Timing Anomalies due to Defects and Process Variations", *Proc. International Test Conference*, pp. 90-96, 2005.

[15] A. Raychowdhury, S. Ghosh, and K. Roy, "A Novel On-chip Delay Measurement Hardware for Efficient Speed-Binning", *Proceedings of the 11th IEEE International On-Line Testing Symposium*, pp. 287-292, 2005.

[16] Pei, S.; Li, H.; Li, X.;, " A High-Precision On-Chip Path Delay Measurement Architecture", *Transaction on Very Large Scale Integration (VLSI) Systems*, pp. 1-13, 2011.

[17] Katoh, K.; Namba, K.; Ito, H.;, "A Low Area On-Chip Delay Measurement System Using Embedded Delay Measurement Circuit", *Proc. IEEE Asian Test Symposium*, pp. 343-348, 2010.

[18] K. Katsuki, M. Kotani, K. Kobayashi, and H. Onodera, "Measurement Results of Within-Die Variations on a 90nm LUT Array for Speed and Yield Enhancement of Reconfigurable Devices", *Proc. Asia South Pacific Design Automation Conference*, pp. 110-111, 2006.

[19] Eun Jung Jang; Gattiker, A.; Nassif, S.; Abraham, J.A.;, "Efficient and Product-Representative Timing Model Validation", *Proc. VLSI Test Symposium (VTS)*, pp. 90-95, 2011.

[20] P. Dudek, S. Szczepanksi, J.V. Hatfield, " A High-Resolution CMOS Time-to-Digital Converter utilizing a Vernier Delay Line", *IEEE Trans. Solid-State Circuits*, pp. 240-247, 2000.

[21] C.C. Chen, P. Chen, C.S. Hwang, W. Chang, " A Precise Cyclic CMOS Time-to-Digital Converter with Low Thermal Sensitivity", *IEEE Trans. Nucl. Sci*, pp. 834-838, 2005.

[22] J. Kalisz, "Review of Methods for Time Interval Measurements with Picosecond Resolution", *Metrologia, 41 (2004)*, pp. 17-32, 2004.

[23] H. Onodera, H. Terada, "Characterization of WID Delay Variability using RO-array Test Structures", *Proc. International Conference on ASIC*, pp. 658-661, 2009.

[24] N. Drego, A. Chandrakasan, D. Boning, " All-Digital Circuits for Measurement of Spatial Variation in Digital Circuits", *Solid-State Circuits, Vol. 45, No. 3*, pp. 640-651, 2010.

[25] M. Bhushan, A. Gattiker, M. Ketchen and K. Das, "Ring Oscillators for CMOS Process Tuning and Variability Control", *Trans. on Semiconductor Manufacturing*, pp. 10-17, 2006.

[26] B.P. Das, B. Amrutur, H.S. Jamadagni, N.V. Arvind, V. Visvanathan, "Within-Die Gate Delay Variability Measurement using Re-configurable Ring Oscillator", *Proc. Custom Integrated Circuits Conference*, pp. 133-136, 2008.

[27] D.J. Kinniment, O.V. Maevsky, A. Bystrov, G. Russell,A.V.Yakolev, "On-Chip Structures for Timing and Measurement", *Proc. ASYNC'02*, pp. 190-197, 2002.

[28] Z. Xin, K. Ishida, M. Takamiya, T. Sakurai, "An On-Chip Characterizing System for Within-Die Delay Variation Measurement of Individual Standard Cells in 65-nm CMOS", *Proc. DAC*, pp. 109-110, 2011.

[29] A. Mantyniemi, T. Rahkonen, J. Kostamovaara, " A CMOS Time-to-Digital Converter (TDC) Based On a Cyclic Time Domain Successive Approximation Interpolation Method", *Solid-State Circuits*, pp. 658-661, 2009.

[30] R. M. Rad, X. Wang, M. Tehranipoor, J. Plusquellic, "Power Supply Signal Calibration Techniques for Improving Detection Resolution to Hardware Trojans", *Proc. International Conference on Computer-Aided Design*, pp. 632-639, 2008.

[31] H. Salmani, M. Tehranipoor, J. Plusquellic, "New Design Strategy for Improving Hardware Trojan Detection and Reducing Trojan Activation Time", *IEEE International Workshop on Hardware-Oriented Security and Trust*, pp. 66-73, 2009.

[32] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar, "Trojan Detection using IC Fingerprinting", *Symposium on Security and Privacy*, pp. 296-310, 1996.

[33] Rad, R.; Plusquellic, J.; Tehranipoor, M., "Sensitivity analysis to hardware Trojans using power supply transient signals", *IEEE International Workshop on Hardware-Oriented Security and Trust*, pp. 3-7, 2008.

[34] J. Aarestad, D. Acharyya, R. M. Rad and J. Plusquellic, " Detecting Trojans Though Leakage Current Analysis Using Multiple Supply Pad IDDQ", *Transactions on Information Forensics and Security*, pp. 893-904, 2010.

[35] F. Wolff, C. Papachristou, S. Bhunia, and R. Chakraborty, "Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme", *Design, Automation andTest in Europe*, pp. 1362-1365, 2008.

[36] Potkonjak, M.; Nahapetian, A.; Nelson, M.; Massey, T., "Hardware Trojan horse detection using gate-level characterization", *DAC '09*, pp. 688-693, 2009.

[37] L. Jie, J. Lach, "At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection", *Proc. Workshop on Hardware-Oriented Security and Trust*, pp. 8-14, 2008.

[38] Xuehui Zhang; Tehranipoor, M., "RON: An on-chip ring oscillator network for hardware Trojan detection", *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1-6, 2011.

[39] S. Narasimhan, D. Du, R. S. Chakraborty, S. Paul, F. Wolff, C. Papachristou, K. Roy, S. Bhunia, "Multiple-Parameter Side-Channel Analysis: A Non-Invasive Hardware

Trojan Detection Approach", *Proc. International Symposium on Hardware-Oriented Security and Trust*, pp. 13-18, 2010.

[40] D. Du, S. Narasimhan, R. S. Chakraborthy, S. Bhunia, "Self-Referencing: A Scalable Side-Channel Approach for Hardware Trojans Detection", *CHES*, 2010.

[41] P. Sedcole and P. Y. K. Cheung, "Within-Die Delay Variability in 90nm FPGAs and Beyond", *Proc. International Conference on Field Programmable Technology*, pp. 97-104, 2006.

[42] B. Hargreaves, H. Hult, and S. Reda, "Within-Die Process Variations: How Accurately Can They Be Statistically Modeled?", *Proc. Asia and South Pacific Design Automation Conference*, pp. 524-530, 2008.

[43] A. Maiti, P. Schaumont, "Improving the Quality of a Physical Unclonable Function using Configurable Ring Oscillators", , pp. 703-707, 2009.

[44] Y. Jin, N. Kupp, and Y. Makris, "Experiences in Hardware Trojan Design and Implementation", *Proc. International Workshop on Hardware-Oriented Security and Trust*, pp. 50-57, 2009.

[45] Laung-Terng Wang, Cheng-Wen Wu, Xiaoqing Wen, " VLSI Test Principles and Architectures: Design for Testability", *The Morgan Kaufmann Series in Systems on Silicon*, 2008.

[46] Cadence Digital Implementation Tool Suite, http://www.cadence.com/products/di/Pages/default.aspx

[47] Open Cores, http://opencores.org

[48] C. Lamech, J. Aarestad, J. Plusquellic, R. Rad, K. Agarwal, "REBEL and TDC: Two Embedded Test Structures for On-Chip Measurements ofWithin-Die Path Delay Variations", *ICCAD*, 2011.

[49] H. Salmani, M. Tehranipoor, J. Plusquellic, "A Layout-aware Approach for Improving Localized Switching to Detect Hardware Trojans in Integrated Circuits", *International Workshop on Information Forensics and Security*, 2010.

[50] D. Rai, J. Lach, "Performance of Delay-Based Trojan Detection Techniques under Parameter Variations", *Proc. International Symposium on Hardware-Oriented Security and Trust*, pp. 58-65, 2009.

[51] John Neter, Michael H. Kutner, Christopher J. Nachtsheim, William Wasserman, " Applied Linear Statistical Models", *IRWIN, USA.*, 1996.