Security and Testability Issues in Modern VLSI Chips

Submitted in partial fulfilment of the requirments

of the Degree of

Doctor of Philosophy

by

Satyadev Ahlawat

(Roll No. 144076004)

Supervisor:

Prof. Virendra Singh

Department of Electrical Engineering

Indian Institute of Technology Bombay

Mumbai – 400076

September 2018

Publications

Included in Thesis

Journal publications

1. **Satyadev Ahlawat**, Jaynarayan Tudu, Anzhela Matrosova, and Virendra Singh, A High Performance Scan Flip-Flop Design for Serial and Mixed Mode Scan Test, *IEEE Transactions on Device and Materials Reliability (TDMR), 2018*, Volume: 18, Issue: 2, pp. 1 - 11, http://doi.org/10.1109/TDMR.2018.2835414

2. Jaynarayan Tudu, **Satyadev Ahlawat**, and Virendra Singh, Architectural Framework for Configurable Joint-scan DFT Architecture, *Journal of Electronic Testing: Theory and Applications (JETTA), 2018* [under review]

Peer reviewed conferences

3. **Satyadev Ahlawat**, Darshit Vaghani, Naveen Bazard, and Virendra Singh, "Using MISR as Countermeasure Against Scan-based Side Channel Attacks", *Proceedings in 16th IEEE East-West Design and Test Symposium (EWDTS) 2018*, Kazan, Russia, September 14 - 17, 2018.

4. Darshit Vaghani, **Satyadev Ahlawat**, Jaynarayan Tudu, Masahiro Fujita, and Virendra Singh, "On Securing Scan Design Through Test Vector Encryption", *Proceedings in 51st IEEE International Symposium on Circuits and Systems (ISCAS) 2018*, Florence, Italy, May 27 - 30, 2018, pp. 1 - 5, https://doi.org/10.1109/ISCAS.2018.8351212

5. **Satyadev Ahlawat**, Darshit Vaghani, Jaynarayan Tudu, and Virendra Singh, "On Securing Scan Design from Scan-Based Side-Channel Attacks", *Proceedings in 26th IEEE Asian Test Symposium (ATS) 2017*, Taipei, Taiwan, November 27 - 30, 2017, pp. 58 - 63, https://doi.org/10.1109/ATS.2017.23

6. **Satyadev Ahlawat**, Darshit Vaghani, and Virendra Singh, "Preventing scan-based side-channel attacks through key masking", *Proceedings in 30th IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT) 2017*, Cambridge, UK, October 23 - 25, 2017, pp. 1 - 4, https://doi.org/10.1109/DFT.2017.8244434

7. **Satyadev Ahlawat**, Darshit Vaghani, Jaynarayan Tudu, and Ashok Suhag, "A Cost Effective Technique for Diagnosis of Scan Chain Fault", *Proceedings in 21st International Symposium on VLSI Design and Test (VDAT) 2017*, Roorkee, India, June 29th - July 2nd, Communications in Computer and Information Science (CCIS), volume 711, Springer, Singapore, pp. 191 - 204, https://doi.org/10.1007/978-981-10-7470-7_20

8. **Satyadev Ahlawat**, Darshit Vaghani, Rohini Gulve, and Virendra Singh, "A low cost technique for scan chain diagnosis", *Proceedings in 50th IEEE International Symposium on Circuits and Systems (ISCAS) 2017*, Baltimore, MD, USA, May 28 - 31, 2017, pp. 1 - 4, https://doi.org/10.1109/ISCAS.2017.8050440

9. **Satyadev Ahlawat**, Darshit Vaghani, and Virendra Singh, "An efficient test technique to prevent scan-based side-channel attacks", *Proceedings in 22nd IEEE European Test Symposium (ETS), 2016*, Limassol, Cyprus, May 22 - 26, 2017, pp. 1 - 2, https://doi.org/10.1109/ETS.2017.7968241.

10. **Satyadev Ahlawat** and Jaynarayan T. Tudu, "On minimization of test power through modified scan flip-flop", *Proceedings in 20th International Symposium on VLSI Design and Test (VDAT), 2016*, Guwahati, India, May 24 - 27, 2016, pp. 1 - 6, https://doi.org/10.1109/ISVDAT.2016.8064878

11. **Satyadev Ahlawat**, Darshit Vaghani, Rohini Gulve, and Virendra Singh, "Enabling LOS delay test with slow scan enable", *Proceedings in 14th IEEE East-West Design & Test Symposium (EWDTS), 2016*, Yerevan, Armenia, October 14 - 17, 2016, pp. 1 - 4, https://doi.org/10.1109/EWDTS.2016.7807648

12. **Satyadev Ahlawat**, Jaynarayan Tudu, Anzhela Matrosova, and Virendra Singh, "A High Performance Scan Flip-Flop Design for Serial and Mixed Mode Scan Test", *Proceedings in 22nd IEEE International Symposium on On-Line Testing and Robust System Design (IOLTS) 2016*, Catalunya, Spain, July 04 - 06, 2016, pp. 233 - 238, https://doi.org/10.1109/IOLTS.2016.7604709

13. **Satyadev Ahlawat**, Jaynarayan Tudu, Anzhela Matrosova, and Virendra Singh, "A New Scan Flip Flop Design to Eliminate Performance Penalty of Scan", *Proceedings in 24th IEEE Asian Test Symposium (ATS) 2015*, Mumbai, India, November 22 - 25, 2015, pp. 25 - 30, https://doi.org/10.1109/ATS.2015.12

Other Contributions

14. Nihar Hage, **Satyadev Ahlawat**, and Virendra Singh, "In-situ Monitoring for Slack Time Violation Without Performance Penalty", *Proceedings in 51st IEEE International Symposium on Circuits and Systems (ISCAS) 2018*, Florence, Italy, May 27 - 30, 2018, pp. 1 - 5, https://doi.org/10.1109/ISCAS.2018.8351000

15. Jaynarayan T. Tudu and **Satyadev Ahlawat**, "Guided shifting of test pattern to minimize test time in serial scan", *Proceedings in 20th International Symposium on VLSI Design and Test (VDAT), 2016*, Guwahati, India, May 24 - 27, 2016, pp. 1 - 6, https://doi.org/10.1109/ISVDAT.2016.8064851

Abstract

The advancements in semiconductor fabrication process has made it possible to built electronic systems with unimaginable functional complexity. This has led to the development of highly complex systems like autonomous vehicles, personnel healthcare, deep neural networks, virtual reality, smart homes and cities, and many more emerging *IoT* (Internet-of-Things) based applications. Many of these new applications are very critical from reliability and security point of view.

The reliability requirement of such systems has made the testing of these systems very challenging. In applications like automobiles or autonomous vehicles, manned space mission, defence etc., where human life is at stake, the system reliability is of utmost importance. For such systems the *DPPM* (defective parts per million) requirement is ideally zero. To ensure shipment of such highly reliable systems the test requirements are becoming very stringent. Moreover, these modern day *IoT* based systems not only have complex functionality they also do continuous communication with other systems in their surrounding. This has put system security along with reliability in forefront of the design challenges in designing such systems.

The problem further escalates because security and testability have orthogonal objectives. To achieve high testability, the internal state information of a chip must be as visible as possible to the test engineer. On the other hand, from security point of view the sensitive information embedded in a chip must not be accessible to the outside world. This thesis investigates the conflicting requirements of security and testability in modern day complex *VLSI* chips. We introduce techniques which can test these security sensitive chips in a secure manner.

The scan-based *DfT* (Design-for-Test) architecture is the only economically viable test technique available today which can effectively test the modern day highly complex chips, and fulfill the stringent quality requirements. Because of the testability and diagnostic requirements scan design has become the de-facto *DfT* technique and is therefore employed in almost every chip. The scan architecture is generally operated by a test engineer to perform scan test operation, however, a malicious user can exploit the scan architecture to observe the sensitive data stored on-chip in a security or cryptographic chip.

The thesis first addresses security issues in scan test. We have proposed a set of techniques which can effectively secures the scan design against all the known scan-based side-channel attacks. The proposed techniques are based on test protocol countermeasures such as encryption key masking, test restriction, and test data encryption. The proposed techniques not only thwart scan attacks but also preserves the test capability of scan architecture. Moreover, the proposed secure scan design techniques are very efficient in terms of design cost.

Next, this thesis also addresses the testability issues like test data volume, test time, and test power which scan inherits because of its serial nature. With ever increasing circuit complexities these parameters are growing exponentially and needs attention as the test cost constitute almost half of the total chip cost. In this thesis, we propose a composite scan architecture which aims to combine both, the serial scan and random access scan, to harness the best out of each. The proposed architecture minimizes test time, test data volume, and test power all together. Further, we carry out a foundational study for its feasibility and to compare its advantage over the existing multiple serial scan architecture and random access scan architecture.

Finally, the thesis delve into some other scan issues like scan performance overhead, unnecessary combinational switching power during scan shift, scan chain diagnosis, and launch-off-shift based delay test with slow scan enable signal. To tackle these issues, we have proposed solutions based on efficient circuit level design of scan cell.

Keywords

VLSI Testing, Design-for-Test, Serial Scan, Random Access Scan, Multiple Serial Scan, Joint-scan, Test Power, Test Data Volume, Power Aware Test, Scan Attack, Secure Scan design, Scan Cell, and Scan Chain Diagnosis

Notation and Abbreviations

C_L	:	Load Capacitance
I_{DDQ}	:	Quiescent Current
L_p	:	Launch Pulse
VDD	:	Drain Supply Voltage
2M-JScan	:	2-Mode Joint-scan
4M-JScan	:	4-Mode Joint-scan
AES	:	Advanced Encryption Standard
ATE	:	Automatic Test Equipment
ATPG	:	Automatic Test Pattern Generation
BIST	:	Built-In Self-Test
C-X	:	Care - Don't care bit pair
CBR	:	Care Bit Ratio
CMOS	:	Complementary MOS
CP	:	Clock Signal/Pulse
CUT	:	Circuit Under Test
DC	:	Direct Current
DfM	:	Design-for-Manufacturability
DfT	:	Design-for-Test
DSM	:	Deep Sub-micron
IC	:	Integrated Circuit
ILP	:	Integer Linear Programming
JScan	:	Joint-scan
LBIST	:	Logic Built in Self Test
LFSR	:	Linear Feedback Shift Register
LOC	:	Launch off Capture
LOS	:	Launch off Shift
LSSD	:	Level Sensitive Scan Design
MBIST	:	Memory Built in Self Test
MISR	:	Multiple Input Signature Register
MOS	:	Metal Oxide Seminconductor
MOSFET	:	MOS Field Effect Transistor
MSDFF	:	Multiplexed Scan D Flip-flop
MSS	:	Multiple serial scan
MT-Fill	:	Minimum Transition Fill
NMOS	:	N-type MOS
P-random	:	Partial Random Access Scan
P-serial	:	Partial Serial Scan
PCB	:	Printed Circuit Board
PDF	:	Path Delay Fault
PI	:	Primary Input
PMOS	:	P-type MOS
PO	:	Primary Output
PPI	:	Pseudo Primary Input
PPO	:	Pseudo Primary Output
PRAS	:	Progressive Random Access Scan
PRAS-FF	:	Progressive Random Access Scan Flip-Flop
RAM	:	Random Access Memory
RAS	:	Random Access Scan
SAF	:	Stuck-at-Fault
SDF	:	Segment Delay Fault
SE/SEn	:	Scan Enable
SF$_i$	:	i^{th} Scan Flip-Flop
SI	:	Scan Input
SO	:	Scan Output
SoC	:	System On Chip
TCL	:	Test Control Logic
TCl/TCLK	:	Test Clock
TDF	:	Transition Delay Fault
TE	:	Test Enable
VLSI	:	Very Large Scale Integration
X-C	:	Don't care - care bit pair
X-Fill	:	Don't Care Fill

Contents

List of Tables

List of Figures

Chapter 1

Introduction to VLSI Test

The semiconductor fabrication process has made tremendous advancements in recent years. With ever decreasing feature size it has become possible to integrate billions of transistors on a single chip with transistor densities as high as 100 million transistors per square millimeter (MTr/mm^2) of chip area [100, 204]. The fabrication process for these present day integrated circuits (IC) has become extremely complex with more than 30 layers and over 1000 process steps [102]. At the heart of the IC fabrication lies the lithography process, which prints the patterns on the silicon wafer. Today's state-of-the-art fabrication lines are based on $193nm$ *immersion lithography*. Since $193nm$ light can not directly define a feature which is smaller than it's own size, hence complex optical techniques along with multiple patterning process are used to define features which the modern day chips require. This has significantly increased the total number of lithography steps required to fabricate a chip. It takes almost 80 lithography steps to fabricate today's most complex of chips [205]. Furthermore, the alignment of all the lithography steps is very crucial to correctly define the features. Also the design for manufacturability (DfM) rules has increased dramatically because of the printing challenges in currently used lithography technique. The explosion in design rules is mainly because of the huge difference between $193nm$ wavelength and the feature sizes needed to fabricate the modern day chips [39]. With ever decreasing feature sizes the design rules are increasing and becoming more restrictive in what are manufacturable layout patterns. All these factors have led to a very little tolerance for variations in layout patterns. Also, with very small feature sizes the devices have become more vulnerable to process variations. A small perturbation in process parameters may cause significant variations in device parameters across dies and wafers. As a result, the probability of occurrence of manufacturing defects during fabrication process has increased.

A small defect may lead to a faulty transistor or faulty interconnect wire. The manufacturing defects can impact the chip in two ways: either the chip will fail to function at all or it will fail to function at target frequency. The former lead to yield loss while the latter causes decrease in revenue. Since there is always probability of a fabricated chip being faulty, testing of all the manufactured chips is necessary in order to segregate the faulty chips from the good ones. The faulty chips must be screened out before they are assembled into printed circuit boards, which in turn must be tested before integration into systems. This is because of the commonly agreed upon *rule of ten* which states that the cost of finding the faulty chip increases ten times as we move from one level of integration to another i.e., chip to PCB to system and finally to field operation [36, 59, 219].

1.1 VLSI Testing

So, how do we test a chip? We apply a sequence of input patterns at the primary inputs of the chip and record the corresponding output responses at the primary outputs. These recorded output responses are compared with the predetermined responses also called golden responses. The golden responses are generated through circuit simulation. The input pattern which produces an erroneous output response in case there is fault present in the circuit is called a test pattern and the corresponding response a test response. The test patterns and test responses together are called test data. If the recorded test responses match with the golden responses (simulated correct responses) that means the fabricated circuit or chip is correct else the chip is flagged as faulty [36, 73]. The basic test principle is depicted in Figure 1.1.

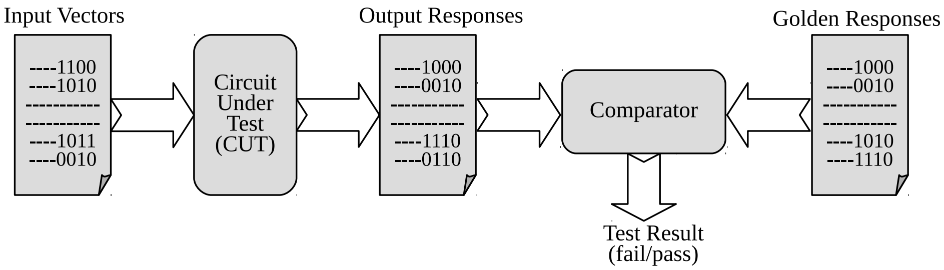


Figure 1.1: Basic VLSI test principle

The testing of logic circuits consists two steps: 1.) generation of test patterns for the circuit under test (CUT), and 2.) application of test patterns to the CUT. The test development and application time decide the test quality and test cost which are interdependent. The test quality depends upon the thoroughness of the test, however, a large test set increases the test development and application time and hence test cost. Furthermore, a large test set also increases the time-to-market which may severely impact the financial success of the product in the market [202]. A good test is supposed to satisfy the following three requirements [36]:

(i) It should detect all the possible defects

(ii) Test development time must be economical

(iii) Test application time must be economical

The first requirement ensures the quality of shipped products. The second and third requirement come from the test cost. Presently, the test cost constitute a large part of the overall product cost [39, 66, 97, 105, 166, 247]. The test cost for present day systems can go as high as 40% of the total product cost [219]. Hence, in order to produce quality products at economical price, a test set should fulfill all the above three requirements.

1.1.1 Functional vs Structural Testing

Historically, the testing of chips was done using functional test vectors. The functional testing is an extension of design verification which is used to verify the correctness of the design [202]. The functional test uses verification vectors which are based on logical relationship between the inputs and outputs of a circuit. The quality of functional tests depends upon the thoroughness of the test vectors. Since, the functional test patterns are generated for verifying a specific functional scenario, hence may not detect all possible defects. However, the functional test guarantees detection of all possible defects if done exhaustively, i.e., all possible combinations of functional input vectors are exercised. The number of functional test vectors required to test a chip exhaustively depends upon the number of primary inputs and increases exponentially. The number of functional test vectors are given by 2^N, where N is the number of primary inputs. Most of the practical *ICs* have more than hundreds of functional inputs which makes it impossible to test them using functional testing.

As the logic circuit size and functionality evolved over the years, the cost of functional test development and application became unviable [66, 202]. The functional test technique has been largely replaced by structural test technique in manufacturing test. A small set of functional test patterns are still used as top up patterns to structural patterns in structural testing. The structural testing, first proposed by Richard D. Eldred [70], is based on the specific structure of the circuit, i.e., type of gates, interconnects, and netlist. Unlike functional testing, the structural test pattern count increases linearly with circuit primary inputs [73]. One of the greatest advantage of structural testing over functional testing is that it allows to develop *automatic test pattern generation* (*ATPG*) algorithms [36]. These *ATPG* algorithms are based on *fault models*. A fault model is an abstract representation of physical defect at functional level. Since the real manufacturing defects can be so many, it is almost impossible to generate test vectors to detect all the physical defects. The fault models accurately reflects the logic level behavior of the defect, and also allows to generate the test vectors and fault simulate them in a computationally efficient way [219].

1.1.2 Fault Models

Many fault models have been proposed over the years [1]. The behavior of all types of defects can not be modeled accurately by a single fault model and hence a combination of fault models is used for test vector generation and fault simulation [202]. The commonly used fault models across industry and academia are *stuck-at fault* [75] model, *delay fault* model, *transistor fault* [38, 216] model, and *bridging fault* [137] model.

Stuck-at fault model

The stuck-at fault model is the most fundamental to structural testing of logic or digital circuit. The stuck-at fault model uses gate level fault modeling in which the faults are modeled only on the interconnects (or net) connecting the logic gates and the logic gates themselves are considered fault free. The interconnect can be subjected to two types of stuck-at faults: *stuck-at-1* (*s-a-1* or *sa1*), and *stuck-at-0* (*s-a-0* or *sa0*). While the *s-a-1* fault permanently sets the faulty net to 1 (logic high), the *s-a-0* fault permanently sets the faulty net to 0 (logic low).

Since a circuit can have multiple faults at a time, the total number of faults or stuck net combinations in a circuit with n lines or nets can be $3^n - 1$. This model is called multiple *stuck-at* fault model and is not used because even a moderate value on n will result in a large number of multiple *stuck-at* faults [36]. As a result, the common practice is to use single *stuck-at* fault model which is characterized by the three properties given below:

(i) Only one net is faulty in the circuit

(ii) The faulty line either can have *s-a-1* or *s-a-0* fault

(iii) The fault can be on an input or output of a logic gate.

Since the single *stuck-at* fault model restricts the number of faults that can exist in a circuit to 1, an n line circuit at most can have $2n$ faults. This number is further reduced considerably by fault collapsing process which is based on fault equivalence [1].

On an average the fault collapsing reduces the total number of faults by 50% to 60% [36]. The single *stuck-at* fault model is simple and computationally efficient for test pattern generation and fault simulation. Furthermore, the total number of modeled faults in this fault model increase linearly with number of circuit nets, which in case of functional testing increase exponentially with circuit size. All these advantages lead to the adoption of structural testing in manufacturing test.

Delay fault model

The other fault model currently in practice is *delay fault* model. The defects which do not change the functionality of the circuit, however, degrade the performance of the circuit are modeled by *delay fault* model. The delay faults increases the combinational delay of the circuit and makes it to exceed the clock period [113]. The delay faults are becoming more prevalent with decreasing feature size. The delay faults can not be detected by *stuck-at* fault test. Hence, a separate type of test called *delay test* is used to detect delay defects. The delay fault detection requires creation of a transition at the fault site and then propagation of that transition to a observable circuit node which could be a primary output (PO) or a clocked flip-flop. The creation of a transition requires a vector pair, $\langle V_1, V_2 \rangle$, at the primary inputs (PI) of the combinational circuit [219]. The most commonly practiced *delay fault* models are *transition delay fault* (TDF) model [48, 128, 218], *path delay fault* (PDF) model [192], and *segment delay fault* model [92, 93]. In most of the *delay fault* models, the gate input to output delay and interconnect delay are combined together and are represented by *gate delay.*

- *Transition Delay Fault* (TDF) model assumes that only one gate in the circuit has delay fault. Further, it is assumed that the delay of the fault site is large enough to cause the propagation delay of signal transition passing through the fault site or line exceed the clock period. The TDF model lump the fault at a single gate or line. Hence propagation delay of every path, irrespective of whether it is a short path or long path, which passes through the faulty line or gate exceeds the clock period of circuit. Therefore, the TDF model is also called a gross-delay defect model. Each gate in the circuit can be subjected to two types of transition delay faults : *slow-to-rise* fault and *slow-to-fall* fault. The number of possible transition delay faults in a circuit is $2N$, where N is the total number of gates or lines. The number of *TDF's* are relatively small and increases linearly with number of gates [219]. The assumption of a single gate affected by delay defect and failing of all paths which pass through that gate is not very realistic. However, because of the practical simplicity of *TDF* model it is most frequently used *delay fault* model. Furthermore, the *ATPG* of *stuck-at* faults can be easily modified to generate *TDF* test patterns [113, 202].

– *Path Delay Fault* (*PDF*) model on the other hand assumes that the delay defect is distributed over the whole combinational path and hence also called distributed delay defect model. In *PDF* model it is assumed that the cumulative delay of the combinational path, which includes the logic gates and interconnects delays, is large enough that it exceeds the clock period. Similar to the *TDF* model each path in *PDF* model may subject to *slow-to-rise* and *slow-to-fall* faults corresponding to rising and falling transitions at the starting point of the path. The total possible number of *PDF's* can be $2N$, where N is the total number of combinational paths that exists in a circuit. As the number of paths in a circuit increases exponentially with the circuit size it is practically impossible to enumerate all paths in a practical circuit. Despite the fact that *PDF* model is more realistic than the *TDF* model, it is not exercised for all the circuit paths. The *PDF* model is used only for a small set of timing critical paths [130, 176, 246].

– *Segment delay fault* (*SDF*) model is another fault model which offers a trade-off between *TDF* model and *PDF* model [113]. The basic assumption in this model is that the delay defect is distributed over a segment of a path or affects several gates in a local region which forms a segment of a path. Further, it assumes that all the paths passing through the affected segment will have delay fault. The length L of the segment can be anywhere between L_{max} and 1, where L_{max} represents the number of gates in the longest path, and $L = 1$ represents a single gate. When $L = L_{max}$, the *SDF* model becomes equivalent to *PDF* model. In case, when $L = 1$, the *SDF* model becomes equivalent to *TDF* model. The length of path segment can be decided based on the available manufacturing defect statistics.

Transistor fault models

Unlike the *stuck-at* and *delay fault* models which are gate level fault models the *transistor fault* model is a switch level fault model [219]. In this fault model it is assumed that a *MOSFET* transistor can be subjected to two faults: *stuck-open* and *stuck-short* also referred as *stuck-off* and *stuck-on*. In a *stuck-open* fault, the *MOS* transistor always remains in *open* state and in *stuck-short* fault, the *MOS* transistor always remains in *shorted* state. Further, it is assumed that only one transistor is faulty out of the total transistors which realize the *CMOS* logic gate [36].

Transistor faults in *CMOS* logic circuits can not be detected by stuck-at fault test. The output of a logic gate containing the faulty transistor with stuck-open fault remains in a floating state. To detect a transistor *stuck-open* fault, a sequence of two vectors is applied. The first vector initializes the affected output node and the second vector which is a *stuck-at* vector detects the presence of the fault. If the two vectors sequence produces a hazard at the output of the faulty gate then fault detection can not be guaranteed [106, 163].

The transistor *stuck-short* fault on the other hand produces a conducting path between power supply and ground supply rails. The fault is activated by a specific input combination at the gate inputs. Once activated, the fault can affect the output logic level of the gate containing the faulty transistor. The output logic level of the affected gate depends upon the impedance of the shorted transistor. The logic level of the output node of the affected gate may or may not be interpreted as an invalid state by the inputs of the gates driven by the faulty gate. However, the transistor *stuck-short* fault is detected by quiescent current measurement called I_{DDQ} test. The transistor *stuck-open* and *stuck-short* fault modeling has been extensively studied by the test community [69, 124, 127, 149, 183, 184]. However, these fault models have not been assimilated into *VLSI* test flow. The *stuck-at* fault test detects the transistor *stuck-open* faults. However, the fault coverage of transistor *stuck-open* faults achieved by *stuck-at* test lags 10% to 15% behind the fault coverage of *stuck-at* faults [6].

Bridging fault model

The *bridging fault* model represents a short between two (or more) signal wires or nets. The bridging fault commonly happens between signal lines or interconnects which are physically very close to each other in the physical layout. The short can happen between transistor terminals or between interconnects of transistors and logic gates [219]. The bridging fault is modeled at the logic gate level or transistor level [202]. The bridging fault is similar to the real defects which happen in silicon and hence is also called defect-oriented fault [169, 175].

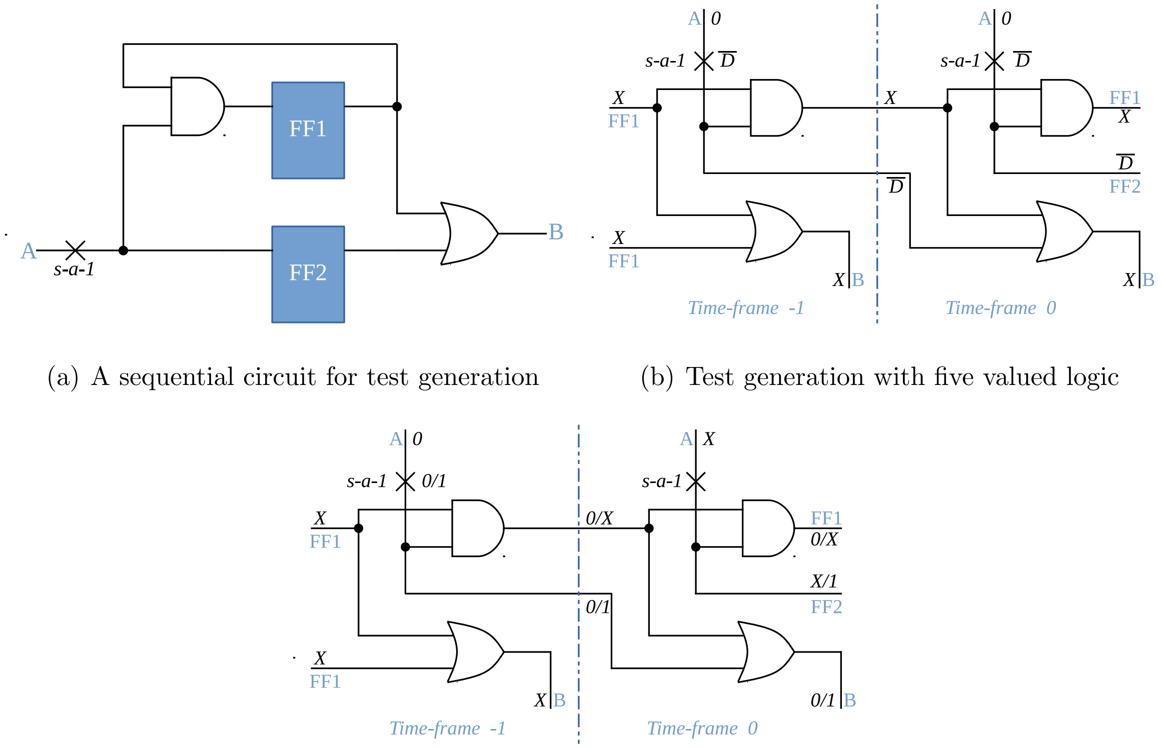
There are many *bridging fault* models available in literature [1, 66]. The very first bridging fault model called *wired-AND/wired-OR* model is a simple fault model and is independent of the bridge resistance and the location of the bridged nodes [137]. There are more complex models available which take into account the bridge resistance and location information [2, 135]. Another complication in modeling of bridging faults is formation of feedback paths which may cause oscillation. The *wired-AND/wired-OR* fault model was developed for *bipolar* devices and does not model the behavior of bridges in *CMOS* devices. The bridging defects in *CMOS* devices are modeled by *dominant bridging fault* model in which it is assumed that the shorted net is dominated by the powerful driver. It has been shown that a test set which detects all the dominant bridging faults is also guaranteed to detect all *wired-AND/wired-OR* faults [219]. A recent fault model called *dominant-AND/dominant-OR* accurately models the behavior of resistive shorts which were not modeled by dominant bridging fault model [198]. The bridging faults closely match with the real defects which commonly occurs in circuit fabrication.

The bridging faults can be detected by I_{DDQ} test. However, with continuously decreasing feature size the leakage current in *CMOS* devices is increasing and the change in I_{DDQ} current due to bridging fault may not be reliable detectable [219]. Also it has been shown that many bridging faults are detected by *N-detect* single *stuck-at* fault test vector set with very high fault coverage [77, 82, 155, 164].

1.1.3 Sequential Circuit Testing

Most of the digital system of significant size are sequential in nature. These sequential circuits have internal memory states which makes testing of sequential circuits very difficult. The sequential circuits have internal states which are unknown at the start of the test. To test the sequential circuit, first it must be brought into a known state. Then the fault is activated and after that the fault effect is propagated to the primary outputs. Thus a sequential test can have a long sequence of test vectors which must be applied in a specific order. There are mainly two techniques for test generation for single-clock synchronous sequential circuits: *time-frame expansion* [114, 157] and *simulation-based* [35, 182] method.

The time-frame expansion method uses a gate level model of the circuit and uses a combinational *ATPG* [81] for test generation. To use combinational *ATPG* for sequential test pattern generation, the sequential circuit needs to be unrolled into a larger combinational circuit. To unroll a sequential circuit, all the state elements are removed from the circuit and the outputs of the state elements are made into pseudo primary inputs (*PPI*). Similarly, the inputs of the state elements are made into pseudo primary outputs (*PPO*). Now this transformed form of circuit is replicated multiple times as per the test generation requirement. An example sequential circuit along with its time-frame expansion is shown in Figure 1.2. It can be seen from the expanded circuit shown in Figure 1.2(b) that the *PPI* of one frame is connected to *PPO* of the preceding frame. The *PPI* of the very first frame (i.e., left most frame) are initialized to *don't-care* or *X* values. This process of unrolling the sequential circuit into a combinational circuit is called *time-frame* expansion.



(a) A sequential circuit for test generation

(b) Test generation with five valued logic

(c) Test generation with nine valued logic

Figure 1.2: Sequential circuit test generation using time-frame expansion method [36]

Now to generate the test pattern the fault is copied in every time-frame. To use the combinational *ATPG* for the expanded circuit a nine-valued logic is required [36]. Because the five-valued logic which is used for combinational *ATPG* can not handle initialization fault of state elements, a situation depicted in Figure 1.2(b). To overcome this issue a nine-valued logic system is required which can handle the state initialization issue due to faults in the state elements [144]. It can be observed in Figure 1.2(c) that the nine-valued logic can successfully resolve the initialization problem and generate a valid test. To justify a state the *ATPG* may have to try $9^{N_{ff}}$ state vectors or time-frames in the worst case scenario before it declares a fault untestable. With a moderate size state vector N_{ff} (number of flip-flops) the total number of distinct state vectors become very large. Further, time-frame expansion method is very inefficient for sequential circuit with cyclic structures, multiple-clocks, and asynchronous circuitry [36]. Hence, sequential test generation based on time-frame expansion method is an intractable problem.

The *simulation-based* method uses a fault simulator and test pattern generation programme. Among simulation-based techniques the concurrent fault simulator (CFC) along with genetic algorithms can provide good efficiency [8, 136]. The existing sequential test pattern generation techniques can not handle the complexity of present day circuits. The use of sequential test generation techniques is practically impossible for present day circuits. The solution to this problem is use of combinational *ATPG* along with the Scan *DfT* architecture also called scan design. The main advantage of scan design is that it allows to test a sequential circuit like a combinational one.

1.2 Design for Testability

The Design for Testability (*DfT*) techniques are the design practices that are used to improve the testability of complex logic designs. There are two types of *DfT* approaches: *ad-hoc* and *structured*. The *ad-hoc* techniques are based on finding testability problems in circuit with manual inspection, and then inserting test points. Algorithmically generated testability measure are also used to find signals with weak controllability and observability and then required changes are made in the design. As the circuit complexity increases, manual inspection becomes difficult. Also the algorithmically generated testability measures are not accurate [5]. Furthermore, the *ATPG* can not provide a test set with adequate fault coverage for logic circuits with *ad-hoc DfT* structure.

The complex logic designs require a structured *DfT* approach to achieve acceptable fault coverage [68]. In structured *DfT* techniques extra circuitry is added to the regular circuit in a systematic way. Presently, scan design and built-in self-test (*BIST*) are the two commonly used structured *DfT* architectures [36]. The *BIST* technique is based on test signature analysis which is a statistical property of the circuit [229]. The correctness of the chip is determined by comparing the test response signature with the golden signature which is stored *on-chip*. The signature is generated by compacting the responses of the pseudo-random test sequences applied to the circuit [25]. The test sequence and the corresponding signature are generated on-chip. The test sequence is generated using an on-chip *LFSR* [65, 83] and the corresponding signature is generated using a *MISR* which is also implemented on-chip. The *BIST* for logic testing as well as for memory testing is available and are called *logic-BIST* (or *LBIST*) and *memory-BIST* (or *MBIST*) respectively. The *MBIST* is commonly practised for memory testing, however, the use of *LBIST* is very limited. The *LBIST* technique does not require an *ATPG* programme for test generation and an *ATE* for test application, which is a very costly equipment. However, the *LBIST* has low fault coverage and high area overhead as compared to scan design [36]. Further, the *BIST* technique suffers from poor diagnosis capability and signature aliasing [230, 231].

1.2.1 Scan Architecture and Scan Testing

Presently, scan-based *DfT* is the only practical test architecture available that can test a present day highly complex *VLSI* chip with a satisfactory test coverage. The concept of scan was already in use for system level test, however its application in structural testing was first proposed by Williams et al. [228]. It was first adopted by *IBM* and *NEC*, which used various implementations of scan design [68]. Later, as the circuit complexities increased, scan *DfT* architecture gained popularity because of its simplicity and effectiveness in testing complex designs.

In a scan-based *DfT* architecture also popularly known as scan design, the main idea is to get controllability and observability of every flip-flops in the *CUT*. This is achieved by replacing sequential element or flip-flop in a circuit by a scan cell. The scan design in which each and every flip-flop in the *CUT* is replaced by a scan cell is called full scan design. The main advantage of full scan design is that it allows to test a sequential circuit just like a combinational circuit. Moreover, a combinational *ATPG* can be used to generate the test vectors and corresponding responses.

The scan architecture along with the scan cell is depicted in Figure 1.3. The scan cell is a multiplexed input D-type flip-flop. In addition to regular data or functional input (D), the scan cell has one extra input called test or scan input (SI). The scan multiplexer is controlled by a global control signal called *scan enable* (SE) or *test enable*

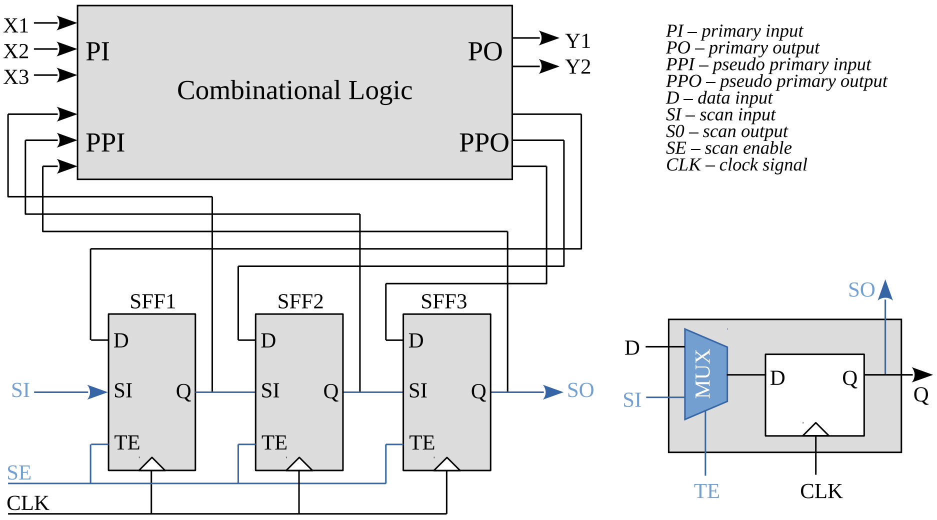


Figure 1.3: Serial scan *DfT* architecture and basic scan cell [36]

(TE). Using the test enable signal either the regular functional input D or the test input SI can be selected by the multiplexer. The scan design basically adds an extra mode of operation to the circuit functionality which is popularly called scan mode or test mode. During scan mode, all the scan cells are connected serially and form one or more serial shift register(s) popularly known as scan chain(s) among the test community. The scan chain has a scan-in (SI) and scan-out (SO) pin which are controlled through primary input/output (I/O) pins. By using the test enable signal, which is available at users disposal, the circuit can be switched between functional and scan mode anytime. During the scan mode, scan chains can be loaded and unloaded serially via the scan I/O pins. By using the scan feature, the circuit can be brought into any known state and the present state of the circuit can be observed.

The scan based testing is carried out in two steps: test generation and test application. While the test generation is carried out using an $ATPG$ program the test application is performed by an Automatic Test Equipment (ATE). During test application the following operations are performed: test vector shift-in through SI, test vector launch, test response capture (this is performed in functional mode), and test response shift-out through SO. The shift-in and shift-out operations are performed simultaneously. While the next test vector is being shifted-in the response from the previous test vector is shifted-out. For a scan chain of length l, where l is the number of scan cells in a scan chain, l shift cycles are required to load/unload the test vector/response. Hence the test time directly depends upon the scan chain length. To reduce the test time multiple parallel scan chains are formed with independent scan-in and scan-out port. The scan design with multiple scan chains is generally called *multiple serial scan* (*MSS*). The number of shift cycles required to load a test vector depend upon the length of the scan chain. The last shift-in cycle also launches the test vector. The test response is captured by switching the circuit in functional mode and applying a functional clock pulse. The captured response is serially observed at the scan-out pin by the *ATE* and compared with the golden response. Depending upon the comparison result the *CUT* is flagged as faulty or fault free.

Partial scan design

A variant of full scan design which is called partial scan design, includes only a subset of the total flip-flops in the *CUT* into the scan chain. It provides a trade-off between ease of testing and the cost associated with the full scan design [28, 200, 237]. The partial scan design performs better in terms of test time and test power dissipation, however, suffers from comparatively lower fault coverage. The increasing circuit complexity has forced the industry to abandon partial scan design, which necessitates a computationally demanding and unaffordable sequential *ATPG* (or combinational *ATPG* with time-frame expansion), and to rather adopt full scan despite its cost.

Random Access Scan Design

The full scan and partial scan both uses serial shift operation for loading/unloading of test vectors/responses. This causes a lot of switching activity in the scan cells, which also propagates in the combinational logic. Because of this a lot of unnecessary power dissipation occurs in the *CUT* during scan shift operation. As observed by some of the recent studies the test power dissipation could be significantly higher than the functional power and poses a big challenge for test engineers. The high test power dissipation restricts the internal scan chain shift frequency at nearly $10MHz$ to $50MHz$ only. An alternate to serial scan design is Random Access Scan (RAS) design which can effectively reduce the test power along with test time and test data volume. The RAS was first proposed by Ando in 1980 [17]. In RAS the test vector loading is performed in a random-access memory (RAM) like fashion. The write operation for care bits of a test vector care bit and read operation care bit of test response are performed bit-by-bit. Hence there is no serial shift operation and hence very less power dissipation. Also, 90 to 95% bits in a test vector are *don't-care* bits, there is a significant reduction in test data volume and test application time. Literature shows that RAS can greatly reduce test application time and test data volume along with a reduction in test power up to 99% [3, 20, 21, 23, 143]. It shows that RAS architecture has the potential towards solving the scan shift frequency issue. The main issue that RAS architecture faces in its practical implementation is the routing congestion.

1.2.2 Scan based Testing

The scan design is commonly used to exercise static (*stuck-at*) and timing (*delay*) test. The procedure to carry out *stuck-at* and *delay* test is briefly explained below.

Stuck-at fault test

The static test is used to detect the presence of stuck-at faults in the circuit. The test vectors are generated by a combinational $ATPG$ using single *stuck-at* fault model. The *stuck-at* test application is carried out using an ATE. The test application involves shift-in and launch of test vector, response capture, and shift-out of test response. First of all the scan enable (SE) is raised to logic high (1) level to switch the circuit from functional mode to test mode. Now by successive application of test clock the test vector is serially loaded into the scan chain through the scan-in (SI) pin. During the last shift cycle the test vector is also launched at the combinational inputs. After that the circuit is switched back to functional mode by making SE signal 0. Once the circuit is in functional mode, the test response is captured back into the scan cell by applying a clock pulse. Again, the circuit is switched back to test or shift mode to unload the captured response. The unloading of test response is done with simultaneous loading of next test vector.

Delay fault test

The delay test requires a test vector pair $\langle V_1, V_2 \rangle$. The first vector V_1 initializes the circuit and the second vector V_2 launches a transition. In scan based delay test the first vector V_1, also called initialization vector, is generated using a combinational *ATPG*. The second vector V_2, also called transition vector, can be generated in two ways. Depending upon how the V_2 vector is generated the delay test is called *Launch-off-Capture* (*LOC*) or *Launch-off-Shift* (*LOS*).

1. Launch-off-Shift: In LOS based delay test the V_2 vector is one bit shift over initialization vector V_1. The LOS test is also called *skewed-load* delay test [178]. The V_1 vector is launched in the last shift cycle. The response of V_1 is not captured in scan cells and the circuit is kept in test mode. To generate V_2 which is one bit shift over V_1 one clock pulse is applied. This extra shift clock pulse generates and launches the transition vector V_2. To detect the presence of transition delay fault the response of transition vector needs to be captured at-speed. Hence, the *SE* signal must get disabled before the arrival of at-speed functional clock [179]. Hence in order to apply *LOS* test, the scan enable (*SE*) signal must be timing closed [134]. The *SE* signal is also a global signal just like the functional clock signal. To make *SE* signal timing closed is a very costly task. In general, the *LOS* test is not exercised into industry due to its very high implementation cost.

2. Launch-off-Capture: The *LOC* test is also called *broad-side* delay test because the second vector of the test vector pair is provided in a broad-side fashion, namely through the logic [179]. In *LOC* test the V_2 vector is functional response of initialization vector V_1. The initialization vector V_1 is loaded and launched into similar manner as in *stuck-at* test. The combinational response of vector V_1 , which also acts as launch of vector V_2 , is captured keeping the *SE* signal low and then applying

a clock pulse. The response of V_2 is captured at-speed or at functional frequency. After the at-speed capture, SE is pulled up to logic high (1) again to shift-in the next test vector. Unlike LOS test the LOC test does not require timing closed SE signal [134].

1.2.3 Issues in Scan Test

Over the years, serial scan design has become the de-facto Design for Testability (DFT) technique. The ease of testing and high test coverage has made it gain widespread industrial acceptance. Furthermore, scan design also plays a critical role in diagnosis and post-silicon debug. Despite all the good things that scan offers, it has some inherent issues which need urgent attention. The issues that scan design is currently facing are:

1. Security Issue
   (a) Scan based side-channel attacks
2. Testability Issues
   (a) Test time and data volume
   (b) Test power
   (c) Scan performance overhead

Presently, scan security is one of the burning issue which needs urgent attention. The scan DFT architecture is posing a threat to security of hardware chips implementing cryptographic algorithms such as Advanced Encryption Standard (AES). Recently it has been shown in literature that the scan design can be exploited to steal sensitive data embedded on-chip such as secret encryption key.

The chip security and testability have orthogonal objectives. From a testability point of view, each and every storage element in the circuit must be observable and controllable through primary I/O pins. On the other hand, from security point of view, the visibility of circuit's internal information must be as less as possible. The scan architecture has been designed to meet the testability requirements without considering the security requirements. A test engineers operates the scan design to shift in the test vector and shift out the corresponding test response. On the other hand, an attacker can exploit the scan design and apply crafted inputs and observe the corresponding sensitive data. So, there is a strong need to secure the scan *DfT* architecture which can effectively test the chip without compromising on the security of the chip.

Other then security the scan design also inherits some testability related issues because of its serial nature. The test related attributes of scan test like test power, test data volume, and test application time are also getting difficult to control with increasing circuit complexity. The test data volume and test time directly impacts the test cost and hence the final product cost. The test power is another very critical issue. It has impact on both the test time and yield. This thesis targets both security and testability aspects of scan design for present day complex *SoC* chips. Both the security as well as the testability issues are discussed in detail in Chapter 2.

1.3 Scope of Thesis

This thesis targets two main issues faced by modern day *VLSI* chips. In first part of the thesis the security issues in chip testing has been investigated. We have proposed techniques to secure the scan design against scan-based side-channel attacks. The proposed techniques secure the scan architecture without compromising on the testability aspects of scan design.

In second part of the thesis, we have worked on some outstanding scan test issues like test power, test time, and test data volume. We have explored an alternate test architecture called *Joint-scan* test which minimizes all the above said testability issues all together. We have proposed a unified framework to integrate both serial scan and *RAS* into a hybrid architecture to harness the best of both the architecture. Further, we also have targeted implementation issues in Joint-scan test architecture.

Moreover, we have also explored structural techniques to eliminate scan cell performance and unnecessary power dissipation in combinational logic during scan shift operation. Basically, we have proposed modified scan cell designs to eliminate these scan overheads. Furthermore, an area efficient hardware-assisted scan chain diagnosis technique has been proposed.

1.3.1 Thesis Organization

The rest of the thesis has been organized as follows: Chapter 2 gives an elaborate introduction to security and testability issues in scan-based *VLSI* testing. It also discusses the state-of-the-art. Chapter 3 describes our proposed secure scan test techniques in detail. In Chapter 4 we delve into testability issues in scan test. We explore a new Joint-scan test architecture to resolve testability issues of conventional scan design. Chapter 5 explains our proposed hardware-assisted area efficient scan chain diagnosis technique. Chapter 6 explores opportunity in scan cell design to eliminate combinational test power and scan performance overhead. Finally, we summarize the whole thesis in Chapter 7 along with summary of present key contribution of this work. The future directions where this work could be taken forward has also been discussed in this chapter.

— * — * —

Chapter 2

Motivation and the State-of-the-art

Scan design is the only practical test technique available today which can effectively test the complex modern day *VLSI* chips. However, scan inherits some which were touched upon in Chapter 1. In this Chapter, we discuss the security and testability cahallenges in scan test. We also explore the existing literature addressing these issues. We first discuss the vulnerabilities in scan design that made it a target of attackers to steal sensitive dat stored on-chip. Section 2.1 explains the mechanism of scan-based side-channel attacks. The state-of-the-art is reviewed in Subsection 2.1.2. The testability issues in scan test are explained in detail in Section 2.2. A brief review of the available literatue on these issues is also given in Subsection. The Chapter concludes in Section 2.3.

2.1 Security Issues in Scan Test

In modern era of information technology, security of the information has become vital. The communication of confidential information over an unprotected channel must be secure and need to be protected from intruders. The common practice is to encrypt the secret information using a cryptographic algorithm before transmission. Several cryptographic algorithms have been proposed and practiced for centuries. Presently, the Advanced Encryption Standard (*AES*) is the most commonly used symmetric cryptographic algorithm. In a symmetric cryptographic algorithm, the data is encrypted and decrypted using a single private key which is also called encryption key. The private key is kept confidential and is not disclosed to any third party. The *AES* supports a fixed block size or plain text size of 128bit and an encryption key of 128bit, 192bit, or 256bit. The *AES* is implemented in both software as well as in hardware.

Over the last two-three decades, the amount of data to be exchanged has increased tremendously and hence high speed encryption algorithm is required. In order to communicate bulk data securely at a very high data rate, the data needs to be encrypted in real-time with very high throughput. The software-based *AES* implementation can not satisfy the requirement of encryption/decryption of data with very high throughput. In order to achieve very high throughput, the *AES* is implemented on a dedicated hardware. This dedicated hardware which implements cryptographic algorithm is often called cryptographic chip or crypto chip. There are two kinds of hardware implementations of *AES*: pipelined architecture and iterative architecture. In a fully pipelined implementation, every cycle there is a 128bit cipher text available at the outputs. The fully pipelined implementation offers highest possible throughput. However, it requires comparatively very large area. The iterative implementations, on the other hand offers reasonably good throughput and requires comparatively very less area. In iterative architecture, the plain text is converted into cipher text over a fixed number of iterative rounds of encryption. The number of rounds required to achieve sufficient level of encryption is a function of the key size. For a key size of 128bit, ten rounds of encryption are used. The level of encryption achieved after ten rounds is considered strong enough against any mathematical attack [151].

The *AES* is considered as a highly secure cryptographic algorithm and so far no brute-force or crypt-analytic attack have been reported in the literature that can break it in practical time. However, the cryptographic chips implementing *AES* cipher are found vulnerable to side channels attacks based on timing analysis, power analysis, and scan based *DfT*. As far as the security of crypto chips is concerned scan based *DfT* is a necessary evil. Despite the threat that it brings to crypto chips, the use of scan-based *DfT* can not be avoided because of the high fault coverage and diagnostic capability that it offers. Presently scan-based *DfT* is the only practical test architecture available that can test a present day highly complex *VLSI* chip with a satisfactory fault coverage. As explained in Chapter 1, scan design adds one extra mode of operation to the circuit functionality. The extra mode of operation is popularly known as scan mode or test mode among the *VLSI* test community. By using the scan enable signal, which is available at the users disposal, the circuit can be switched between functional and test mode at anytime. During the scan mode, scan chains can be loaded and unloaded serially via the scan *I/O* pins. Hence, the circuit can be brought into any known state and the present state of the circuit can be observed.

The scan architecture generally is operated by a test engineer to perform three basic operations: test stimuli loading, stimuli launch, response capture (this is performed in functional mode) and response shift-out through scan-out pin. However, a malicious user can exploit the scan architecture to observe the intermediate state of the crypto chip during encryption operation. The fact that the level of encryption of intermediate states is not very high makes them the target of attackers. The level of encryption of the intermediate state after the first round of encryption is very poor. Because of this reason, almost all the known scan-based attacks target the first round intermediate data. The intermediate data for a sufficient number of plain-text input pairs are collected through scan operation. The attacker then analyzes these intermediate states to retrieve the secret encryption key.

2.1.1 AES and Scan-based Attack

A high-level block diagram of an iterative *AES* cipher is shown in Figure 2.1. In *AES* algorithm, initially, plain-text P is bit-wise *XOR*ed with private encryption key K. This step is called *pre-round* and is performed before the first round operation only. The *XOR*ed data is then transformed by three successive layers: *S-Box*, *ShiftRows*, and *MixColumn*. Finally, the transformed data is again *XOR*ed with Round key (RK) and stored in the round register R. This completes a single round. The transformed data stored in the round register is used as input for the next round. As stated earlier, for a

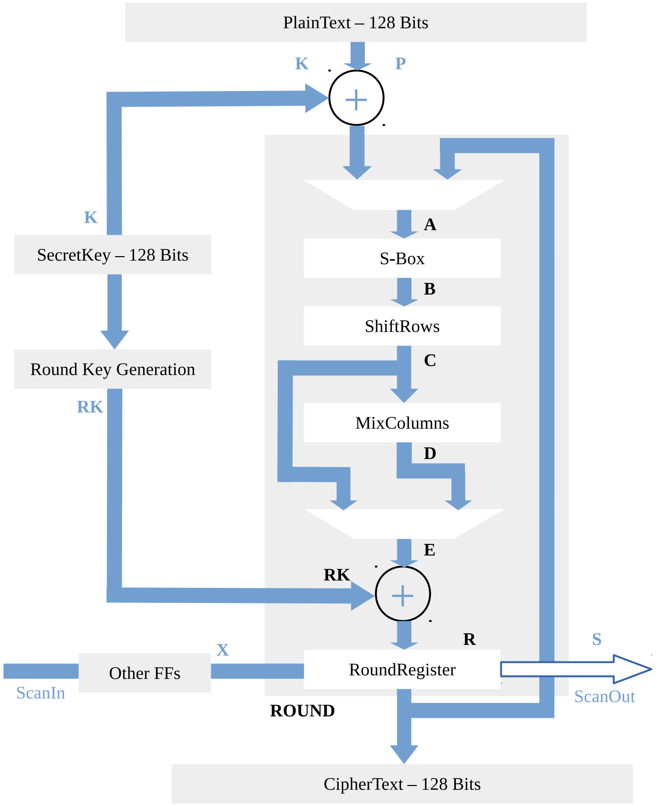


Figure 2.1: Schematic diagram of an *AES* cipher along with scan design

key size of 128bit, ten such rounds are performed iteratively to convert the plain-text into cipher-text. At the end of each round, transformed data is stored in the round register R. During the last round, i.e., the tenth round, the *MixColumn* operation is not used to make the encryption and decryption operations symmetric [151]. For every round, it uses a separate round key RK which is derived from the encryption key K. The RK is either pre-generated and stored on-chip or it is generated on-the-fly using round key generation logic. The level of encryption achieved in ten rounds is considered to have sufficient security against any known mathematical attack [151].

Now to make the *AES* circuitry fully testable, the round register R has to be included in the scan chain. As explained in the Introduction section, using the scan enable signal SE, the circuit can be switched at any time from functional mode to test mode and vice-versa. In test mode, the scan chain values can be unloaded serially through the scan-out port. The attacker uses this test feature to unload the intermediate values stored into the round register after the completion of the first round. Since the level of encryption after the first level is very poor, first round intermediate results are analysed to retrieve the original encryption key K. The attacker applies some specific plain-text (P) and runs the chip in functional mode for one round. After the first round, attacker changes the chip to test mode and unload the value of R. This procedure is done repeatedly with different plain-texts and sufficient amount of intermediate data is collected. These intermediate results are then analyzed to get the encryption key. To mount a scan-based attack on a cryptographic chip the attacker must have the capability to apply desired stimuli to the circuit and observe the corresponding responses stored in the round register. Without having controllability and observability of the round register none of the known scan-based side-channel attacks are possible.

Scan attack: How it works?

The first scan based attack was reported by Yang et al., on a Data Encryption Standard (DES) cipher [242]. The same authors later extended the scan attack to AES cipher [243]. The attack on AES is performed in two steps. In the first step, the position of all the scan cells in the round register R is determined. In the second step, the encryption key K is retrieved by analyzing the intermediate encryption data. To understand the attack procedure we need to look at the internal structure of AES encryption round logic which is shown in Figure 2.2. The AES is basically a byte oriented cipher. As can be observed from Figure 2.2, in every encryption round the four AES layers transformed the plain-text in byte wise fashion. The byte substitution or $SBox$ has 16 input bytes (A_0 to A_{15}). It transforms it into 16 corresponding output bytes, i.e., B_0 to B_{15}. It means that any change in A_0 will change only B_0. The next layer, which is *shift rows*, is nothing but a simple permutation logic implemented using only interconnect wires. The mix column performs a specific arithmetic operation based on *Galois Field*. It first mixes the four input bytes B_0, B_5, B_{10}, B_{15} and transforms them and produces four bytes of output. These four output bytes $(C_0, C_1, C_2, C_3,)$ forms a word W_0 which depends

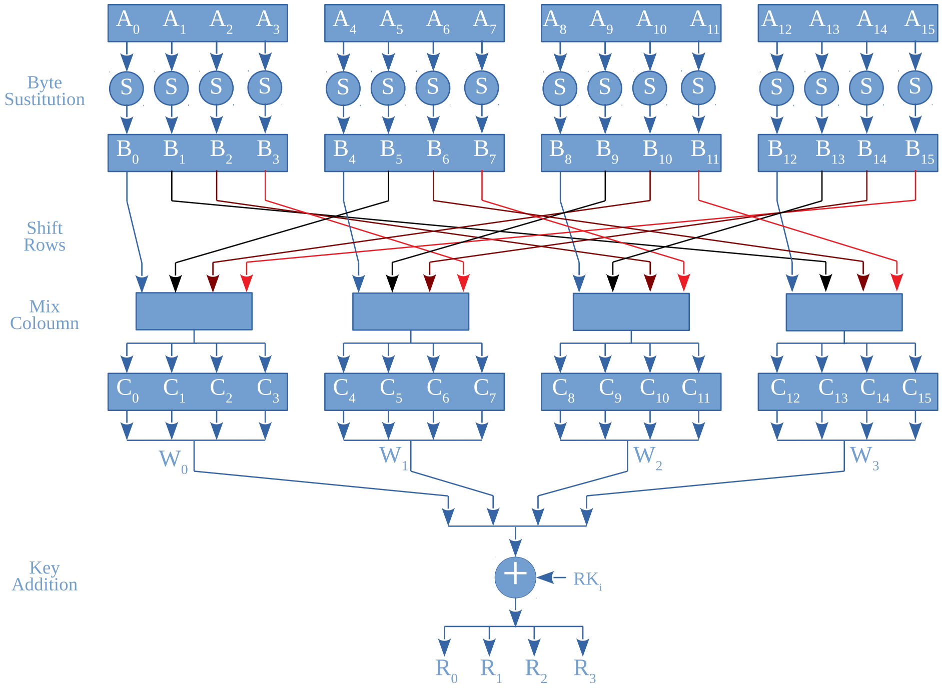

A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15
Byte Sustitution
S
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13 B14 B15
Shift Rows
Mix Coloumn
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14 C15
W0
W1
W2
W3
Key Addition
RKi
R0 R1 R2 R3

Figure 2.2: Internal structure of an *AES* cipher [151]

upon all the four input bytes $(B_0, B_5, B_{10}, B_{15})$. This simply means that any change in one of the input bytes will cause a change in W_0. After transformation by these three layers the four words W_0, W_1, W_2, W_3 are bit-wise *XOR*ed with the 128bit round key RK_i. This round key addition layer generates four words R_0, R_1, R_2, R_3 which are stored in the round register R, at the end of every encryption round.

Since, *AES* is a standardized publicly available cipher, the attacker knows the functional operation of if. He can exploit the scan design to retrieve the encryption key in two steps which are explained below:

1. Determine scan chain structure

   (a) Reset the chip, apply plain-text P $(P_0, P_1, ..., P_{15})$ and perform only one round of encryption.

   (b) Switch the chip to scan mode and shift out the intermediate encrypted data of first round stored in round register.

   (c) Repeat the first two steps by using a different P which differs from the previous P only at 1bit position in A_0. Repeat this for all 256 (2^8) combinations of A_0.

   (d) Now, change in A_0 will cause a change in B_0, which in turn will result in a change in C_0, C_1, C_2, C_3 and hence in W_0. Finally, after completion of one encryption round this change will reflect in 32 scan cells which corresponds to first round word R_0 of the round register R.

   (e) By, comparing all these 256 output patterns, the correspondence between output stream bits and scan cell in round register can be established very easily. It has been shown in [243] that on an average it takes only 6 pattern and in worst case 15 patterns to locate these 32 scan cells in R. Hence, it takes only 24 input plain-texts to determine all 128 scan cells in round register R.

2. Retrieve encryption key

   (a) In the first round of encryption, the pre-round operation is performed before the round operation. In pre-round operation the plain-text P is bitwise *XOR*ed with the encryption key K $(K_0, K_1, ..., K_{15})$. So, the input A to the byte substitution layer is given by $A = P \oplus K$, and hence $A_0 = P_0 \oplus K_0$,. So, if we know A_0 then we can find out K_0, as $K_0 = A_0 \oplus P_0$.

   (b) Apply a pair of plain-text P_1 and P_2 which differ only by 1bit position in byte P_0. This will produce an input pair A_0^0 and A_0^1 at the input of byte substitution layer. As a result we will have round word pair R_0^0 and R_0^1. Now perform *XOR* operation on R_0^0 and R_0^1. This will give the hamming distance or by how many bits R_0^0 and R_0^1 differs from each other.

   (c) W_0^0 and F_0^1 also will differ by the same number of bits with each other because
   $(R_0^0 \oplus R_0^1) = (W_0^0 \oplus RK_0) \oplus (W_0^1 \oplus RK0)$, or
   $(R_0^0 \oplus R_0^1) = (RK_0^0 \oplus RK_0) \oplus (W_0^0 \oplus W_0^1)$, or
   $(R_0^0 \oplus R_0^1) = (W_0^0 \oplus W_0^1)$.

   (d) Now the *AES* has this specific property by which a hamming distance of

$9, 12, 23, 24$ between W_0^0 and W_0^1 corresponds to a fix input pair value $226 - 227, 242 - 243, 112 - 123, 130 - 131$ respectively of A_0^0 and A_0^1.

(e) Once A_0^0 and A_0^1 are known we can find the value of first byte $K0$ of the encryption key K. It has be shown in [243] that to retrieve all the 16 bytes of encryption key on an average 544 plain-text pairs are required.

The above demonstrated scan attack can be successfully mounted on more complex scan chain structures. The authors in [54] further extended the attack algorithm used in [243] to mount an attack on advanced *DfT* architecture with feature like test decompression, mask decoder, and response compactor. Furthermore, scan-based attacks have also been shown to break the hardware implementation of public key ciphers like Rivest-Shamir-Adleman (*RSA*), and Elliptic Curve Cryptography (*ECC*) [55].

An unsecured *DfT* structure is a big threat to the security of cryptographic chips. There is always a risk of losing vital information from unsecured scan-based *DfT* architecture. So, keeping in view the security aspects of crypto chips the scan design needs to be secured without any impact on its test and diagnosis capability.

2.1.2 The State-of-the-art

Several countermeasure techniques have been proposed to secure the cryptographic chips against scan-based side-channel attacks. These countermeasure techniques can be broadly classified into three main categories:

1. Inherent Countermeasures

   (a) Advanced *DfT* architecture [54, 55]

   (b) *MISR*, mask decoder, *BIST* [86, 147, 215]

2. Countermeasures against micro-probing

   (a) Test interface un-bounding [57]

   (b) Physical probing alert [91]

3. Protocol countermeasures

    (a) Secure test wrappers [49, 150]

    (b) Encryption key masking [52, 56, 193, 243]

    (c) Test restriction [46, 90, 122, 123, 152, 162]

    (d) Test vector encryption [188–190]

The authors in [86] and [147] explored an alternate *DfT* architecture called *Built-In-Self-Test* (*BIST*) to test the *AES* circuitry. The *BIST* technique can test the circuit without exposing the scan chain to outside world. However, the *BIST* technique suffers from poor fault coverage and severely impacts diagnostic and debug capabilities. Secure test wrappers [49, 150] is another countermeasure which restricts the access to scan architecture. To access the scan test infrastructures the user needs to supply a test session key. Secure test wrapper techniques suffer from test session key management and also found vulnerable [57]. Fujiwara et al. [74] propose the use of sequential or combinational functions in the scan chain to obfuscate the content of the scan chain. The problem with this technique is that it assumes that the attacker does not have access to the scan chain design information. This assumption fails if the attacker is an insider. Further, the assumption is against Kerckhoff's principle of minimum secret information and is not considered as strong.

Another very effective countermeasure is based on lock & key [90, 122, 123, 152] method wherein test authorization keys are stored on-chip. To exercise the scan test the user needs to supply a proper authentication key. If the key matches, the scan test can be performed, else either the scan chain will be scrambled [90, 122] or the scan-out port will be masked [123, 152]. These techniques use the same test keys embedded in the test vectors itself. The analysis of the test vectors can reveal the test authorization key. To overcome this problem the approaches *SS-KTC* [46] and *SS-TKR* [162] use separate key for every test vector rather than having the same key in all test vectors. These techniques incur significant area and have test data overhead along with reduced fault coverage.

Test key masking and resetting the round register whenever the circuit switches from functional mode to test mode have been explored in [52, 193, 243]. These techniques either supply a pseudo key or an all zero or all one key to the round module instead of secret key in test mode. Hence, the scanned out internal states in test mode would be related to the fake key only, therefore securing the secret key. The techniques based on encryption key masking can effectively prevent the scan-based attacks, however, they do not allow the sanity checking of the test key itself. Furthermore, the extra logic used in implementing the key masking technique can not be tested using the scan test and need to be tested using *BIST*. Test interface unbounding is another very effective technique and is generally practiced by industry to secure the debit/credit card chips. This technique allows to perform manufacturing testing, however, in-field test can not be performed. Furthermore, physical probing based attacks can get access to the test enable signal and scan *I/O* ports.

DaRolt et al. [58] avoid test response observation at the scan-out port by making the on-chip comparison of the test response with golden circuit response. This technique has limited debug capability and loading of golden circuit response in the chip severely impacts the test time. In a recent work by Mathieu et al. [188] encryption of the scan chain content is used to counteract the security threats posed by scan-based attacks. The main idea is to encrypt the test stimuli and the corresponding fault free test responses off-chip. The encrypted test stimuli is decrypted on-the-fly during scan-in operation using the *PRESENT* [31] cipher implemented on-chip at the scan input side of the *AES* cipher. The corresponding test responses are encrypted on-the-fly during scan-out operation using the *PRESENT* cipher implemented on-chip at the scan-out side of the *AES* cipher. This restricts the attacker's capability to apply desired test values to the *AES* circuitry and to observe the corresponding responses. Hence, the attacker no longer knows what is being applied to the circuit and what is the real response of the circuit to the corresponding inputs. This technique can effectively prevent the scan-based attacks, however, the area overhead is prohibitively high. It can not be implemented for a standalone *AES* chip. Furthermore, the ciphering/deciphering *PRESENT* cipher module can not be tested using scan and needs to be tested using functional means.

In this thesis, we have proposed set of techniques to secure the scan *DfT* architecture against scan-based side-channel attacks. The proposed techniques can be used for testing of cryptographic chips in a secure manner. The proposed techniques are based on test protocol countermeasures such as encryption key masking, test restriction, and test data encryption. The proposed techniques can exercise all the conventional scan based tests without compromising the security of the cryptographic chip. All the proposed techniques are explained in detail in Chapter 3.

In rest of this Chapter, we delve into the testability issues in scan design of very complex and large designs. Because of the serial nature of scan design it inherits some issues like test data volume, test time, and test power. These issues along with the existing solutions are discussed in Section 2.2. Also, the integrity of the scan design itself is very crucial in order to apply the correct test. We also explored the scan chain diagnosis techniques in case there is a fault in the scan chain itself. Further, the performance overhead of scan design along with existing solutions is explored.

2.2 Testability Issues in Scan

The process of scan insertion and test pattern generation is a highly automated task. The use of scan design together with *ATPG* provides a very high quality test set with very high fault coverage. An *ATPG* program targets a specific fault and generates a test vector for that by assigning some specific values to a few scan cells. The rest of the scan cells are assigned random values called *don't-care* bits or *X-bits*. Further, static or dynamic compaction is used to exploit these unspecified bits to detect multiple faults using the same test vector. Even after compaction, it has been observed that the fill rates of specified bits in a test vector is anywhere in the range from 0.2% to 5% [115]. However, to detect a fault, the locations of only 10% of the specified bits in a test cube are essential. In other words these specified values cannot be replaced with specified values in other locations. On an average the number of unspecified bits in compressed test patterns are 80%-90% [115, 158].These unspecified bits are filled using *X-filling* techniques based on the criterion of low test power or *n-detect*. These overly specified test patterns are loaded into the tester memory which is a costly resource. With ever increase circuit size and complexity the test data volume is significantly impacting the test cost because of longer test application sequencing and elevated tester memory space requirements. The test time and test data volume are directly related to the test cost which constitute a large part of the overall product cost [39, 66, 97, 105, 166, 247]. The test cost for present day systems can go as high as half of the total product cost [219]. The relation between test data volume, test time and test frequency is given by the following equations [158]:

$$Test\ data\ volume \approx number\ of\ scan\ cells \times number\ of\ test\ vectors \tag{2.1}$$

Now, for balanced scan chain in which the number of scan cells are equal, the test application time can be calculated as [158]:

$$Test\ application\ time \approx \frac{number\ of\ scan\ cells \times number\ of\ test\ vectors}{number\ of\ scan\ chains \times scan\ frequency} \tag{2.2}$$

With increasing circuit complexity, the number of scan cells increase and hence the test data volume. This in turn increase the test application time and hence test cost. Furthermore, with ever decreasing feature sizes, new types of fault models are becoming necessary to cover faults other then stuck-at and delay faults. These new fault models require additional test patterns and hence increases test data volume [206].

As given by Equation 2.2, the test application time can be decreased either by increasing the number of scan chain or by increasing the test frequency. The number of scan chains which can be formed is restricted by increasing logic-to-pin ration. The scan or test frequency is decided by scan path timing and test power. Presently, the test power is the bottleneck in test frequency. The low cost ATE can drive the scan channels at at frequency ranging from $100MHz$ to $200MhZ$. The chip scan I/O pins can also supports scan shift at these frequencies. However, the internal scan chains of complex commercial chips typically operate at frequencies ranging from $10MHz$ to $50MHz$ due to power dissipation and scan path timing constraints [171]. For the present day complex

SoC chips test power dissipation is the main bottleneck in increasing the test frequency. It is desirable to exercise the scan test at maximum possible scan frequency because it directly impacts the test time and hence the overall test cost.

2.2.1 Test Time and Test Data Volume

Many techniques have been proposed in literature to minimize test data volume and hence test time. These techniques involves partial-scan, *BIST*, *RAS*, static or dynamic test compaction, multiple parallel scan chains, and test compression and response compaction. In partial-scan design, the length of scan chain is reduced by including only a subset of scan cells. Several techniques have been proposed to select the appropriate set of flip-flops to include in the scan chain. The flip-flops which are hard to control and observe are considered for partial scan chain formation [126, 199, 237]. The selection of partial scan design flip-flops can be: structure based [19, 40, 47, 85, 119], testability measures based [34, 107, 177, 237, 238], and test-generation based [7, 94, 131, 132, 187]. The partial scan design techniques provides a way of trade-off between ease of testing and cost of scan. The partial-scan techniques do not comply with the existing industry design flow and also incapable of insuring the quality of full-scan. Moreover, Increasing complexity of integrated circuits has forced the industry to abandon partial scan, which necessitates a computationally demanding and unaffordable sequential *ATPG* (or combinational *ATPG* with time frame expansion), and to rather adopt full scan despite its costs.

The use of *BIST* is another approach which eliminates the requirement of a costly tester and hence drastically reduces the test cost. The use of stand-alone *BIST* eliminates the need of tester storage space, however, it suffers from poor fault coverage due to random-pattern-resistant (*RPR*) faults [206]. Hybrid *BIST* is proposed to overcome the problem of stand-alone *BIST* by using some tester storage for detecting these *RPR* faults. The fault coverage which full scan provides can not be achieved with *BIST* based testing. Test pattern compaction is another software based technique which significantly reduces the test pattern count and hence test time. Several techniques have been proposed based on static test compaction [133, 153, 154] as well as dynamic test compaction [168]. These techniques exploit the availability of unspecified test cube values available in the test vectors.

One very effective and simple solution to reduce the test time is to break a single scan chain into n number of parallel scan chains [208, 209]. This reduces the scan chain length by a factor of n and as a result the test time reduces by a factor of n. This can effectively reduce the test shift time, however the number of scan chains that can be formed depends upon the scan channels available on the tester and the scan *I/O* pins. The available chip scan *I/O* pins is becoming a bottleneck because of the already very high and continually increasing logic-to-pin ratio [158, 206]. On the other hand tester *I/O* channels are a costly resource and directly related to the *ATE* cost. Furthermore, in some cases the test data bandwidth between the *ATE* and the chip limits the scan frequency [206].

To overcome this problem test compression and response compaction have been proposed [115, 139, 158, 160, 171, 206, 224]. These test compression schemes take advantage of low test pattern fill rates. In these schemes compressed test patterns are delivered to chip using fewer channels and an on-chip de-compressor expands them into original test patterns which are loaded into the scan chains [171]. There are mainly two fundamental test compression methods which are: single-phase and two-phase method. The single phase method uses a simple hardwired or reconfigurable fan-outs which restricts the *ATPG* by defining permanent or temporary equivalence of scan cells [206]. Some of the single-phase compression based techniques are broadcast scan [125], Illinois scan [88], adaptive scan [233], and virtual scan [220]. The two-phase method uses combinational compression [27, 140] based schemes and *LFSR* coding [111] based schemes. The *ATPG* in this scheme generates partially specified test cubes which are further encoded with some encoding scheme [206]. The *LFSR* reseeding based techniques further evolved into static reseeding [78, 87, 89, 112, 227, 232, 234] and dynamic reseeding [26, 159]. The test compression and response compaction technique greatly reduces the test data volume, however, these techniques are unable to meet the demand for the exponentially growing design size [29, 224]. Furthermore, test power dissipation in these techniques is still a big challenge and needs to be addressed [115].

Random Access Scan (RAS) architecture has been explored as a possible alternative to serial scan to overcome the test time problems all together. RAS was first proposed Ando [17] in 1980 which was considered impractical at that time. However, because of the rising test cost RAS architecture is gaining attention in the back drop of current technology scenario. In RAS architecture the flip-flops are read and write in a random access memory fashion, thereby eliminates the need for long test shift sequences. The RAS can significantly reduce the test time and test data volume. Furthermore, as the activity per pattern load is very less in RAS, hence, the average and peak power consumption which are proportional to flip-flop toggle is greatly reduced [21, 101, 217]. Therefore, the test power in RAS is much less than functional power of the chip.

The feasibility of RAS was evaluated by Wagner et al. [217] and Ito et al. [101] in 1980s, and found routing congestion and area overhead as a big concern for practical implementation. In recent works, Baik et al. [20, 21, 23] have proposed techniques to improve upon routing congestion, area, and pattern loading time for RAS architecture implementation. In [21], Baik et al. proposed a test pattern reordering methodology to minimize test time. The same author in [22, 23] addressed test time and pattern volume problems by proposing a progressive random access scan ($PRAS$) which uses a row enable shift register to address each row one at a time and parallel reading of response. The proposed $PRAS$ simultaneously achieves, on average, nearly 40% reduction in the test data volume and more than $3X$ speed up in test application time along with 99% reduction in switching activity [23]. In spite of these efforts, routing congestion is still a critical issue that needs to be resolved in order to make RAS implementation practical. Furthermore, observability of storage cells and RAS architecture implementation are some other issues which need to be addressed properly [23].

Mudlapura et al. also addressed the routing wire length problem in [143] by eliminating the Scan-in and Scan-enable lines. However, an additional gate delay is introduced in the clock path because of clock gating. To eliminate the clock gating, Adiga et al.

uses a modified T flip-flop based scan cell [3]. Furthermore, on an average $2-3$ times speed up in test write time and average 60% reduction in write test data volume has been shown [4]. Recently, the *Joint-scan* test has gained attention by the test community which tries to exploit the best of both serial scan and RAS based DFT techniques [210]. In *Joint-scan* (also called *JScan*) test architecture, a partial serial scan and a partial random access scan are integrated to form a hybrid scan architecture. The *Joint-scan* architecture provides a trade-off between hardware overhead and inherent penalties of the serial scan. However, *Joint-scan* also faces some unresolved issues. One of such issues is grouping of the scan flip-flops between the two DFT implementations, and unavailability of a single scan cell that can be used for both DFT structures. To operate the serial part and RAS part in parallel, either a common scan cell which can act as both serial cell as well as RAS cell is required or separate test clock is needed for both parts.

In this thesis, we have proposed a framework for new *Joint-scan DfT* architecture. The existing *JScan* architecture which we call *4M-JScan* and a new *2M-JScan* is proposed in Chapter 4. The proposed architecture is shown to be effective in minimizing the test time, test data volume, and test power compared to the multiple serial scan (*MSS*), and progressive random access scan (*PRAS*). We also propose a new test control mechanism which enables the architecture to function in similar way the standard scan *DfT* architecture does. Procedure for alignment of serial test vector and RAS test vector to equalize the loading time is also described. An *1D* clustering is proposed for efficient grouping/clustering of scan flip-flops. Further, we have come up with a special scan cell design which can be used to implement both the serial part as well as the RAS part of the proposed *2M-JScan*. The scan cell, can function both as a serial scan cell as well as a RAS cell and found effective in reducing the average and overall interconnect wire length. The functional and implementation details of the proposed *2M-JScan* architecture is given in Chapter 4.

2.2.2 Test Power

In scan design, to apply the test vectors the test stimuli is serially shifted into the scan chain with consecutive application of clock signal. As the test stimuli ripple through the scan cells a lot of switching happens into the scan chain. This switching in scan chain also propagates to the combinational logic and causes a very high switching activity. The loading/unloading of test stimuli/response is necessary to bring the circuit into a known state but the serial shifting has nothing to do with the actual test application. It is a well established observation that the test mode power dissipation could be significantly higher than the functional mode power dissipation [80, 249]. Furthermore, the use of aggressive low power design strategies such as dynamic voltage scaling, clock gating or power gating are widening the gap between functional and test power [80]. Therefore power-aware test for these low power devices is increasingly becoming a major concern these days.

It has been shown that the average test power could be as high as two to three times $(2-3X)$ than the average functional power [109, 115, 249]. The peak test power dissipation could be thirty times $(30X)$ higher than functional peak power dissipation [109, 115]. This huge power dissipation difference between functional and test mode can be attributed to various reasons. In functional mode, there is a high degree of correlation between consecutive cycles of data processing. However, in test mode there is a very poor correlation between consecutive test vectors [80]. Excessive average power is responsible for higher die temperature, which can provoke irreversible structural degradation. It may affect circuit performance and can have an impact on the circuit reliability, because high die temperature promotes corrosion, electro-migration, and hot-carrier-induced defects [148]. On the other hand elevated peak power causes yield loss [180]. Excessive peak power causes supply voltage droop or ground bounce and results in increased gate delays, which may cause a good chip to falsely fail the *at-speed* test [201].

Test power dissipation can be divided into two categories based on the scan operation period: *shift power* and *capture power*. The shift power is classified as the power dissipation that happens during scan shift operation which all the shift cycles and the last shift-cum-launch cycle. The shift power is further studied in terms of peak and average power [79]. The peak shift power is computed as the maximum power dissipated over all the shift cycles for the complete test vector set [172, 211]. The average shift power on the other hand is the total energy consumed for all the test vectors over the total shift periods [173]. Three main contributors to shift power are combinational toggling (switching activity in the combinational logic), sequential toggling (switching activity in the scan cells), and clock toggling (switching activity in the clock tree). Test vector loading/unloading is necessary part of scan based test, and causes sequential toggling and clock tree toggling, however test values need not to be propagated to the combinational logic. Therefore, power dissipation in combinational logic is unnecessary. It is reported by Wunderlich et al., that around $70 - 80\%$ of scan shift power is dissipated in combinational logic alone [18, 76, 118]. Thus it is very important to eliminate useless power dissipation in combinational logic during scan shifting.

Once the test vector is loaded and launched, the circuit is switched to functional mode and one clock pulse is applied to capture the response. The power dissipated during the capture cycle is called *capture power*. For stuck-at fault test there are only one capture cycle however, for delay testing, there are two or more then two capture cycles. The capture power is related to the Hamming distance between the test vector and its corresponding response. Capture power is not addressed in this thesis. This thesis concentrates only on the peak and average power minimization during shift and launch cycles.

A lot of research work has been carried out to reduce power dissipation during scan test. Numerous techniques have been proposed to reduce peak and average power during shift as well as capture operation. A very good survey of these techniques is given in [80]. These low power scan test techniques can be classified into three main categories:

1. Algorithmic or *ATPG* based [80, 222],
2. Structural or *DFT* based, and
3. System level [79, 148].

The *ATPG* based techniques mainly targets shift power by using *X-filling*, test vector reordering, input control vector, and test compaction. The *X-filling* techniques assign values to the *don't-care* bits for minimizing transition during shift operation [221, 222]. Some *X-filling* techniques also targets capture power reduction by assigning specific values to *don't-care* bits such that the hamming distance between test vector and corresponding response can be reduced [42, 129, 165, 225–227]. Huang et al. identify a specific input pattern to control the propagation of scan chain switching activity into combinational logic during shift operation [98]. Power aware merging of test cubes during static test vector compaction have been explored in [173] to minimize both average and peak power.

The structural techniques explores scan chain ordering and partitioning, power supply gating, and output gating schemes. Scan chain reordering and restitching has been explored in [53, 212, 213]. These techniques uses graph theoretic formulation to reduce both inter-pattern and intra-pattern switching activity. Output gating is another low power test scheme based on masking of scan cell's output so that no switching happens in the combinational logic. These schemes uses *AND*, *OR*, or *MUX* based output gating [61, 76]. The authors in [32, 156, 174] use clock gating to disable some scan chain and hence reduces test power in *CUT* as well as in clock tree.

The system level approaches are mainly based on test scheduling for large *SoC* chips in which testing of power compatible modules are scheduled together such that the overall test power remains below some certain threshold [50, 104, 167]. In recent years, researchers have proposed many methodologies based on design of low power scan cell [185, 203, 248]. In one such proposal by Mishra et al., an extra transmission gate has been used to isolate the slave latch during scan shift operation [138]. The authors in this technique bypass the slave latch and use a dynamic slave latch instead to propagate the scan values. Extra circuitry is required to generate the control signals to make it work for both *LOC* and *LOS* based *at-speed* delay test. This technique effectively reduces the combinational switching, however, is very costly in terms of both area and performance overhead.

In this thesis, we present a new scan flip-flop design which suppresses the redundant/useless switching in combinational logic during scan shift operation. The proposed design also reduces power dissipation inside the flip-flop itself by disabling the slave latch during serial scan of test vectors. The proposed scan cell provides a way of combinational switching suppression with comparatively very less functional performance degradation compared to the existing output gating techniques. Furthermore, the proposed design has the capability of exercising all conventionally used structural tests. The proposed design maintains the existing industrial test flow and can be easily integrated into current *DfT* flow. The proposed design effectively eliminates the useless switching activity and can have a profound impact on scan shift frequency. The proposed low power scan cell design is explained in detail in Chapter 6

2.2.3 Scan Chain Diagnosis

Almost every complex circuit today employ scan-based Design-for Testability (*DFT*) architecture to enhance testability and diagnostic capabilities. The effectiveness of these techniques rely upon the proper functioning of the scan design, i.e., the scan chain itself is fault free. However, it has been reported in the literature that the chip area consumed by the scan path along with the scan control signals may range from 15% to 30%. Furthermore, it has been observed that 10% to 30% of the total defects may cause the scan chain to fail [99]. A faulty scan chain hinders the chip failure mode analysis process for yield enhancement. The presence of a fault in scan chains can be easily detected by performing a simple flush test, however, identifying the exact location of the fault in the scan chain is a tedious task. Several techniques have been proposed in the literature for diagnosing scan chain faults. These techniques can be broadly classified into three main categories:

1. Simulation-based [84, 117, 195],
2. Tester-based [60, 194, 196], and
3. Hardware-assisted [64, 67, 145, 181, 236].

A good overview of all the above mentioned schemes can be found in a recent review article by Huang et al. [99]. The simulation-based techniques identify the faulty cell by using simulation tools which make use of the tester log data. The simulation based techniques do not have any hardware overhead, however they have poor diagnostic resolution and are very complex and time consuming. The tester based approach on the other hand uses a tester in conjunction with a physical failure analysis tool. They are very good in finding the root cause of failure, however, they are very costly and are time consuming. The third technique is based on custom scan cell design which helps in diagnosing the location of the faulty scan cell. These technique either uses modified scan cell or some extra circuitry. These techniques have very good diagnostic resolution, however, have some hardware overhead.

In Chapter 5, we have proposed a hardware assisted scan chain fault diagnosis technique. The proposed technique is very simple to implement and is capable of diagnosing both stuck-at and timing faults in scan chain. The proposed technique has the diagnostic maximum resolution for *stuck-at* faults and hence can locate the exact position of the faulty scan cell. Furthermore, the proposed technique is capable of diagnosing hold time faults with slightly diminished diagnostic resolution. In addition to that, the proposed design incurs insignificant area overhead and minimal performance penalty.

2.2.4 Test Performance Overhead

Most complex circuits today employ full-scan to obtain controllability and observability for every flip-flop in the design. In this technique, all the memory elements are replaced with muxed input D flip-flop (*MDFF*) also popularly known as scan cell. Further a test mode is added to the circuit such that when the circuit is in this mode all flip-flops are chained into one or more serial shift register (scan chain). The inputs and outputs of these scan chains are made into primary inputs and primary outputs. Thus during test mode all flip-flops can be directly controlled and observed by serially shifting in/out the logic states into the shift register. While the full scan design eliminates the sequentiality of the test generation problem, multiplexer of the scan flip-flop adds delay equivalent to two gate-delays in all clocked paths and thus degrades the functional speed. In addition, flip-flop outputs have one extra fanout (i.e., next flip-flop's test input), which increases the capacitive loading of the signal. In general, scan design can reduce the clock speed by 5 to 10% [36]. In order to fulfill the demand for high performance systems, use of very aggressive performance improvement techniques, such as minimum possible logic depth designs, magnify the necessity to eliminate the scan multiplexer performance penalty. Traditionally partial-scan has been the alternative approach to alleviate the performance penalty of full scan. Partial-scan provides a trade-off between the ease of testing and the cost associated with the scan design by selecting a subset of the flip-flops for inclusion in the scan chain. Existing partial-scan methods can be categorized as: structure based [19, 40, 47, 85, 119], testability measures based [34, 107, 177, 237, 238], and test-generation based [7, 94, 131, 132, 187].

In Chapter 6, we propose a new transistor level scan cell design to eliminate the scan multiplexer off the functional path. The proposed scan cell design uses separate master latch for functional and test mode whereas there is a common slave lath for both the modes. Our proposed scan cell fully comply with the conventional test flow. Post layout simulation results justify the effectiveness of the proposed scan cell design in eliminating the performance penalty of scan. The scan cell can be used in improving the timing performance of high performance designs.

2.3 Conclusion

From the literature survey, it can be concluded that the scan design, which is the de-facto *DfT* technique for present day highly complex chips, faces two main issues: security and testability. The earlier works on securing the scan architecture against scan-based side-channel attacks are based mainly on test protocol countermeasure.The existing techniques have some drawbacks such as fault coverage reduction, increase in test time and data volume, design area, and test session key management. Our proposed secured scan design technique which are explained in detail in Chapter 3, overcome these issues.

In terms of testability, the test data volume, test time, and test power are the main issues which needs to be resolved. These parameters are very crucial from test cost and yield point of view. A lot of research have been done to address these issues which have effectively reduced these parameters. However, still there is a need to further minimize these parameters. We have proposed a composite framework for an alternate *Joint-scan* test architecture in Chapter 4, which can resolve the test data volume, test time, and test power all together.

The success of scan test relies upon the integrity of the scan chain itself which can be subjected to manufacturing defects. The literature shows that almost half of the failing chips are failing because of the failing of the scan chain itself. We have developed an area efficient hardware-assisted scan chain diagnosis technique which is discusses in detail in Chapter 5.

The conventional scan design also have overhead in terms of performance. The literature shows that scan cell design can be exploited to eliminate performance penalty of scan. Further, scan cell design based approach can be used to eliminate the unnecessary switching activity in combination logic during scan shift operation. Also, scan cell design can be used to enable *LOS* based delay test using slow scan enable signal. Motivated from this survey, we came up with techniques based on scan cell design, which are explained in Chapter 6, that can be used to eliminate scan performance penalty, minimize scan shift power, and enable *LOS* test using slow scan enable signal.

– ∗ – ∗ –

Chapter 3

Secure Scan DfT Architecture

We discussed in Chapter 2, that how an unprotected scan design poses a threat to the security of cryptographic chips as it gives the user the capability to control/observe the circuit state. Also, the existing techniques to secure the scan design against scan attack were discussed in detail along with their advantages and disadvantages. In this Chapter, we propose a set of techniques to secure the scan design against scan-based side-channel attacks. The proposed techniques provides a way to exercise the scan test in a secure manner without compromising the security of the cryptographic chip [11–13, 214]. Moreover, our proposed techniques preserve the test capability of the conventional scan *DfT* architecture. Our proposed techniques are based on protocol countermeasures namely encryption key masking, test restriction, and test data encryption.

The rest of the chapter is organized as follows. The first technique which is based on encryption key masking is explained in Section 3.1. The second proposed technique which is based on test restriction principle is discussed in detail in Section 3.2. The last scheme which is based on test data encryption is explained in Section 3.3. Finally, the Chapter is concluded in Section 3.4.

3.1 Securing Scan through Key Masking

Our first technique to secure the scan design is based on encryption key masking technique. In this technique the encryption key is isolated from the encryption module during the test. The proposed technique masks the cipher key from the encryption circuitry as soon as the circuit is switched to test mode [11, 12]. In addition to that, the last functional state of the security sensitive scan cells is also flushed or masked. Hence, the attacker can not observe the intermediate encryption results from the last functional mode cycle. The proposed technique allows exercising all kinds of conventional stuck-at and timing tests. Furthermore, the proposed secure scan test technique has no test time overhead and uses minimal extra circuitry.

Yang et al. proposed a secure test method based on two modes: secure mode and insecure mode [243]. The proposed technique uses mirror key registers (*MKR's*) to isolate the encryption key during test process. A pseudo key is loaded in the *MKR's* through scan-chain and used for test purpose only. After the completion of test session, the circuit is switched back to secure mode by resetting the chip. Once the chip is in secure mode, the encryption key is loaded into the *MKR's* and normal encryption function can be performed. The proposed technique can effectively *fend-off* scan attacks, however, the key stored in the *mirror-register* can not be tested.

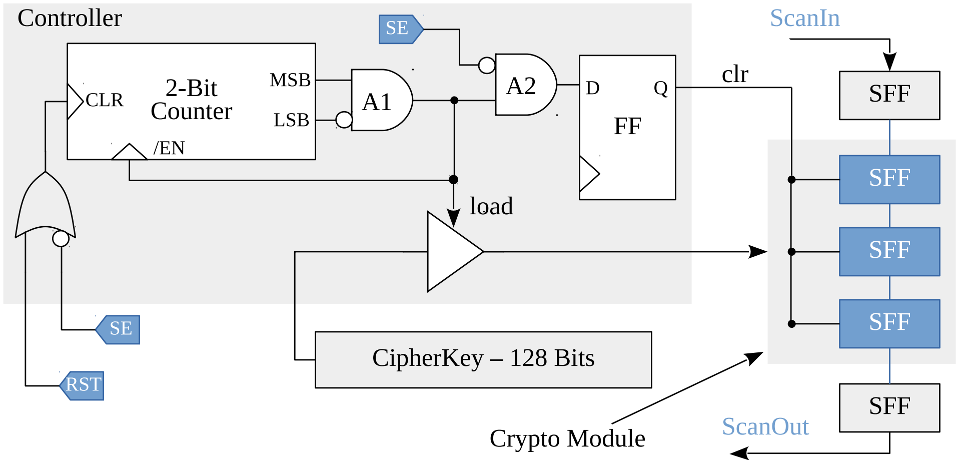


Figure 3.1: A secure scan test controller proposed in [52]

In a similar approach, Cui et al. proposed a secure scan design [52]. In this technique, a controller is used to discriminate between normal mode and test mode. In normal mode, the controller enables the encryption key to propagating to the round module, however, in test mode the key is *cut-off* from the round module. In addition to that, the controller also clears the last functional state of the round register. This technique seems to work properly, however, a closer analysis of the controller makes it clear that it can not be used to exercise launch-off-capture (LOC) based delay test. The secure design by Cui et al. is illustrated in Figure 3.1. Note that the circuit operates in functional mode when SE is high (1) and in test mode when SE is low (0). Once the test vector is loaded, SE needs to be kept high for two consecutive cycles for launching the transition vector and capturing the response. This makes the output of AND gate $A1$ high (1) at the end of the second functional cycle. To shift the test response SE is pulled low which makes the output of AND gate $A2$ high (1). Now, at the first shift cycle, the clr signal gets high and resets all the scan cells of the round register. As a result the captured response gets cleared which makes it impossible to test the cipher logic against delay faults.

Rolt et al. proposed another key isolation technique that uses a test controller to mask both encryption key and the scan-out signal [56]. It does not deliver any sensitive data until the whole scan chain is first flushed. The proposed technique can not be used to exercise multi-cycle LOC test as it resets the controller if more than two capture cycles are applied in test mode. Furthermore, The test process can be started only after flushing the scan chain. In case of circuits with single scan chain this may cause significant test time overhead. We have proposed a new secure scan test technique based on encryption key masking principle [11, 12]. The proposed technique is explained in detail in the following subsections.

3.1.1 Proposed Key Masking Technique

The high level schematic of the proposed technique is illustrated in Figure 3.2. It uses a secure scan test controller to mask the encryption key during the test mode. In addition to that, the secure scan test controller also masks the last functional state of the round register when the circuit enters into the test mode. Once in the test mode, the circuit can be tested for all the conventional *stuck-at* and delay faults. To unmask the encryption key and perform normal encryption function the controller needs to be reset which can be done using a system reset or by a dedicated controller reset signal.

At *power-on*, the circuit starts in the normal functional mode wherein the encryption key is enabled and encryption operation is performed. After *power-on* when the first time the circuit is switched from functional mode to test mode the test controller masks the encryption key. The encryption key is masked by the key masking logic shown in Figure 3.4. The secure scan test controller also flush out the last functional state of the round register R using the state masking logic shown in Figure 3.5. The test controller ensures a secure scan test without revealing the information related to the secret key. Once the encryption key is masked it remains isolated from the round module during the whole test session. The working details of the secure scan test controller, key masking logic, and state masking logic are explained in detail in the next subsections.

Secure scan test controller

Schematic of the secure scan test controller is shown in Figure 3.3. The test controller has two inputs: scan enable signal SE, and reset signal RST. Test controller introduces two new test control signals: *key_mask* and *state_mask* to mask the encryption key and intermediate encryption data stored in the round registers respectively. At *power-on*, the initial value of the flip-flop $FF1$ is 0 and the two bit counter is cleared (i.e., both MSB and LSB are 0). Also initially the circuit is in functional mode and hence the scan enable signal SE is 0. Note that the OR gate $O1$ that drives the *key_mask* test control signal outputs a logic 0 value as both of its input SE and Q are 0. As the output of the OR gate $O1$ is fed back to the $FF1$ input D, the test control signal *key_mask* remains *de-asserted* as long as SE is 0. A *de-asserted* *key_mask* signal keeps the key masking logic transparent to the encryption key. So, in functional mode the encryption key propagates to the encryption module or round logic and encryption key operation is performed normally by the circuit.

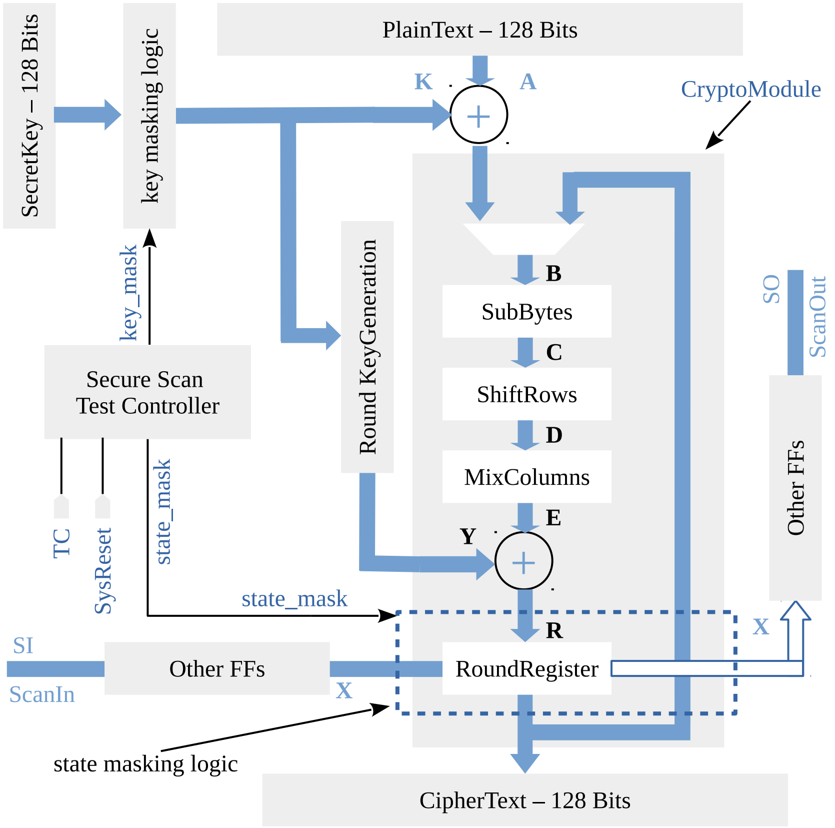


Figure 3.2: Proposed secure scan test technique schematic [12]

Since SE remains 0 throughout the functional mode, the output of the AND gate $A1$ is forced to 0 which in turn keeps the counter disabled as long as the circuit is in functional mode. Also, a logic 0 value of SE forces the output of AND gate $A2$ to a logic 0 value as SE is 0. This keeps the *state_mask* signal always *de-asserted* during functional mode which in turn keeps the state masking logic inactive. Both the test control signals *key_mask* and *state_mask* remain inactive during the functional mode of operation.

Now, to switch the circuit from functional mode to test mode the scan enable signal SE needs to be pulled to logic high (1). A rising edge on SE while the clock signal is low (0) marks the start of the test mode. As soon as SE gets from 0 to 1, the output of OR gate $O1$ gets to 1. A logic high OR gate output asserts the *key_mask* signal which in turn activates the key masking logic and isolates the encryption key from the crypto or round module. Note that the *key_mask* signal gets asserted as soon as SE turns high and masks the encryption key before the arrival of the next active clock edge. Observe that the output of OR gate $O1$ is fed back to the $FF1$ input D. So, at the arrival of the first clock cycle after switching to test mode the output Q of the $FF1$ gets permanently 1 as the flip-flop gets stuck in the feedback loop. A permanent high input at one of the OR gate's input forces the output to a constant logic 1 level. As a result, the *key_mask* signal gets permanently asserted and keep the encryption key masked during the test mode. Once the *key_mask* signal gets asserted it no longer depends on the SE signal. The SE signal can be switched back and forth between 1 and 0 any number of times to shift-in/launch test stimuli and capture test response.

Meanwhile, when SE gets to 1 the outputs of both the AND gates $A1$ and $A2$ gets to 1 as the other input of both the gates is already 1. This asserts the *state_mask* test control signal along with enabling the two bit counter. On arrival of the first clock cycle in test mode, the output of the two-bit counter becomes 01 (i.e. $MSB = 0$, $LSB = 1$). A 0 on the MSB keeps the output of inverter $I1$ at logic 1 and the *state_mask* signal remain asserted during the first clock cycle. When the second clock cycle comes the counter counts to 10 i.e., $MSB = 1$, and $LSB = 0$. A logic 1 on the MSB makes the output of inverter $I1$ low (0). This *de-asserts* the *state_mask* test control signal and also disables the counter. Note that the counter gets stuck into the feedback loop and

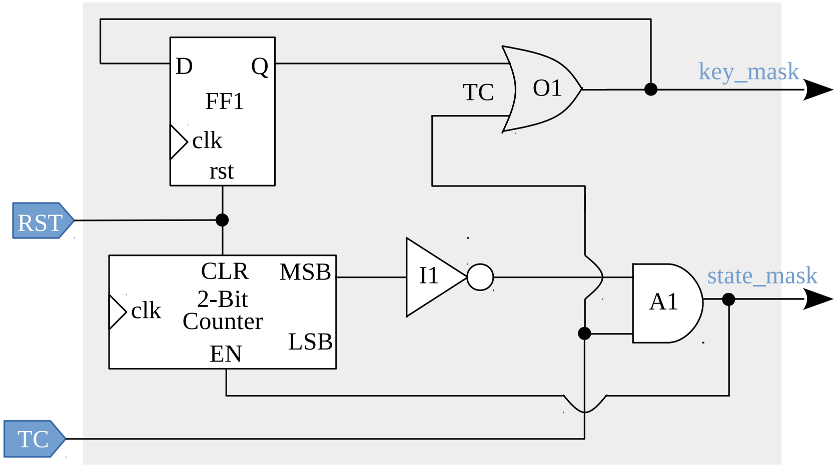


Figure 3.3: Secure scan test controller schematic [12]

stays in the same state permanently until unless the secure scan test controller is reset. Similar to the *key_mask* signal once the *state_mask* signal gets *de-asserted* it no longer depends on the *SE* signal. The *SE* signal can be switched back and forth between 1 and 0 any number of times to carry out the test process. The *state_mask* signal is asserted for only one clock cycle to flush or mask the last functional state of the round register *R*. By masking of the encryption key and flushing out of the intermediate encryption results stored in the round registers it is ensured that the attacker no longer can mount a *scan-based side-channel* attack. The details of key masking logic and state masking logic are explained in the next subsection.

Encryption key masking

The encryption key masking operation can be performed in two ways. The logic circuit to implement the key masking is shown in Figure 3.4. One way is to use a multiplexer to mask the encryption key. The encryption key can be masked by using a *two-to-one* multiplexer for masking every bit of the secret key. As can be observed from Figure 3.4, one input of the multiplexer is connected to the encryption key bit and the other input of all the 128 multiplexers are tied together to a constant 1/0 logic level. The constant 1/0 value can be either supplied from primary input pin or can be tied to *VDD*/*GND*

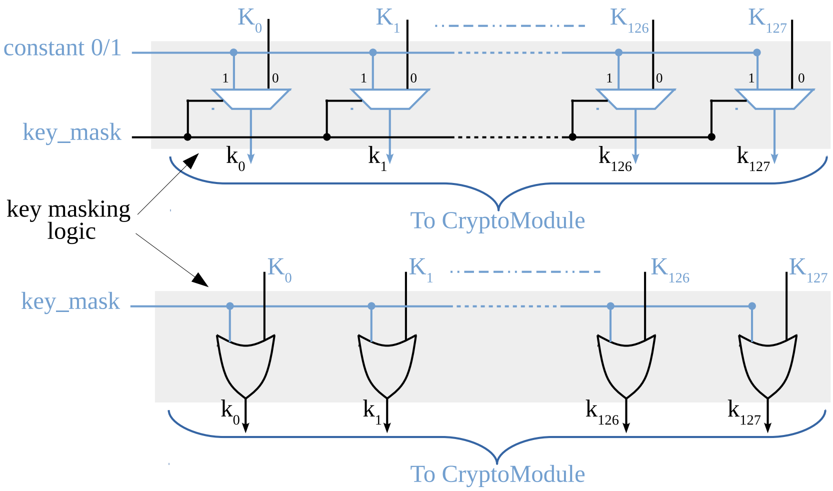


Figure 3.4: Encryption key masking logic schematic

node. In functional mode the *key_mask* signal will be 0, the encryption key input (i.e., K_0, K_1, ..., K_{126}, K_{128}) will be selected and propagate to the crypto module. On the other hand, when the circuit is in test mode (i.e. $SE = 1$) the *key_mask* will get to logic 1 and a constant 1/0 input will be selected and passed to the crypto module. Another way to mask the encryption key is to use a two input logic *OR* gate instead of a multiplexer. Again one input of the *OR* gate will be feed by the encryption key bits and another input of all the *OR* gates will be tied together to *key_mask* signal. In case of *OR* gate based key masking, a constant 1 will propagate to the crypto module during the test mode. The *OR*-based technique is much more area efficient compared to the multiplexer based masking.

Round Register State Masking

To refrain the attacker from observing any intermediate encryption data the last functional state of the round register R needs to be flushed or masked. The proposed state masking logic masks the last functional state of the round register in the first shift cycle after the circuit switches from functional to test mode. The state masking operation

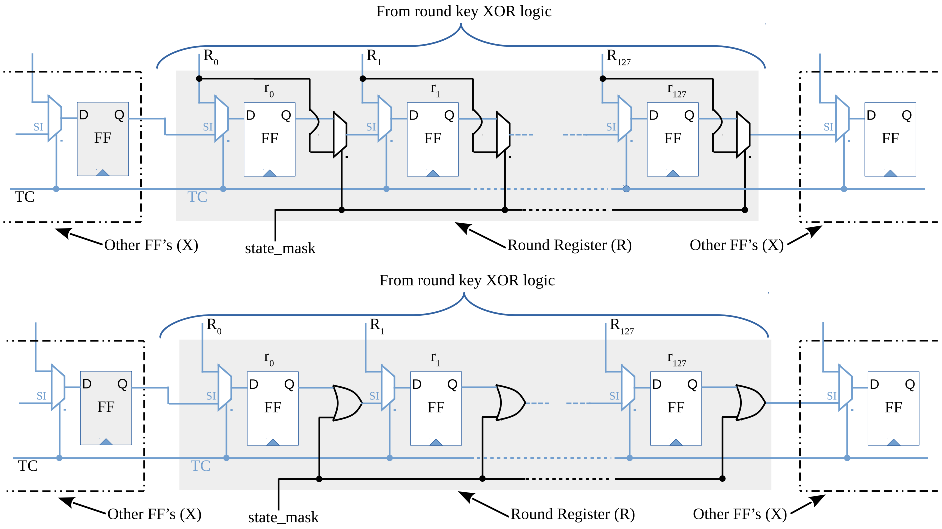


Figure 3.5: Round register state masking logic schematic

again can be carried out in two different ways. The state masking logic using two different techniques is shown in Figure 3.5. Similar to the key masking, the state masking can also be done either by using an extra multiplexer in the scan path between every pair of round register scan cells or by using an *OR* gate in place of a masking multiplexer.

As can be observed from Figure 3.5, consider the extra multiplexer inserted between output Q of round register r_0 and scan input SI of round register r_1's scan multiplexer. One input of the masking multiplexer is connected to output Q of FF r_0. Another input is connected to the data input D of the FF r_0. It should be noted that as SE gets 1, the encryption key get masked and constant 1/0 value propagates to the encryption module and thus update the output of the encryption circuitry, i.e., the intermediate encryption data at the inputs of the round register flip-flops gets updated before the arrival of the first shift cycle. This updated intermediate data do not have any information related to the encryption key and would not help the attacker in figuring out the secret key. However, the intermediate encryption data latched into the round register flip-flops have information related to the encryption key which needs to be flushed or masked.

In a normal scan shift operation, at the arrival of the active clock edge, the logic value latched into r_0 should get shifted to r_1, value latched into r_1 to r_2, and so on. However, in the scan chain with the proposed state masking multiplexer the value latched into r_0 from the last functional cycle will not be shifted to r_1. Recall that the *state_mask* signal remains asserted only for the first shift cycle in test mode. So, during first shift cycle instead of the data latched into r_0 the updated value at the data input of r_0 will be selected and will propagate to r_1. This will mask the last functional value stored into r_0 at the first shift cycle. Similarly, the masking multiplexer inserted in the scan path between every round register flip-flop pair will mask the functional state of the round register. The proposed technique also works in a multiple scan chain scenario where the round register flip-flops are stitched into different scan chains. In that case, a masking multiplexer must be inserted in the scan path between every round register flip-flop and the consecutive flip-flop.

3.1.2 Security and Testability Analysis

This section analyzes the efficacy of the proposed secure scan test techniques in terms of security and testability. It also discusses the design cost in terms of and area and performance overhead.

Security analysis

Most of the scan-based side-channel attacks exploit the scan design capability to bring the circuit into a desired state and then shift-out the content of the scan cells containing security-sensitive data. Without the capability of observing the sensitive data related to secure key none of the existing *scan-based* attacks would work. The proposed technique masks both the encryption key and the intermediate encryption data of round register at the first instance the circuit enters the test mode. The proposed technique allows the attacker to carry out *shift-in/shift-out* operation. However, the attacker no longer can access the sensitive data related to the key as it is already masked by the test controller. Hence, none of the existing *scan-based* attacks can be mounted on the security chip.

Testability analysis

The proposed technique allows to exercise all type of conventional stuck-at and delay fault tests. The test controller in [52] do not allow to apply *LOC* test and the controller in [56] is not capable of exercising multi-cycle *LOC* test. However, the proposed secure test controller allows to apply all kinds of test that a conventional scan test design can exercise. In addition to that, the original test patterns can be used without any loss of the fault coverage. However, the encryption key can not be tested by scan test as it remains masked during the whole test session. Nonetheless, the sanity of the encryption key can be easily verified using functional means. For example, the cipher text generated using the encryption circuitry in functional mode can be converted into plain text using the decryption circuitry and can be compared against the input plain text.

Area and performance overhead

In terms of area overhead, the proposed technique is most efficient compared to the existing blocking key techniques. The *NOR* gate based cipher key masking and the round register masking offers comparatively less area overhead compared to [52, 243] wherein scan cells with *RESET* capability has been used. The use of scan cells with *SET/RESET* capability requires having separate standard cell library. Furthermore, the use of *SET/RESET* scan cells complicates the overall test process. The *NOR* based key masking scheme could be much area efficient in case of pipelined *AES* architecture without key-schedule algorithm. On the other hand masking all the round keys by *MKR's* used in [243] could incur significant area overhead. In contrast to [56, 243], the proposed technique does not have any test time overhead as *shift-in/shift-out* of test stimuli/response can be started just after switching to test mode. In case of [243] the pseudo key needs to be loaded before starting the test process and in [56] the scan chain needs to be flushed out which may cause considerable test time overhead in case of a single scan chain scenario.

3.2 Securing Scan through Test Restriction

The encryption key masking principle based technique can effectively secure the scan design, however, the sanity of the encryption key can not be ensured through test process. Since, the encryption key masking principle based technique isolates the original key and use a pseudo key instead, the correctness of the original key can not be ensured by scan test. We have proposed a technique to secure the scan *DfT* architecture through test restriction. The proposed technique uses a test authorization step to unlock the scan architecture. To use the scan architecture the user first needs to supply the test authorization key. Once the user is authorized, the conventional test sequence can be started. Further, we have suggested two ways to implement the test authorization logic: *LFSR* based and *MISR* based [13, 15].

3.2.1 Test Authorization using LFSR

The proposed secure scan design is based on lock & key and key masking techniques. The schematic diagram of the proposed secure scan design technique is shown in Figure 3.6. The proposed technique uses a test authorization logic to secure the scan chain and masks the encryption key whenever the circuit is switched from functional mode to test mode. To unmask the encryption key the user needs to supply a valid test authorization key. The test authorization step is an one-time task, however, the test authorization key changes dynamically with every shift cycle. Furthermore, the proposed technique ensures that the intermediate data stored in the round register from the last functional state remain masked until a valid test authorization key is supplied by the user.

The proposed technique allows the user to use the original test vector set without any modification. In addition to that, all kinds of conventional *stuck-at* and *timing* test can be exercised without any decrease in test coverage. Furthermore, there is no test time and test data volume overhead as in the case of existing lock & key based techniques. The proposed approach has marginal hardware overhead and it is capable of preventing all the known scan-based side-channel attacks.

During functional mode, the test authorization logic remains inactive. The encryption key propagates to the round logic and normal encryption function is performed. When the first time, after *power-on*, the circuit is switched from functional mode to test mode, the test authorization logic gets activated and masks the key. The test authorization logic also masks the round register data using round register masking logic. The test authorization logic is explained in detail in the next subsection.

3.2.2 Test Authorization Logic

The test authorization logic is formed by the test controller, *LFSR*, test multiplexer *TestMux*, and the round register masking-logic. As can be seen from Figure 3.6, the logic circuitry shown in the largest gray box forms the test controller. The *LFSR* and *TestMux* form the key masking logic and the dotted line box around the round register

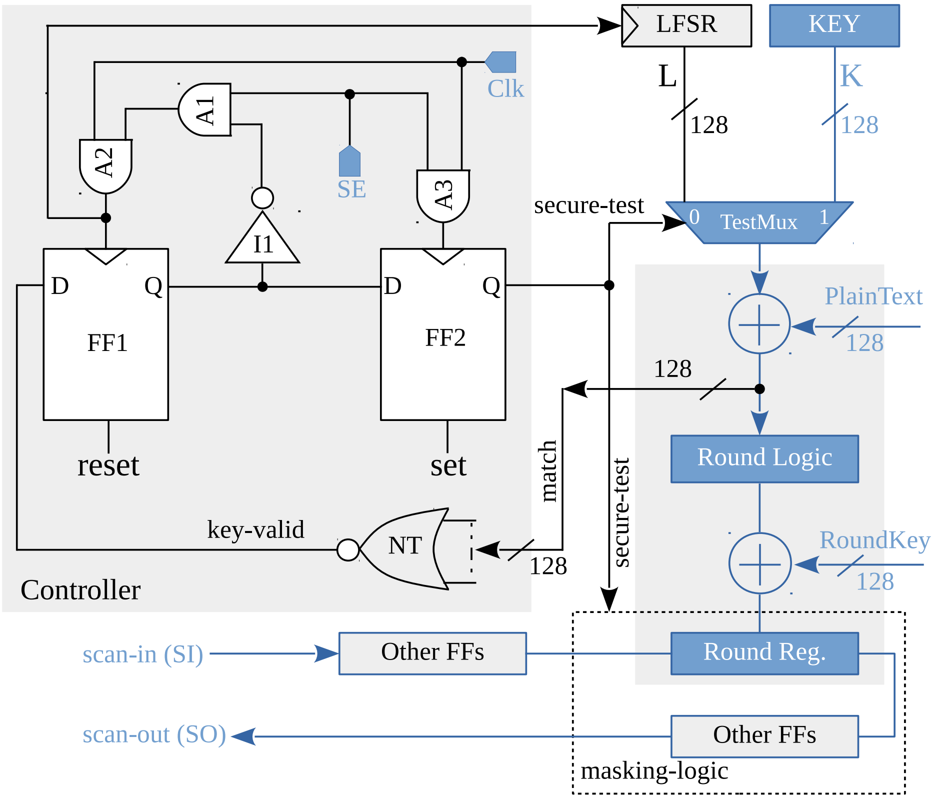


Figure 3.6: Proposed secure scan test technique schematic [13]

forms the round register masking logic. All the components of the test authorization logic are explained in detail in the following subsections.

Test controller

As can be seen from Figure 3.6, the controller comprises of two flip-flops ($FF1$, $FF2$), three two-input AND gates ($A1$, $A2$, and $A3$), one inverter ($I1$), and a NOR gate tree (NT) to logically OR 128 inputs. The controller has one input signal and one output signal. The input signal is a 128-bit signal denoted by *match*. The output signal is denoted as *secure-test*. The controller also gates the clock signal of $LFSR$.

Functional mode operation: At power-on, the scan enable signal SE is low (0) and the crypto chip initializes into the functional mode. Also, at power-on the $FF1$, and $FF2$ initialize to reset (0) and set (1) states respectively. As the *secure-test* signal is driven by the output Q of $FF2$, at power-on it initializes to logic 1 level. This keeps the *secure-test* signal always asserted during the functional mode of operation. Also, during functional mode, the *SE* signal will always be at logic low (0) level. This will keep one of the inputs of *AND* gates $A2$, and $A3$ to 0. As 0 is a controlling value for an *AND* gate, it will not allow the another input value, which is a clock signal, to propagate to the output. As a result the clock input to both the flip-flop $FF2$ and *LFSR* will remain gated as long as the circuit is in functional mode. As the clock to the *LFSR* remains gated the *LFSR* circuitry will remain idle and would not dissipate any dynamic power during functional operation. Also, the output of $A2$ feeds to one of the inputs of the *AND* gate $A1$, and $A1$ gates the clock signal to the flip-flop $FF1$. This will keep the flip-flop $FF1$ idle during the functional mode of operation. As a result both $FF1$ and $FF2$ remain idle during functional mode. During functional mode, the *secure-test* signal which controls the test multiplexer *TestMux* always remains asserted. As a result, the original key K is selected at *TestMux* and normal encryption operation is performed.

Test mode operation: After power-on, when the first time the circuit is switched from functional mode to test mode by pulling the scan enable signal *SE* to logic high level, the controller gets activated. As soon as *SE* gets to 1, the outputs of *AND* gate $A3$ becomes a function of both the inputs i.e. *SE* and Clock signal (*Clk*). Also, $FF1$ was in a reset state during the functional mode, so the output of inverter $I1$ was 1. As soon as *SE* gets 1 the output of $A1$ also gets to logic 1 level. As a result, the input of *AND* gate $A2$ also becomes a function of both *SE* and *Clk*. When the circuit switches from functional mode to test mode the clock signal of FFI, and $FF2$ gets enabled. Note that the flip-flop $FF1$ always remains reset (0) and flip-flop $FF2$ always remains set (1) during the functional mode. Now, at the first test clock or shift cycle the value of $FF1$ (i.e., 0) propagates to $FF2$ and de-asserts (0) the *secure-test* signal. As a result, the *LFSR* output L gets selected at the *TestMux*. This masks the encryption key K, and the *LFSR* output L propagates to the round logic. The secure-test signal also masks the scan-out port and does not allow the user to scan out the intermediate data related to the encryption key stored in the round register. In order to apply the test, the user needs to supply a valid test authorization key at the plain text input port of the crypto chip.

As can be observed from Figure 3.6, the output of *TestMux* drives one of the inputs of the XOR gates which are part of the original crypto circuit. Now, as the *LFSR* output sequence L which is selected at the *TestMux*, it drives one of the inputs of the *XOR* gates. The *LFSR* output sequence L which changes every cycle acts as a test authorization key. As stated earlier, the *XOR* gates are part of the original crypto circuit and perform the pre-round operation in which the original key is logically *XOR*ed with the plain text. These *XOR* gates are used to compare the test authorization key supplied by the user with the *LFSR* output sequence L. If all the 128 bits of the test authorization key match with L then all the *XOR* gates will produce a logic low output because the *XOR* gate is an inequality detector. In other words, all the 128 bits of the *match* signal will be 0. The *match* signal feeds the *NOR* gate tree *NT*, which in turn drives the *key-valid* signal or the input D of the flip-flop $FF1$. If a valid test authorization key is supplied by the user the *key-valid* signal will get to logic 1. This will make the $FF1$ to latch a logic 1 value at the arrival of the active edge of the clock cycle. However, the *secure-test* signal remains de-asserted until the arrival of the active edge of the next clock signal because $FF2$ latches the output of $FF1$ from the previous clock cycle which was 0.

It should be noted that once the output of $FF1$ gets high it's clock input will get permanently gated. It is because a logic 1 value forces the output of inverter $I1$ to 0, which in turns forces the output of $A1$ to 0. As a result, the output of $A2$ becomes 0 and masks the clock input of $FF1$. This makes the test key authorization a one-time task. When the active edge of the next clock arrives, the value latched into $FF1$ propagates into $FF2$ and asserts (1) the *secure-test* signal. This, in turn, will select the encryption key K at *TestMux*. Once the encryption key is selected, the original test vectors can be applied.

The proposed test authorization logic gives the flexibility to supply the test authorization key during any cycle starting on from the first shift-in cycle to the second last shift-in cycle. However, the one-time test key authorization needs to be done one cycle prior to the launch of the first test vector. With a minor modification in the test controller, the test key authorization can be enforced with for each test vector. However, in that case, the test data volume will increase significantly. As stated earlier, the test authorization logic also includes the round register state masking logic which refrains the unauthorized user to have access to the intermediate data stored in the round register. The round register state masking logic is explained in detail in the following subsection.

Round register state masking

Most of the scan-based side channel attacks target the intermediate encryption results after the completion of the first round which is stored in the round register. The last functional state of the round register R needs to be flushed out or masked to fend off the scan attack. We propose three schemes to mask the last functional state of round register. The three schemes whose schematics are shown in Figure 3.7 are: A) scan-out masking, B) round register bypassing, and C) scan-out feedback.

In scheme A the scan-out port is masked using a simple AND gate. One input of the AND gate is controlled using the *secure-test* signal. As explained in the previous subsections the secure-test signal gets 1 only after a valid test authorization key is supplied, until then it remains 0. As a result, the output of the masking AND gate or the scan-out port is forced to a constant 0 value. This will prevent an unauthorized user to shift-out

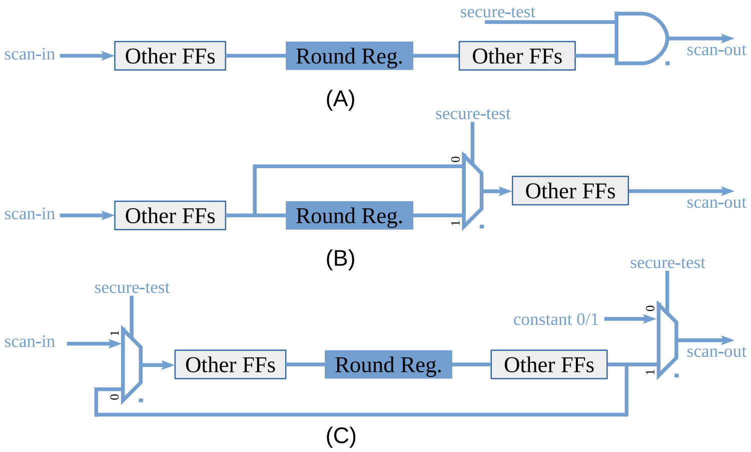


Figure 3.7: Round register state masking logic schematic

the round register data. On the other hand, as soon as a valid test authorization key is supplied the scan-out port gets unmasked. As stated in the previous subsection the proposed test authorization logic gives the flexibility to supply the test authorization key anytime between the first shift cycle and the second last shift-in cycle. However, this limits the debug capability of the scan design because with every cycle the last functional state data stored in the scan chain keep on getting masked. In order to have full debug capability, the user needs to supply the test authorization key in the very first shift cycle so that the last functional state can be unloaded without any loss of data due to masking.

In scheme *B*, the round register is bypassed using a multiplexer. The multiplexer is controlled using the *secure-test* signal. As explained earlier, the *secure-test* signal remains de-asserted (0) until a valid test authorization key is supplied. Hence, any unauthorized attempt to access the intermediate data will result in bypassing the round register in the scan chain. This scheme could be very effective in case the crypto circuit is part of a bigger *SoC*. In such a scenario, the attacker will get only the scan chain data from the other scan cells which may be part of the *SoC*. The round register bypassing technique can be easily extended to multiple scan chain configuration, where round register scan cells might be distributed or stitched among multiple scan chains. It should be noted that similar to the scan-out masking, in order to achieve full debug capability the user needs to supply the test authorization key in the very first shift cycle.

The third round register scheme is based upon the scan-out feedback. As can be observed from Figure 3.7, the scan-out is fed back to the scan-in using a multiplexer. In case there is an unauthorized attempt to access the scan chain the *secure-test* signal will get de-asserted. As a result, a constant 1/0 value will be selected at the scan-out multiplexer and passed to the scan-out port. At the same time, the multiplexer at the scan-in port will select the feedback value of the scan-out port. This will make sure that with every test clock cycle the last functional state value in the scan chain will shift by one bit. This will provide the user full debug capability irrespective of the cycle in which the test authorization key is supplied. The proposed technique also works in a multiple scan chain scenario where the round register flip-flops are stitched into different scan chains. In that case, a masking multiplexer and a feedback multiplexer must be placed at the scan-out and scan-in ports of every scan chain respectively.

Once the test process is complete the circuit can be switched from test mode to functional mode by just pulling down the *SE* signal to a logical low (0) level. However, the test controller needs to be brought into its initial state. This can be done either with chip reset or a dedicated test controller reset input.

3.2.3 Security and Testability Analysis

Security analysis

The proposed technique offers very high security against scan based side channel attacks without any test time and test data volume overhead. In order to perform a successful attack on the proposed secure scan design, the attacker needs to have the following information.

1. Size, polynomial, and seed of *LFSR*
2. *LFSR* output bit positions corresponding to the Key.

We assume the attacker model where the attacker is an insider and have access to the design parameters like *LFSR* size and polynomial. We further make the assumption that the initial seed of the *LFSR* is managed or embedded using the same method or logic in which the encryption key is managed or embedded. In such a scenario, the attacker must try 2^{128} number of combinations for a 128-bit *LFSR*. This is as difficult as guessing a 128 bit encryption key. There is a trade-off between the *LFSR* size and the security level and area overhead. In an attacker model where the attacker is not an insider and do not have access to *LFSR* design implementation details, a smaller size *LFSR* size can be used. We have implemented the proposed design using both 128 and 64 bit *LFSR*. As the proposed technique do not allows the attacker to observe the scan-out port without supplying the valid test authorization key, the known plain-text attack against the *LFSR* can not be carried out [151].

Testability analysis

The proposed technique allows to use the original test vector set without any change. In contrast to the key masking techniques in [52, 56, 243] wherein a pseudo key is used during test and the encryption key remains masked, the proposed technique carry out the test process with the original encryption key. As a result, the test also verifies the sanity of the encryption key embedded into the hardware. Furthermore, the proposed technique allows exercising all types of conventional stuck-at and timing fault tests that a conventional scan design can exercise which is not the case in [52, 56]. The test controller in [52] lacks LOC test capability and the controller in [56] does not allow multi-cycle LOC test. In addition to that, since the original test set is used there is no loss in fault coverage.

Test data volume and test time

To start the test process the user needs to supply the test authorization key only once. Because of this, there is no test data volume overhead other than the 128-bit test key. With minor modification, the test controller can be extended easily to enforce test key authorization for every test vector. However, in that case, the test data volume will increase by $128 \times N_{TV}$, where N_{TV} is the total number of test vectors. It can be observed from Table 3.1, the proposed technique do not incur any test time overhead whereas the techniques in [46], and [162] have 0.18%, and 7.64% test time overhead respectively. As the user has the flexibility to supply the test authorization key anytime during the scan shifting, the loading/shift-in of test vector can be started as soon as the circuit is switched into test mode.

Area overhead

The proposed technique has marginal area overhead compared to the existing lock & key and key masking techniques. We have synthesized the proposed design for 128 and 64 bit $LFSR$ size using Synopsys's Design Compiler using $45nm$ standard cell library. It can be observed from Table 3.1, that the proposed design has 0.90% and 0.48% area overhead

Table 3.1: Test time and area overhead comparison

Design → Overhead ↓	*SS-KTC* [46]	*SS-TKR* [162]	*Proposed* (128bit-LFSR)	*Proposed* (64bit-LFSR)
Time	0.18%	7.64%	nil	nil
Area	6.03%	2.58%	0.90%	0.48%

for 128 and 64 bit *LFSR* size respectively which is minimum compared to SS-KTC [46], SS-TKR [162], and MKR [243]. The area overhead in case of [46], [162], and [243] is as high as 6.03%, 2.58%, and 1.38% respectively.

3.2.4 Test Authorization using MISR

Another way to implement the test authorization logic is to use *MISR* instead of *LFSR*. The *MISR* authorization logic is used to obfuscate the plain-text inputs as well as the scan out port whenever the circuit enters in test mode. In order to unmask the *plain-text* inputs and the scan-out port the user needs to validate his authenticity by supplying a specific input value at the plain text inputs. If the user is authenticated the *plain-text* inputs and the scan-out port get unmasked to carry out the conventional scan test procedure can be started else the scan-out port remains masked.

3.2.5 Test Authorization Logic

The schematic design of the proposed *MISR* based secure scan test architecture is shown in Figure 3.8. In addition to the regular *AES* design and scan test circuitry the proposed design has three extra logic modules named *secure test controller*, *plain-text obfuscation logic* and *round register masking logic*. These three logic blocks forms the test authorization logic. The next subsection explains the functionality of the test authorization logic in detail.

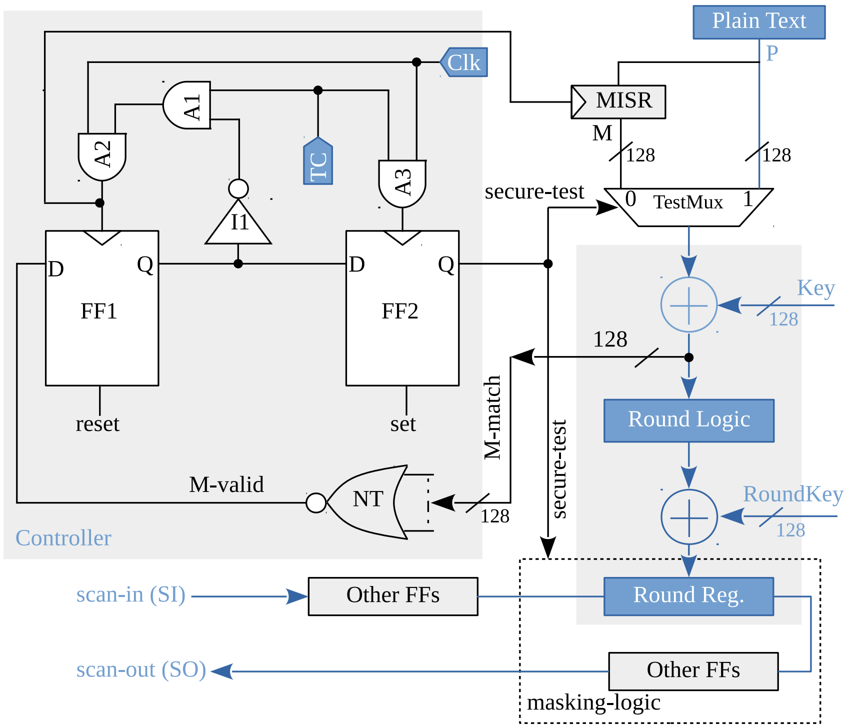


Figure 3.8: Schematic design of the proposed secure scan test architecture

Secure test controller

It consists of flip-flop $FF1$ and $FF2$ with *RESET* and *SET* features respectively. Also there are three *AND* gates $A1$, $A2$, $A3$, an inverter $I1$, and a negated *OR* gate tree *NT*. The input signals to test authorization logic are scan enable signal *SE*, clock signal (*Clk*), and *M-match* signal. Here *M-match* signal is bitwise Exclusive-OR of encryption key K and output M of the *MISR*. The output signals of test authorization logic is *secure-test* signal. Further, the test controller also gates the clock signal of the *MISR*.

Plain-text obfuscation logic

It consists of a Multiple Input Signature Register (*MISR*) and 128bit input $2x1$ multiplexer called *TestMux*. The *TestMux* selects between the MISR's output M and plain-text P depending upon value of the *secure-test* signal. Here the *MISR* is assumed to have a polynomial of degree 128 i.e it has 128 flip-flops in it and has 128 multiple inputs. These 128 inputs are connected using a random permutation to the 128 bits of plain-text inputs. Similarly, the 128 output bits of *MISR* are connected to the *TestMux* using a random permutation. The initial seeds of *MISR* is assumed to be managed by the secret key management logic block. At power-on, the *MISR* will be seeded by the secret key management logic. Note that *MISR's* output M is a function of initial seed, polynomial of *MISR* and plain-text P. Hence, M is called an obscured version of P.

Round register masking logic

The scan-out port is masked by using the round register masking logic which is controlled by the test controller through *secure-test* signal. A de-asserted *secure-test* signal (i.e., at logic 0) masks the scan-out port. The round register masking logic restricts an unauthorized user to peek at the intermediate values stored in the round register. The masking logic can be implemented by any of the simple masking logic schemes are shown in Figure 3.7.

Functional mode operation

The test authorization logic always remains inactive while the chip is in functional mode and normal encryption operation is performed. The chip initializes in functional mode at power-on. The scan enable signal *SE* is at logic 0 level and the secure test controller flip-flops $FF1$ and $FF2$ initializes to *RESET* and *SET* logic respectively. The *AND* gate $A1$ is driven by test control signal TC, which in turn drives gate $A2$. As the value of *SE* in functional mode always remains 0, the outputs of $A1$ and $A2$ remain freeze at logic 0 throughout the functional mode operation. As a result the *Clk* signal to FFI and $FF2$ remains gated and hence, disabling the clock of $FF2$ and $FF1$. So, the output of $FF1$ and $FF2$ do not change in functional mode. The *secure-test* signal which is driven by output of $FF2$ remains asserted, i.e., constant 1. A constant 1 *secure-test* signal at selection line of *TestMux* will make plain-text propagate to *AES* logic. Hence in functional mode, the plain-text value will propagate in the *AES* round logic. Also, note that since the clock signal of *MISR* is also gated by $A1$, the *MISR* will also remain inactive during functional mode of operation.

Test mode operation

The test controller gets activated as soon as the circuit enters in test mode. The circuit is switched to test mode by pulling *SE* signal from 0 to 1. Changing *SE* to 1 will make the output of $A3$ to be driven by clock signal *Clk*. This enables the clock signal of $FF2$. Also, inverter $I1's$ output is still 1, which is a non-controlling value for AND gate $A1$, hence the output of $A1$ get 1. This in turn will make one of the inputs of $A2$ a non-controlling value. As a result, the output of $A2$ will be driven by clock signal Clk. This will enable the clock signal of $FF1$. So now, both $FF1$ and $FF2$ have their clock signal enabled (unmasked). Any changes at the input of these flip-flops will be latched into them at the arrival of the next clock pulse.

Assume that, when *SE* is switched from 0 to 1, *M-valid* signal is initially 0. The validity of this assumption is proved in Section 3.2.6. As can be seen from Figure 3.8, *M-valid* is the negation of the output of logical *OR* of all 128 bits of *M-match* signal. After *SE* is switched from 0 to 1, on the first clock pulse, the value 0 of *M-valid* signal gets latched in $FF1$ and the previous 0 reset output of the $FF1$ gets latched in $FF2$. The value of *secure-test* signal changes to 0. This in turn selects input M at the inputs of *TestMux* and pass it to the AES logic. Since the *secure-test* signal is de-asserted it masks the *scan-out* port. This prohibits the user to scan out the intermediate values stored in the round register. Hence, an unauthorized user can not analyze the intermediate round register values to get the encryption key. Without having access to the intermediate data no scan attack is possible.

Test authorization

To perform scan test the user needs to go through the test authorization step and get authenticated. For this, the output of flip-flop $FF1$ and $FF2$ has to be changed to 1. This is only possible if *M-valid* can be changed to 1, which requires all 128 bits of *M-match* to be 0. For that M should be exactly match to K. Remember, that in test mode after one clock cycle, M gets propagated to AES logic. Now K is embedded securely on chip and user have no access to it. So in order to pass test authorization step and unmask *scan-out* port the user needs to supply a plain-text P such that after obscuring through the *MISR* proper M gets produced. This process of supplying proper P is called *test authorization process*. Hence in test mode user applies proper P for one cycle, then in next cycle, direct plain-text gets selected and propagates to the *AES* logic. Also, the scan-out port gets unmasked. Now the user can shift In/Out test vector/response and carry out the conventional test operation. Note that, once the M-valid value propagates in $FF1$, the input of inverter $I1$ becomes 1. This in turn masks the clock input of $FF1$ and disables it. Hence, the controller now remain freeze in the same state until unless it is reset. The controller can be reset either using the chip power reset or a dedicated test controller reset signal.

3.2.6 Security and Testability Analysis

The security and testability analysis along with test time and test data volume overhead has been analyzed in this section.

Security analysis

The proposed idea permits the sensitive data to be scanned out in a secured way by having a test authorization and masking logic. Earlier, assumption was made that M-valid will be 0 when the chip functionality is changed to test mode. This is a pretty fair assumption, as only one of the 128bits of *M-match* signal being 1 will drive it to 0. The probability of *M-valid]* to be 1 is $1/(2^{128})$. Hence the probability of having the signal value of secure-test equal to 0 is same as exhaustive searching the encryption key itself. Hence in test mode, the probability of P getting selected by the *TestMux* is just $1/2^{128}$. It's kind of a lock & key scenario i.e., P will be get locked as soon as changed to test mode and only one unique combination of plait-text input will unlock it. In order to attack the proposed design, the attacker must have the following design information:

1. *MISR* input bit positions corresponding to the Plain-text

2. *MISR* size, polynomial equation, and initial seed value

3. *MISR* in/out bits permutation with plain-text/key bits.

Assuming that attacker has no information about above design parameters would be quite an optimistic thought. Considering that the attacker is an insider then he/she will have information about the *MISR* implementation such as size, polynomial, and input/output bit permutations. However, the attacker would not have the initial seed value which is assumed to be managed by key management logic. In that case the probability of guessing the seed is as difficult as guessing the encryption key itself. Hence we can claim that security provided by the proposed idea is very high. If we go with the optimistic thought that the attacker is an outsider and has no information about the design and security infrastructure at all, then there exists a trade off between *MISR* size and security level with area overhead. As the output of *MISR* is a function of size, polynomial, and seed input bit position and output bit position, the level of security would be very high even with a 64bit *MISR*.

Testability analysis

As far as the quality of test is concerned the proposed technique allows to exercise all kinds of conventional tests which is not the case in earlier proposal presented in [52, 56]. Since the test vector set of the original design can be exercised without any modification, hence there will not be any drop in the fault coverage. Furthermore, unlike the previous secure scan test techniques [52, 56, 243] which use a pseudo test key, the proposed technique carry out the test process with the original encryption key. Testing the *AES* with original verifies the correctness of the embedded key as a byproduct.

Test time, test data volume, and area overhead

The test authorization is carried out only once at the start of the test session. Hence, in terms of test data volume overhead the proposed scheme uses only 128-bit extra key data. Furthermore, as can be observed from Table 3.2, the proposed scheme does not have any overhead in term of test application time compared with the existing schemes. The scheme proposed in [46] has 0.18% test time overhead, while it is 7.64% in case

Table 3.2: Test time and design overhead

Design → Parameter ↓	*SS-KTC* [46]	*SS-TKR* [162]	*Proposed* (128bit-MISR)	*Proposed* (64bit-MISR)
Test time	0.18%	7.64%	–	–
Design area	6.03%	2.58%	1.22%	1.08%

of [162]. Moreover, the proposed scheme incur least area overhead compared with the earlier schemes proposed in [46], [162], and [243]. We synthesized the proposed secure scan design using 45nm standard cell library with two different *MISR* sizes of 64bit and 128bit. The proposed design has 1.08% and 1.22% area overhead for 64 and 128 bit *MISR* size respectively. On the other hand the area overhead for the schemes proposed in [243], [162], and [46] is 1.38%, 2.58%, and 6.03% respectively.

3.2.7 Securing Scan through Plain-text Restriction

The techniques to secure the scan design based on test restriction uses a scan key which is used for test authorization purpose. In these schemes it is assumed that the scan key is managed by the encryption key management logic. The use and management of multiple keys is a disadvantage from security point of view. To avoid the use of scan key, we improved upon our test restriction schemes, proposed in previous sections. We propose a scheme to secure the scan test through *plain-text* restrictions.

The schematic diagram of the proposed secure scan design is shown in Figure 3.9. The proposed technique restricts the value of *plain-text* to a fix value during scan or test mode. The *plain-text* inputs can take only an *all-0* or *all-1* value during scan mode. The *all-0* or *all-1* value simply means that all the *plain-text* input bits must be either 0 or 1. No other input value is allowed at the *plain-text* inputs during scan mode. To switch the cryptographic chip from functional mode to test mode the *plain-text* input must be at *all-0* or *all-1* value during last functional cycle else the circuit will not switch to scan mode. Also, this restriction must be obeyed throughout the test session. Recall that all the scan-based side-channel attacks uses a differential input pairs at the

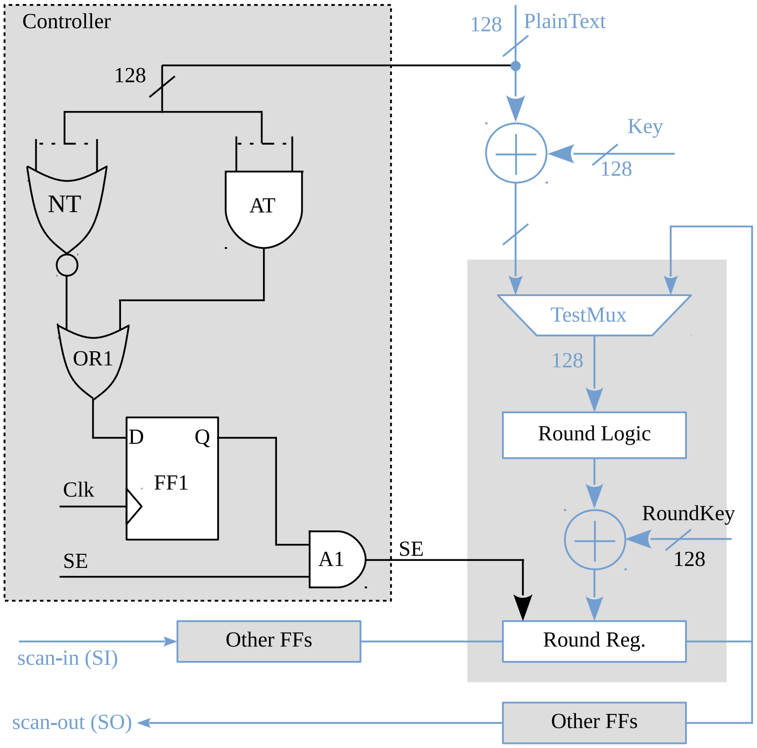


Figure 3.9: Schematic diagram of the proposed secure scan design

plain-text and observe the corresponding differential output responses after one round of encryption operation. The encryption key is retrieved by analyzing these differential output responses. As we are restricting the *plain-text* inputs to *all-0* and *all-1*, there is no possibility for the attacker to get differential response data for different differential input *plain-text* pairs. Hence, none of the know scan attack can be mounted.

3.2.8 Test control Logic

The proposed *plain-text* restriction scheme uses a simple test controller that enforces the *all-0* and *all-1* restriction. As can be seen in Figure 3.9, the controller is formed by an *AND* tree, an inverted output *OR* tree, an *OR* gate *OR1*, an *AND* gate *A1*, and a flip-flop *FF1*. The inverted output *OR* tree enforces the *all-0* restriction. The output of *NT* gets to logical 1 only when all the *plain-text* inputs bits are 0. On the other hand the *AND* tree *AT* enforces the *all-1* condition. The output of *AT* gets to logic 1 only when all the input bits of *plain-text* is 1. Now, the inputs of *OR1* are driven by the output of *NT* and *AT*. The output of *OR1* gets to logic 1 only at least one of its inputs is 1. Which means the *OR1* gate produces a logic 1 value only when the *plain-text* is either at *all-0* or *all-1*. Now, the output of *OR1* is latched into the flip-flop *FF1* at the arrival of the clock pulse. The value latched in *FF1* is used to mask or gate the scan enable signal *SE* using *AND* gate *A1*. If the value latched into *FF1* is 1 the *SE* signal is allowed to pass else if the value latched in to *FF1* is 0 the *SE* is masked. This simply means that the circuit will be allowed to switch to scan mode only if the *plain-text* inputs are at *all-0* or *all-1*. The following subsection explains the working of the proposed scheme in detail in both functional and test mode.

Functional mode operation

The scan enable signal *SE* always remains 0 during functional mode of operation. This makes one of the inputs of *A1* always at a controlling value i.e., at logic 0 value. As a result, the output of *AND* gate *A1* is always at logic 0. This means, the *all-1* and *all-0* test condition are not in effect during functional mode. The secure test controller remains inactive and the chip performs the regular encryption operation.

Test mode operation

The secure scan test controller enforces the *all-0* and *all-1* restriction during test mode of operation. To switch the circuit from functional mode to test mode the *plain-text* inputs must be kept at *all-0* or *all-1* values during the last functional cycle. If the *plain-text* are at either *all-0* or *all-1* value then at the end of the clock cycle, a logic 1 value will be latched in to flip-flop *FF1*. This will unmask the scan enable signal *SE*. Now the *CUT* can be switched to test mode by pulling up the *SE* signal to logic 1. However, if the *all-0* or *all-1* condition is not met in the last functional cycle a logic 0 value will be latched in to *FF1*. As a result the *SE* signal will be masked and *CUT* will not switch to test mode.

Once the circuit is in scan mode, test vector can be serially scanned in to the scan chain. It should be noted that the *all-0* and *all-1* condition is evaluated in every cycle. So, if in any cycle the restriction is violated, the circuit will switch to functional mode in the following cycle. Once the test vector is loaded and applied, the circuit can be switched back to functional mode and response can be captured by applying a functional clock pulse. Note that during response capture cycle the *plain-text* inputs must be kept at *all-0* or *all-1* value so that the *FF1* latch a logic 1 value in capture cycle. This will allow to switch the circuit again to scan mode, and shift operation can be performed. Hence the *plain-text* must always obey the *all-0* and *all-1* restriction throughout the test session.

3.2.9 Security and Testability Analysis

This section assesses the effectiveness of the proposed techniques in securing the scan design against the known scan attacks. Further, the testability aspects and design cost is also assessed.

Security analysis

The proposed technique enforces the *all-0* and *all-1* condition on the *plain-text* inputs throughout the test session. The violation of this condition during any cycle in test mode will result in automatically switching of the *CUT* to functional mode. Further, since the *all-0* and *all-1* condition must be enforced from the last function cycle, hence the attacker only can observe intermediate encryption result corresponding to only two values of *plain-text*. As a result, the attacker no longer can collect intermediate encryption data from various differential input pairs at *plain-text* inputs. Hence, the encryption key can never be retrieved from the round register data. Hence, the proposed technique can thwart all the known scan-based side-channel attacks.

Testability analysis

The proposed technique preserves the test capability of the conventional scan *DfT* architecture. It allows to exercise all the conventional scan based static and timing test.

Table 3.3: Fault coverage and design overhead

Parameter →	*# Faults*	*Fault Coverage*	*Area* (μm^2)	*Area overhead*
AES with scan	32138	99.54%	13599.04	−−
Proposed	32138	99.54%	13808.00	1.54%

Furthermore, there is no loss in fault coverage due to the *all-0* and *all-1* test restrictions on the *plain-text* inputs. The *plain-text* inputs values are assigned to detect faults only on the pre-round logic input and output nets. The round logic faults are targeted through the scan chain. The stuck-at fault coverage for both conventional as well as the proposed secure scan design is put into Table 3.3. Furthermore, the proposed secure scan architecture does not put any other restriction other then the *all-0* and *all-1* conditions on *plain-text*.

Design cost

In terms of area overhead the proposed technique incurs only 1.54% extra area compared to the conventional iterative *AES* architecture. As can be observed from Table 3.3, the conventional *AES* area is 13599.04μm^2 whereas the proposed secure scan architecture has an area of 13808.00μm^2.

3.3 Securing Scan through Test Data Encryption

Another way to secure the scan *DfT* architecture is to use encryption of test data on-chip. In test data encryption based technique the test data both test vectors and responses are in encrypted form. Hence, the attacker no longer can apply crafted inputs and can not analyse the test responses. Moreover, the test data can be supplied to any third part for in-field test, and scan test can be carried out without compromising the security of the chip. In a recent work, Mathieu et al. uses scan chain content encryption to restrict the attacker to control and observe the test data [188]. An on-chip block-cipher *PRESENT* [31] is used to encrypt the test vector before loading it into the scan chain.

Also, another on-chip *PRESENT* cipher module is used on the scan-out side to encrypt the test response data. This technique is very effective in fending-off all the existing scan-based side-channel attacks. However, the problem with this technique is that the area overhead is prohibitively too high for using in a standalone *AES* crypto chip. In this section we propose a secure scan architecture based on the principle of test data encryption used in [188]. The proposed technique provides same level of security at comparatively very less area overhead. The major contribution of our architecture is area efficiency and its security features without hampering test, diagnose, and debug capability of the original scan chain.

3.3.1 Securing Scan through Test Vector Encryption

The main idea of the proposed technique is to use an on-chip lightweight block cipher to decrypt the encrypted test data provided by the user or Automatic Test Equipment (*ATE*). The decrypted test data is then loaded into the *AES* scan chain and applied. A high level schematic diagram of the proposed technique is shown in Figure 3.10. Note that the round logic circuitry is not shown in the schematic diagram for the purpose of simplicity. As can be seen from Figure 3.10, a light weight block cipher *PRESENT* [31] is added before the scan-in port of *AES*. Also a n-bit test key matching logic is embedded in the original *AES* core. The test key matching logic checks for a n-bit test key which is embedded in every test vector. If the test key matches it allows to scan out the test response else it masks the *SO* pin. To carry out scan test with the proposed technique following steps are followed:

1. Generate test vectors as well as the corresponding fault-free circuit test responses;
2. Embed a fix n-bit scan shift key (*SK*) in every test vector;
3. Encrypt the test vector off-chip with *PRESENT* cipher algorithm using the same test encryption key (*EK* which is embedded on-chip to decrypt the test vectors;
4. Scan-in the encrypted test vector, which get decrypted on-the-fly by the on-chip *PRESENT* cipher embedded on-chip;

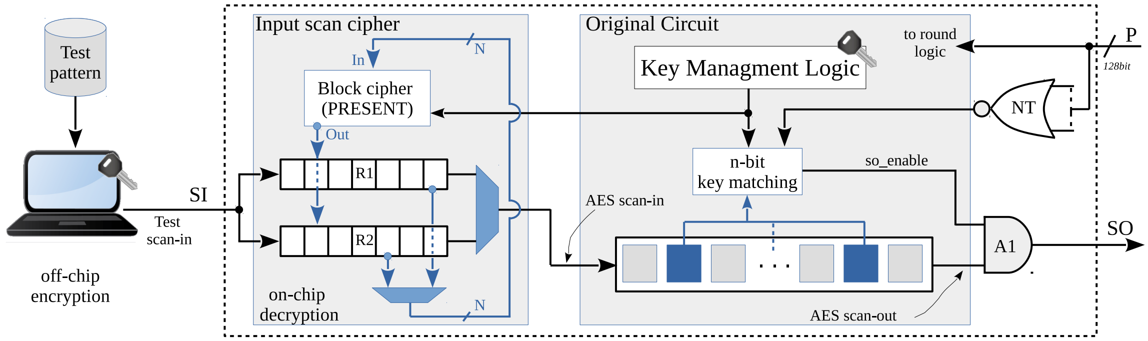


Figure 3.10: Schematic design of the proposed technique

5. Collect the test response and compare with the golden circuit response;

We assume that the test encryption key (EK) and the scan shift key (SK) is managed by the AES key management logic which is already embedded in the hardware. We further assume that the unspecified bits (X or *don't-care* bits) of the test vectors are filled before encryption by the person who manages the system level security features. Also, the scan shift key SK is embedded in every test vector before encryption of the test vectors. After encryption, the test vectors are loaded serially on the chip. The test vectors are decrypted on-the-fly by the $PRESENT$ cipher as they are being loaded on the chip. The decrypted test vectors are then serially shifted in the AES scan chain. The test application procedure is explained in detail in the following subsection.

3.3.2 Test application

The $PRESENT$ block-cipher has a very low cost implementation and it uses 32 clock cycles to encrypt/decrypt 64-bit data blocks. As can be seen from Figure 3.10, in the proposed implementation the $PRESENT$ cipher has two 64-bit round registers $R1$ and $R2$. Both the round registers have serial as well as parallel load capability. The purpose of using two round registers is to decrypt the encrypted test vector on-the-fly while they are being shifted-in and avoid any test time penalty. To apply the test, the circuit is switched from functional mode to test mode by controlling the test enable signal. Since the $PRESENT$ cipher block size is 64-bit, the test vector bits are shifted in a group of

64-bit. However, as the implementation has two round registers, while one round register is being used in encryption process, the other round register is serially loaded with next 64-bit of the test vector. The shifting of test data alternatively in $R1$ and $R2$ makes it possible to decrypt the test vector on-the-fly without any test time penalty. Note that the *PRESENT* cipher remains inactive during functional mode and become functional as soon as the circuit is switched into test mode.

After power-on, at the first instance when the circuit is switched from functional to test mode, by default $R1$ is in parallel load mode (i.e., round register for encryption) and $R2$ is in serial shift mode (i.e., for serial loading of test vector). While the first 64 bits of test data is shifted in $R2$, the garbage data available in $R1$ is getting encrypted during this time. Since it takes the *PRESENT* cipher 32 clock cycles to encrypt the data it waits for next 32 cycles. After the first shift slot the functionality of $R1$ and $R2$ is swapped. So, $R2$ is now in parallel load or encryption mode and the test data loaded into $R2$ during the first shift-in slot will get encrypted during the second shift-in slot. Now, as $R1$ is in serial shift mode, next 64 bits of test vector will be loaded into it during the second shift-in slot. In the third shift-in slot, again the role of $R1$ and $R2$ will be swapped. Also, it is important to note that the scan-in port of the *AES* scan chain remains masked during the first shift-in slot. During the second shift-in slot the AES scan chain is feed by the $R1$ register. During the next shift-in slot the *AES* scan chain will get its input from $R2$, then by $R1$ in the next shift-in slot and so on. In this way, the encrypted test vectors are loaded in the *AES* scan chain only after being decrypted by the on-chip *PRESENT* cipher. The sequence of operations during test application is controlled by a test controller.

Initially, there will be latency of 128 cycles at the start of the scan test process because the first set of 64-bit valid test data will be available only after the end of the second shift-in slot. On the output side of the *AES* scan chain the scan-out pin *SO* by default remains masked until the first capture cycle. *This make sure that the intermediate encryption data from the last functional round, stored in the AES round register, can not be observed by the attacker.* Without being able to observe the intermediate encryption data from the last functional state, the attacker no longer can mount a scan based attack.

Once the test is loaded and applied, the circuit is switched back to functional mode to capture the test response. The response is captured in the scan chain by applying a functional clock. Now to shift-out the test response the circuit is switched back to test mode and shift operation is performed. However, to observe the test response at the SO pin must be unmasked. The masking of SO pin is done based on two conditions. First, the scan test key SK embedded in the test vector must match which is evaluated during the capture cycle. Second, during capture cycle, all the plain-text inputs must be held at a constant logic 0 level. If all the plain text inputs are not held to 0 the SO port will not be unmasked even if the scan key is matched. The plain text constraint is ensured using a 128 bit NOR tree NT shown in Figure 3.10. To unmask SO both the scan key matching and plain text inputs constraints must be satisfied. The constraint on plain text inputs during capture cycles is necessary because once the SO is unmasked the attacker can remain in functional mode and apply any crafted input at the plain text inputs and capture the response. Once the response for the crafted input is captured the circuit can be switched back to test mode and the response can be unloaded. However, the constraints on the plain text inputs makes sure that the attacker can not apply random values at the plain text inputs and observe the corresponding response by unloading. The scan key matching constraint on the other hand ensures that the attacker can not apply random test patterns and observe the corresponding responses.

There are two possible ways to supply the scan key to the key matching logic. One way is to use the round register scan cells itself to pass the scan key SK to the key matching logic. In that case the scan key SK is embedded in the test vector by exploiting the large number of unspecified test bits available in a test vector. It is reported in literature that the amount of *don't-care* bits could be as large as 90% in the test pattern set [115]. Depending upon the size of the scan key the fault coverage could drop because of the conflict between scan key bits and specified test key bits in some of the vectors. To avoid that another possible way is to add extra dummy scan cells in the scan chain. At the end of the test vector loading these dummy scan cells will be loaded with the scan key

SK and will be used for evaluation during capture cycle. The use of dummy scan cells will not have any impact on the test coverage. However, depending upon the size of the scan key there will be some increase in area overhead. The test time may also increase in some cases depending upon the length of the original scan chain. The effect of dummy scan cells on test time is explained in detail in the next section.

3.3.3 Security and Testability Analysis

Security

We assume the attacker model where the attacker is an insider and have access to all the design and security features related information. We further make the assumption that the attacker can generate the original test vector for the circuit-under-test. However, original test vector set can be manipulated by random filling of dont care bits. On an average the number of unspecified bits in compressed test patterns are 80%-90%[115]. There are many *X-filling* techniques which can be used to fill the *dont-care* bits. The *X-filling* technique can be exploited to change the test vector set significantly from the initial test vector set before encryption. The person who is in-charge of the system level security features can use any random *X-filling* technique and embed the security key and then can encrypt the test vector set off-chip with the *PRESENT* cipher with the same encryption key which is embedded on-chip.

Since the attacker does not know the *PRESENT* cipher key *EK* he can not apply desired inputs. Hence the attacker no longer can mount control based attacks. Also, this makes it impossible for the attacker to embed the scan key *SK* in the test vector in such a way that after decryption during the shift-in process, it matches with the scan key stored on-chip. Even if the scan key is public and known to the attacker, still the attacker would not be able to generate the correct encrypted test vector with embedded scan key. The scan key size can be as small as five to ten bits because to embed the scan key bits at proper bit position in every test vector the attacker needs to break the *PRESENT* cipher which have a very high security level [188]. The masking of scan out pin does not allow the attacker to mount observe based attack. In the proposed implementation we have used five dummy scan-cells for loading a scan key of size five. In terms of area, it will result in negligible area overhead.

Testability

The proposed technique allows to exercise both the stuck-at as well as the delay fault test without any loss in fault coverage. Also, since the test vectors are encrypted, they can be supplied to any third party for in-field test without revealing the secret information. In terms of test time, the proposed technique has a minimum 128 cycle overhead. Since the *PRESENT* cipher has a block size of 64, then if the scan chain length is not a integral multiple of 64 there will be a test time overhead per test vector which will be equal to $t_{extra} = 64 - m$ for $l = 64 \times n + m$. Here, l is the length of the scan chain, n is an integer and m is remainder. In all such cases, a maximum of m number of dummy scan-cells can be used without causing any extra penalty on test time. In case the value of m is 0, i.e., the scan chain length is an integer multiple of 64, there will not be any test time overhead per test vector if round register scan cells are used to embed the scan shift key. In case of dummy scan cell, the test time overhead per test vector will be $t_{extra} = d$ clock cycles, where d is the number of dummy scan cells. It should be noted that the *PRESENT* cipher logic can not be included in the scan chain because that will jeopardize the whole security feature. The *PRESENT* logic must be tested through functional random patterns. It is shown in [188] that the *PRESENT* cipher logic can be tested with 100% stuck-at fault coverage by applying 1000 random functional patterns with a pattern size of 64 bits. Furthermore, the proposed secure scan design technique does not hamper the diagnosis capability as the test response data is directly observable at the primary scan-out pin if the test vector encryption scheme is known to the user.

Design cost

To estimate the design overhead we synthesized the proposed secure scan design along with the conventional scan design, and scan chain encryption technique proposed by

Table 3.4: Area overhead comparison

Iterative Architecture			
Design →	*Conventional scan design*	*Scan chain encryption* [188]	*Proposed secure scan*
Area (um^2)	22397.12	30488.96	26457.92
Overhead (%)	−−	36.12%	18.13%
Pipelined Architecture			
Overhead (%)	−−	2.92%	1.41%

Mathieu et al. [188]. The synthesis has been carried out for both Iterative as well as fully pipelined architecture based implementations. The synthesis has been done using Synopsys Design Compiler at *UMCs* 65*nm* low leakage process technology node. The area numbers are reported in Table 3.4 for all the three designs. As compared to the conventional scan design the proposed secure scan design has an area overhead of 18.13% in case of iterative implementation and 1.41% in case of pipelined implementation. On the other hand the area overhead of scan chain encryption technique proposed in [188] is 36.12% for iterative implementation and 2.92% for pipelined implementation, which is prohibitively too high to use in a standalone iterative *AES* design.

3.3.4 Securing Scan through Test Response Encryption

Our test vector encryption based technique which is explained in the previous section is very effective in fending off all the known scan attacks. However, the design cost is a bit higher. Also, the *PRESENT* cipher which is embedded on chip for test vector encryption can not be used during functional mode operation and remains idle. In this section we purpose another test data encryption technique to secure the scan design. Instead of using *PRESENT* cipher we use *AES* itself to encrypt the test responses. The proposed technique uses a two stage pipelined *AES* architecture for securing the scan test operation. Moreover, the proposed pipelined architecture function in a pipelined

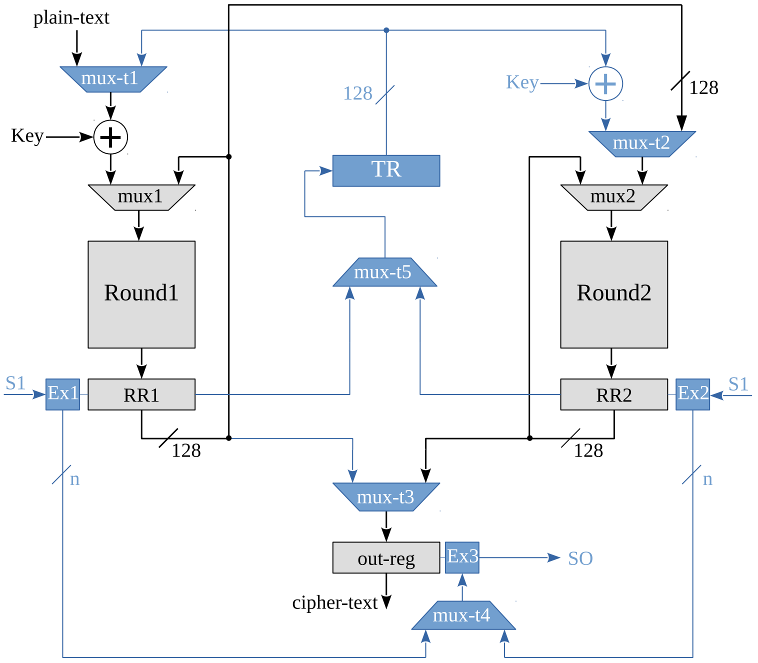


Figure 3.11: Schematic diagram of the proposed secure scan design

fashion during functional mode of operation. As a result the throughput increases by a factor of 2X compared to the standard iterative *AES* architecture.

The schematic design of the proposed architecture is shown in Figure 3.11. The architecture has two *AES* modules. The first module is formed by round multiplexer *mux1*, round logic *Round1*, and round register *RR1*. The second module is formed by round multiplexer $mux2$, round logic $Round2$, and round register $RR2$. In addition to that there are extra logic which is for test purpose only. The test purpose only circuitry consists test multiplexer *mux-t1* to mux-$t5$, register $Ex3$, and test register *TR*. The mux-$t1$, mux-$t2$, and mux-$t3$ are 128bit input 2×1 multiplexer whereas mux-$t4$ is an n-bit variable size 2×1 multiplexer, and mux-$t5$ is a 1bit 2×1 multiplexer. The test register *TR* is a 128bit serial-in and parallel-out register.

3.3.5 Functional and Test Mode Operation

During functional mode of operation both the *AES* modules functional in a pipelined manner. However, during test mode of operation, while one module is being tested the second module is used to encrypt the test response of the first module and vice-versa. The operational details of the proposed architecture in both modes of operation are explained in the following subsections.

Functional mode operation

The proposed secure architecture operates as a two stage pipelined *AES* architecture in functional or mission mode of operation. The data path line in black colour shows the functional data path. During functional operation, plain-text is always selected at *mux-t1*. After being bitwise *XORed* with the encryption key (also called pre-round step), the transformed data is passed to the round logic *Round1*. The first rounds of encryption are performed by the first *AES* module. At the end of the fifth encryption round the intermediate encryption data stored in round register *RR1* is passed to the second *AES* module through $mux\text{-}t2$ and $mux2$. The next five rounds of encryption are performed by round logic $RR2$. At the end of tenth encryption round the encrypted data or cipher-text is written into output register *out-reg*. During the last five rounds of encryption, the first *AES* module partially encrypts next plain-text input. So, after an initial latency of ten cycles, we get 128bit of cipher text in every five cycles. Hence, the throughput of the proposed architecture is almost two times than of the standard iterative *AES* architecture. It should be noted that during functional mode of operation the output cipher register *out-reg* is always written by the second *AES* module.

Test mode operation

To perform the scan test the architecture operates in a specific manner. Let us suppose that first the first *AES* module is being tested. The *CUT* is switched to test mode by pulling the scan enable signal *SE* to logic high level. Note that two scan chains are formed during scan or test mode. The first scan chain is formed by the extra register and the round register of the *AES* module being tested. The second scan chain is always formed by the cipher-text output register out-reg followed by the extra register $Ex3$. Also the fist scan chain is serially connected to the test register *TR* via test multiplexer *mux-t*5. It should be noted that the first scan chain is isolated from the second scan chain and it can not be accessed via the scan-out port *SO*. Only the second scan chain is observable through the *SO* pin. As assumed earlier the *AES* module is in scan mode and *AES* module two is in response encryption mode. So the first scan chain is formed by *Ex1* and round register *RR1*. The *Ex1* register is formed by the round controller of the first *AES* module. The size of the *Ex1* should be at least nine bit (9bit). To fulfill this condition dummy scan cells can be inserted in *Ex*3. This restriction will become evident as we explain the test mode operation. Let us assume that the length of the first scan chain $137 = 128 + 9$bit (128bits of *RR1* and 9bits of *Ex1*).

Now, the test vector is shifted in by consecutive application clock pulse. As the test vector serially shifts into the round register *RR1* the intermediate encryption data stored in it from the last functional cycle shifts into the test register *TR* through *mux-t*5. Since the size of *RR1* is 128bit at the end of the first 128 shift cycles the last functional round data in *RR1* completely propagates into *TR*. At the end of 128th shift cycle the data gets bitwise *XORed* with the encryption key and propagates into round logic of second *AES* module. Now, the second *AES* module runs for next nine cycles in encryption mode. During these nine cycles the test vector is completely shifted into the scan chain and gets applied. Now to capture the test response the circuit is switched back to functional mode and one clock pulse is applied. This captures back the test response into first scan chain. At the same time the response captured into *Ex1* will be copied to *Ex3* of the second scan chain via test *mux-t*4. During the capture cycle the tenth round of encryption is performed by the second *AES* module and encrypted data is written into the output register *out-reg*. The encrypted data written into *out-reg* is from the last function cycle data stored into *RR1*.

Now to load the second test vector the circuit is again switched back into test mode. While the second test vector is being shifted in the encrypted test data in the *out-reg* is also being shifted out. Note that as the second test vector is being shifted and applied the response of first test vector vector is encrypted and written into output register *out-reg*. So, the shift-in operation of the third test vector will also perform two other operations simultaneously. First, the test response of the second test vector will shift into *TR* and will be encrypted, and second the encrypted test response of the first test vector available in the out-reg will be shifted-out. This means that when the current test vector response is captured in the first scan chain the encrypted test response of the previous test vector is captured in to the *out-reg*. All the test vector for the first *AES* module can be exercised in a similar way and corresponding encrypted test responses can be observed through *SO* pin. Now to test the second *AES* module the role of the two *AES* modules are swapped. The second *AES* module will be tested and first *AES* module will be used to encrypt the response of the first. During the test of second *AES* module the first scan chain will be formed by the extra register $Ex2$ and the round register $RR2$. As noted earlier, the second scan chain will always be formed by *out-reg* and $Ex3$.

The test process is controlled by a test controller. The test controller also ensures that once the role of the two *AES* modules is decided at the start of the scan shit process it can be changed only after the capture operation. This condition is very critical from the security point of view. It ensures that the cipher-text output register only gets written by the *AES* module which is presently in response encryption mode. That means, when *AES1* is being tested out-reg is written only by $AES2$ and vice-versa. The security and testability of the proposed architecture is evaluated in the following subsection.

3.3.6 Security, Testability, and Design cost Analysis

Security analysis

The proposed secure scan architecture is immune to all the known scan-based side-channel attacks. The attacker does not have any access to the intermediate encrypted data from the round register hence there is no possibility of mounting a differential scan attack. The test responses are available only in encrypted form through the cipher-text register. The proposed technique actually isolates the round register from the output scan chain which is the second scan chain. The first scan chain which have the intermediate encryption data are not accessible through the scan-out pin *SO*. Hence, the proposed technique is immune to all the scan attacks.

Testability analysis

In terms of testability, the proposed architecture does not put any restriction on the type of scan test that can be applied. All the conventional scan based test can be applied. The scan design insertion can be done by minor modification in the scan synthesis process. The test controller can not be tested by scan test, however it can be tested indirectly using functional test sequences.

Design cost

In terms of area, the proposed design have almost 73.31% area overhead if the logic sharing is not allowed among the two *AES* modules. However, with logic sharing between the two *AES* modules, the area overhead can be reduced. The proposed architecture offers a secure way to scan test the *AES* chip and offers a throughput which is higher than the iterative architecture by a factor of $2X$. The proposed secure scan architecture allows to use the test encryption hardware during functional mode of operation. Further, the pipelined architecture can be tweaked to for reliability purpose. The two *AES* modules can be configured to operate in parallel and the cipher-text can be compared at the end of the encryption operation to detect any occurrence of a transient error.

3.4 Conclusion

The scan *DfT* architecture is a big security issue for cryptographic chips which needs urgent attention. The scan design needs to be secured without compromising on its testability aspects. We have proposed a set of techniques to secure the scan design. The proposed techniques also preserves the test capability of conventional scan design. The proposed technique are based on test protocol countermeasure namely encryption key masking, test restriction and test data encryption. The first proposed technique which is based on encryption key masking principle. It keeps the encryption key masked throughout the entire test process and allows to exercise all types of conventional test including delay test. We proposed another technique based on test authorization mechanism. The user needs to supply a valid one-time test authorization key at the start of the test session. Once the user is authorized, the conventional test sequence can be started. We also suggested two ways to implement the test authorization logic: *LFSR* based, and *MISR* based authorization. There is no test time overhead for the proposed technique. Also, compared with existing secure scan designs, the proposed technique incur least area overhead. Moreover, all kinds of conventional scan based tests can be exercised using the proposed secure scan design.

Also, we have suggested a scheme to carry out the scan test in a secure manner. The proposed scheme is based upon plain-text restriction during test mode. It has the advantage that it neither uses key masking nor it has any scan key based test authorization. Hence, is extra test key to manage. The proposed solution does not have any impact on the test efficiency, fault diagnosis, and post-silicon debug capabilities.

Finally, we come up with two schemes to secure the scan design through test data encryption. In first scheme, we uses a low cost on-chip encryption engine to encrypt the test vectors. It does not allow the attacker to apply crafted test vectors. In second scheme we use a pipelined *AES* architecture to encrypt the test responses. In this scheme the user only gets encrypted test responses and hence can not analyzes the test response to retrieve the encryption key. The proposed scheme can be extended to secure the fully pipelined *AES* architecture against scan attacks.

— ∗ — ∗ —

Chapter 4

Joint Scan Test Architecture

Test time, test data volume, and test power have been major concerns in serial scan based manufacturing test. The problems have been associated with serial scan architecture since the beginning. For the contemporary *SoC* designs which are complex and having billions of transistors, the test time and test data volume are escalating almost exponential rate. The test time and test data volume directly affect the product cost. Moreover, the test power is adding up on the manufacturing cost primarily due to yield loss. Therefore, these three parameters have been considered as a mandatory design optimization point for general as well as special purpose designs. So far, several research have been carried out to mitigate these problems which are inherently associated with the standard serial scan architecture. Apart from the serial scan architecture there have been a research on another class of scan architecture called Random Access Scan (*RAS*). The *RAS* architecture is proven to be extremely good at reducing test power. However, some of the drawbacks of *RAS* makes it difficult for its industry adoption.

In this chapter, we propose a composite scan architecture which aims to combine both, the serial scan and random access scan, to harness the best out of each. The proposed architecture minimizes test time, data volume, and test power altogather. The architecture is designed by hybridizing the standard serial scan and random access scan architecture. The principle is to harness the advantage of test time and test data volume from serial scan architecture and advantage of test power from *RAS*. The test time and test power will be optimized by joint configuration of serial scan and *RAS* to get the best possible results. The effectiveness of the architecture is experimentally demonstrated on the modified (enlarged *SoC* designs) *ISCAS89* circuits. The architecture is compared with the standard serial scan, the random access scan, and with the four-modes *Joint-scan*.

4.1 Introduction

The serial scan architecture is one of the widely used test mechanism for most of the modern designs. Although scan *DFT* provides a set of advantages yet it is challenged by high test power, increase in test data volume for complex *SoCs*, and gradual hike in test cost due to large test time [103]. It has been observed that these parameters follow Moore's trend [170].

Several methodologies have been proposed to solve the above stated problems [33, 43, 79, 153]. However the continuing technology scaling have been demanding for an efficient methodology for above stated problems [103]. Compression and response compaction based techniques have been proposed to solve the test data volume and test time issues [159, 186]. Currently these techniques are being used widely in industry. Despite the reduction provided by these techniques, further reduction in test data volume for current *SoCs* is necessary. Further, these technique are often challenged with pin count limitation [41]. Several partial scan techniques have been proposed to curb the scan chain length to minimize test time. *ITRS* projection show that the test cost will remain as a dominating factor in overall product cost in coming years [103].

The another severe issue that scan test is facing is test power dissipation. The test power has direct impact on yield and reliability. The excessive peak power causes yield loss whereas the sustained average power causes chip burnout and also has impact on long term reliability [80]. Several methodology, viz scan flip-flop gating, pattern *X-filling*, scan chain and pattern reordering, test scheduling for *SoCs*, scan chain masking, power aware scan flip-flop design, have been proposed in recent past [80]. Some of these techniques have reduced the power to a certain limit. However, the ever increasing design complexity demands for high activity and compact test pattern set in order to curb the menace of test data volume and to attain an acceptable fault coverage. This further increases the test power dissipation. Furthermore, the power density in modern *3D SIC* has been shown to be a major problem which requires new test methodology [121]. Hence, there is a dire need of an efficient low power test methodology which is also scalable with the design complexity.

An alternative to serial scan *DFT* is the random access scan (*RAS*), proposed back in 1980. The *RAS* architecture is recently proven to be very much efficient for test power, test data volume and test time compared to standard serial scan design [20]. In our recent work we have proposed a new *Joint Scan* architecture, called *JScan* [210], which further reduces these parameters.

4.1.1 Outline of Problem and Contributions

In this work, we propose a new scan *DfT* architecture which integrates the serial and random access scan architecture to harness the best of each. We name it as *2M-JScan* because it functions in just two modes of operation unlike the earlier implementation of Joint-scan which operates in four modes. Experimental results show improvement, compared to the previously proposed *Joint-scan* architecture, on test time without affecting data volume and test power. Furthermore, we also have proposed a new scan cell design that can be used as a common scan cell in *2M-JScan* architecture wherein it can be used both as a serial scan cell as well as a Random Access Scan (*RAS*) cell. The *Joint-scan* test architecture has been implemented using the proposed scan flip-flop. The experimental results show a promising reduction in interconnect wire length. The reduced interconnect wire length could help in overcoming the routing congestion which impedes practical implementation of *RAS* architecture. Contributions of this Chapter could be summarized as follows:

1. Proposes the *2M-JScan* architecture

2. Develops an efficient test control mechanism
3. Reduces test pin count and minimizes test time
4. Provides design of a new scan cell for *Joint-scan* architecture

Rest of the chapter is organized as follows. Related works and background on random access scan are explained in Section 4.2. The proposed architecture with test control mechanism is elaborated in Section 4.3. Experimental results are discussed in Section 4.4. Section 4.5 explains the design of a new scan flip-flop for *Joint-scan* architecture. This Section further elaborates on the details of test application process and post-layout timing results of the proposed scan cell. Section 4.6 explores implementation of Joint-scan architecture using the proposed scan flip-flop along with the experimental results. Section 4.7 concludes this chapter.

4.2 Related Work, Background and Motivation

In this section we will look at some of the earlier work on random access scan. Some key features and functionality will be explained with respect to test power, test data volume, and test time problems.

4.2.1 The Standard Serial Scan Architecture

Serial scan have been a standard practice as an industrial *DFT* architecture. In academia as well, a large body of research in *VLSI* test has been devoted to various issues in serial scan architecture. The current practice in industry is to deploy a multiple parallel scan architecture with de-compressor and compactor at the input and output of scan chains respectively. The design of *DFT* engine by different industry varies with the specific problem of interest at the first place [41, 108, 142]. Similarly, researchers from academia also have proposed several interesting architectures targeted towards a particular research issue. At the core of serial scan architecture there lies inherent issues which are sustainable cause of worry for the test engineers and researchers. As discussed in the previous section, the contemporary problems are test power, test time, and data volume. As projected that the future designs will be more complex with respect to the functionality and number of transistors on a single chip, these parameters are bound to scale to the limit of worry [103, 109]. Several approaches have been proposed so far to mitigate the problem of test power, test time, and data volume. We will highlight some of the approaches which are based on architecture/design and discuss their limitations and challenges.

Among the most widely used serial scan architectures is the multiple parallel scan chains [76]. The architecture is proven to be efficient in reducing test time since each of the scan chain length is kept under a limit by increasing the number of chains. However just plain parallel scan chains does not seems to scale well as the complexity of the chip increase. The I/O pin counts become a limiting factor for such architecture. Therefore, the innovation led to test de-compressor and response compactor ($MISR$ etc) based scan architecture design. There have been class of serial scan architecture which are based upon this technique: *Embedded Deterministic Test* (EDT) [159], *DFTMax Compression Architecture* [45, 108], Scan Tree [141]. The primary objective of these methodologies is to minimize the test time and test data volume as efficiently as possible without any toll on fault coverage. Most of the industry currently uses the architecture from this class. Another class of serial scan architecture uses broadcast based input test data compression/decompression and the output compactor mechanism to restrict the I/O pins. Scan tree [24, 141], scan forest [239], Illinois scan [63, 95] are some of the well researched serial scan architectures. These architectures provide a reasonable solutions to test time and test data volume issues. Many of the architectures are being augmented with low power features [44, 72, 161].

Many research foresight that the problem of test power, data volume, and test time will scale up as the design gets more complex and dense [103, 235]. In this work we address these challenges with a long term vision. There have been an alternative approach to the serial scan chain called Random Access Scan (RAS) which provides a promising solution to the test power problem along with test time and data volume. We discuss some of the random access scan architectures in the following section.

4.2.2 The Random Access Scan Architecture

The random access scan architecture was proposed by Ando [17] in 1980. Thereafter, Wagner [217] in 1983 and Ito [101] in 1991 have implemented the *RAS* architecture to evaluate its feasibility. However due to excessive hardware overhead, added by the *DFT* logic and routing congestion, the architecture remained as a future hope. The architecture was not cost effective at that time primarily due to area and routing congestion overhead.

Recently, Baik et al. [20, 21, 23] have explored the feasibility of *RAS* architecture in the context of contemporary technology node at 180nm and 65nm. In the earlier implementations one of the limitations was the address decoder, due to which the routing of global address lines were creating excessive congestion and area overhead. In [21] the single address decoder is split into row and column address decoders. The scan flip-flops are now connected by the row and column address lines in a two dimensional grid. The experiment show that this modification reduces the area and congestion compared to the older architecture.

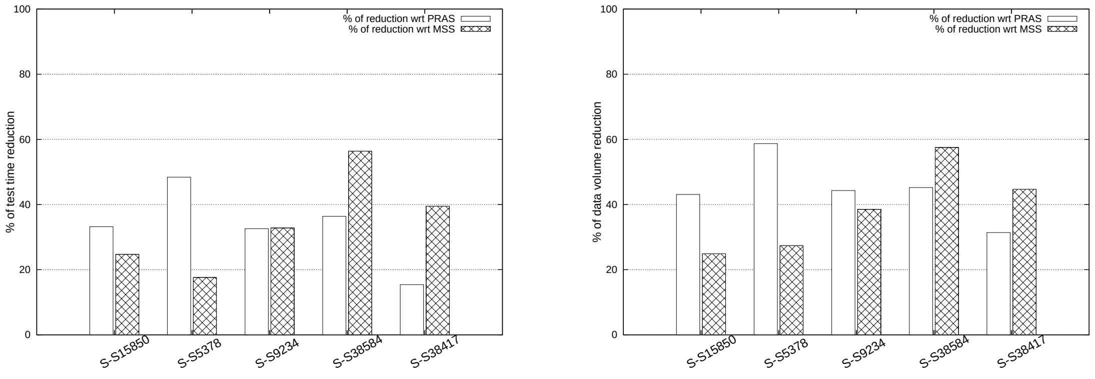
The routing wire length is further minimized by Mudlapura et al. [143] using T flip-flop (*TFF*) based scan cell. The proposed method eliminates the scan-in and scan-enable lines. However an additional gate delay is introduced in the clock path. To eliminate the clock gating, a modified *TFF* based scan cell is proposed by Adiga et al. [4]. A cluster based grouping of scan cells to minimize the routing congestion is proposed by Hu et al. [96]. A methodology to hybridize serial scan and *RAS*, called as *WOR-BIST*, is reported by Yao et al. [51] to reduce area overhead in *BIST* environment.

Test time and data volume problems in random access scan architecture are addressed by Baik et al. [22, 23] and Adiga et al. [3]. Adiga et al. used an additional data buffer for buffering test data while shifting the address. Baik et al. proposed a progressive random access scan (*PRAS*) in which a row enable shift register is used to address each row one at a time and parallel reading of response and hence minimized the test time. In our proposed architecture the *PRAS* is used along side the standard multiple sequential scan. *RAS* architecture is highly power efficient compared to serial scan[23]. In *RAS*, only one flip-flop is loaded with test stimuli at a time without creating any toggles in other flip-flops. The earlier works report $80 - 90\%$ of power reduction in comparison to the standard serial scan architecture [23].

4.2.3 The Joint-scan Architecture

For the future large scale designs the *DFT* solutions are required to be scalable. The *DFT* architecture must be capable of reducing the test time, test data volume, and test power proportionately for the larger designs as the design gets more complex. The *Joint-scan* architecture aims at that problem [210]. We have proposed for the first time the *Joint-scan* architecture to carry out a foundational study for its feasibility and to compare its advantage over the existing multiple serial scan architecture and random access scan architecture. We call this architecture as *2M-JScan* which stands for *two-mode Joint-scan* architecture. To avoid ambiguity we make the nomenclature distinct for the earlier proposed *Joint-scan* architecture as *4M-JScan* and the current *Joint-scan* as *2M-JScan*, where the *Joint-scan* refers to this class of *DFT* architecture. Our first study reveals a promising results along with a couple of challenges. We reproduce some of the results here in Figure 4.1(a) and Figure 4.1(b) for test time and test data volume respectively. The results show the percentage of reduction obtained due to *4M-JScan* architecture over progressive random access scan (*PRAS*) and multiple sequential scan (*MSS*) [210].

In the *Joint-scan*, for both *4M-JScan* and *2M-JScan*, one of the challenges is to design a scan flip-flop which could be used in *Joint-scan* architectures without much overhead in the existing synthesis process, particularly when the architecture is implemented with *PRAS* and *MSS*. To address this problem, we have proposed a special scan cell which can be used both as a serial scan cell as well as a *RAS* cell [10]. Moreover, the proposed scan cell eliminates the scan multiplexer off the functional path and hence offers better timing performance compared with conventional scan cell. The proposed scan cell could



(a) Test time reduction

(b) Data volume reduction

Figure 4.1: Test time and data volume reduction by *4M-JScan* architecture [207]

be used as a conventional scan flip-flop in contemporary serial scan architecture.

The second issue that *JScan* faces is the interconnect congestion which is introduced by the *RAS* part. We synthesized the *JScan* architecture using the proposed scan cell. The experimental results shows significant reduction in average and total interconnect wire length [14]. The proposed scan cell is explained in detail in Section 4.5. We convey here a note that the challenges may vary as any other suitable architectures are employed in place of *PRAS* and *MSS*. In this chapter we present the entire frame work for configurable *Joint-scan* architecture along with the new proposed *2M-JScan* architecture.

4.3 *2M-JScan* Architecture

The proposed architecture integrates the serial and random access scan architecture to harness the best of each. Serial scan architecture requires minimal circuitry whereas the random access scan is extremely power efficient and up to some extent better than serial scan in test time and test data volume reduction. On the other hand the serial scan dissipates undesirable power and the random access scan occupies comparatively large layout area and introduces routing congestion. The proposed *2M-JScan* architecture simplifies the test control mechanism and further reduces test time keeping the data volume and power same as in the *4M-JScan* [207].

The *2M-JScan* consist of two sub-scan architectures: Partial serial scan (*P-serial*) and Partial random scan (*P-random*). The available flip-flops in *CUT* are segregated into two groups to form *P-serial* and *P-random*. The *P-serial* is a serial scan chain formed from the first group of flip-flops and the *P-random* is a random access scan formed from the second group of flip-flops. Following sub sections give detail on the architecture.

4.3.1 The Architecture

The three primary components in the proposed architecture are *P-serial*, *P-random*, and test control logic (*TCL*). The *P-serial* and *P-random* are implemented with multiple serial scan chains (*MSS*) and progressive random access scan (*PRAS*) [23] respectively. To realize the proposed architecture we have identified three main challenges:

1. Integrating and operating the *P-serial* and *P-random*
2. Maintaining equilibrium in shift time across all patterns
3. Grouping of flip-flops in *P-serial* and *P-random* to obtain the best results

The proposed architecture is shown in Fig. 4.2. The test control logic ensures correct operation of the architecture. The challenge in the part of test control logic is to operate both the architectures concurrently for load/unload of test/response, launch, and capture operations. The *P-serial* consists of *scan-in* lines/pins to shift in the test stimuli and *MISR* at the output to compact the response and a *scan-out* line to shift out the final response. The test control logic controls the activities in *P-serial* using serial scan enable (*SSE*) signal. When the architecture is implemented with our proposed scan flip-flop shown in Figure 4.10, the *SSE* signal is connected with the *SCK* test clock input of the flip-flop. The detail control lines of course are an implementation issues and may vary for different implementation, our primary goal here is to establish the architectural challenges, and, within a scope, the implementation issues. The *P-random* which in

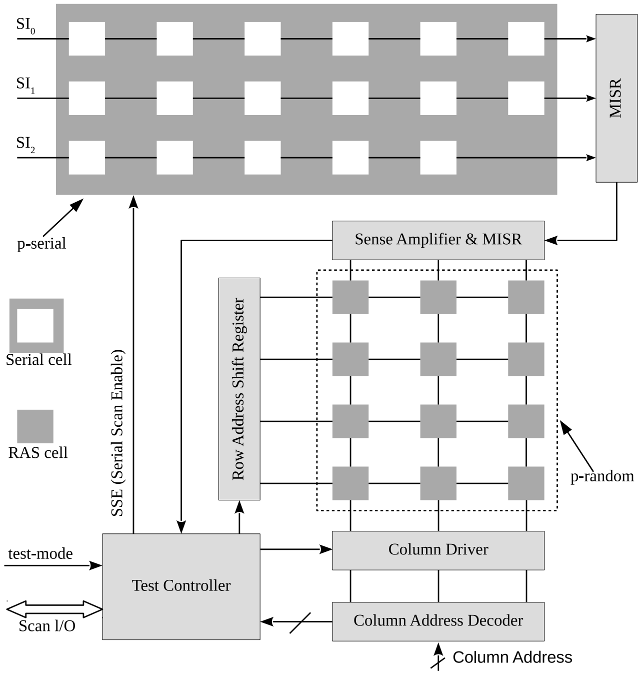


Figure 4.2: Proposed two-mode Joint-scan Architecture (2M-JScan)

this work is *PRAS* consists of three major units: row address register, column address decoder, and *MISR*. It also uses a sense amplifier to read responses. The scan flip-flops are arranged in grid fashion, and they are connected with row and column enable lines. The column enable lines are driven by column driver and the row enable lines are driven by row address shift register. The responses in *P-random* are compacted using *MISR*. The *MISR* is connected with test control logic through scan out line [23].

The test control logic in this architecture is simplified to operate with only one test mode line/pin unlike the *4M-JScan* where two test mode pins were required. The serial scan enable signal is generated using test control logic. The test control logic also gets input from column address decoder. These inputs are generated from reserved address bits dedicated for generating control signals. Test control logic steers overall test and functional mode operations.

4.3.2 Limitations of *4M-JScan*

In the *4M-JScan* (four-modes *JScan*), pattern loading/unloading time varies for *P-serial* and *P-random.* The variation arises due to the fact that in *P-random* number of write operations vary from a test stimuli to another test stimuli. The write of test stimuli bits in *P-random* is based on two criteria: 1.) the stimuli must be specified bit (the don't care bits in stimuli are not considered for write), 2.) the stimuli bit must differ from the don't bit of last response. This leads to variation in load/unload time. To control this variation the test control logic for *JScan* needs two test mode pins. Furthermore, additional circuitry is required to keep track of completion of stimuli/response loading/unloading in *P-serial* and *P-random.* Following example demonstrates the scenario.

Example 4.1. Let us consider an example configuration where *P-Serial* has two parallel scan chains. The first scan chain consists two scan flops: sf_0 and sf_1; the second scan chain also consists two scan flops: sf_4 and sf_6. The *P-random* has two rows; the first row consists three scan flops: sf_2, sf_3, and sf_5; the second row consists two scan flops: sf_7 and sf_8. Also, the *P-random* has three columns. As can be seen from Table 4.1, the *P-Serial* for loading S_0 takes 2 cycles where as *P-random* takes 4 cycles which results into a variation of 2 cycles. Similarly, for S_1, it results in variation of 1 cycle. For *P-Serial* it is easy to see that it consumes two cycles to load/unload test/response. For *P-random* it varies for S_0 and S_1, for S_0 it takes two cycles; one cycle to read response and one cycle to write bit 1 to sf_5. Similarly *P-random* takes three cycles for S_1. Therefor, there is a variation in test time across the patterns. To keep track of this variation for each pattern a dedicated controller is employed in *4M-JScan.*

In the *2M-JScan* architecture proposed in this work, we equalize the variation and thereby eliminate the need for additional controller. We have proposed a methodology to align the patterns for *P-Serial* and *P-random* in such a way that the pattern loading time for each pattern turns out to be equal.

Table 4.1: Variation in loading/unloading (l/u) time

Pattern	*P-Serial* (2-chains)	*P-random* (2-row/3-col)	l/u time
	sf_0 sf_1 sf_4 sf_6	sf_2 sf_3 sf_5 sf_7 sf_8	P-s | P-r
S_0	1 x 0 1	x x 1 0 x	2 | 4
R_0	x x 0 0	1 x 0 x x	
S_1	0 x 0 1	x x 1 x x	2 | 3
R_1	1 x x 1	0 0 1 0 x	

4.3.3 Test Pattern Alignment

In this section, we will describe the procedure for padding the patterns such that each of the patterns will be aligned to have equal loading/unloading time. First of all we have to find out the maximum load/unload cycle (this is synonymous to shift time in case of serial scan chain) by any individual pattern. Thereafter the candidate test patterns, which are to be appended, will be selected and padding bits will be computed.

Let $S_i(S_i^s, S_i^r)$ and $R_i(R_i^s, R_i^r)$ be i^{th} stimuli and response pattern respectively, where S_i^s and S_i^r are the stimuli pattern for *P-Serial* and *P-random* respectively and R_i^s and R_i^r are response pattern for *P-Serial* and *P-random* respectively. And, let $T_i = \{S_i, R_i\}$ be the i^{th} test set with stimuli and response pattern. The computation of maximum load/unload cycle can be done as follow:

$$maxcycle = Max_{i=0}^{N-1}(lutime(T_i)) \tag{4.1}$$

where $lutime(T_i)$ is load/unload time for test set T_i, and N be the total number of test set. Here we assume that stimuli loading time ≥ response unloading time for *P-Serial* as well as for *P-random*. Now, the padding bit pattern for each test set, T_i, will be determined using *maxcycle*. We have to note that only the test stimuli needs to be padded with required number of bit patterns. The required number of bit patters are decided based on the number of additional loading/unloading cycles needed to equal the *maxcycle*. For example, if the stimuli S_1^s are to be padded then the number of padding bit pattern needed are in the order of *maxcycle* - *loadtime*(S_1^s). Similarly for stimuli S_1^r the padding bit patterns are in the order of *maxcycle* - *loadtime*(S_1^r).

Appending for *P-Serial* Pattern: For *P-Serial* scan chain the padding bit patterns are appended at the beginning (right most places) of the test stimuli. The number of additional bits appended are equal to the *maxcycle* - *loadtime*(S_j^s), where S_j^s is the j^{th} stimuli. The bit pattern considered for appending would be all 0s (0 0 0 0 . . .) or all 1s (1 1 1 1 . . .). The appending pattern should be chosen such that it does not affect response compacter.

Example 4.2. Let's consider the scenario described in Table 4.1. Let *P-Serial* be two parallel scan chains chain-1: sf_0 and sf_1, and chain-2: sf_4 and sf_6. Considering stimuli S_0 with chain-1: $< 1\ X >$ pattern and chain-2: $< 0\ 1 >$ pattern, the new pattern with padding would be $< 1\ X \mid 0\ 0 >$ to be shifted to chain-1 and $< 0\ 1 \mid 0\ 0 >$ to be shifted to chain-2.

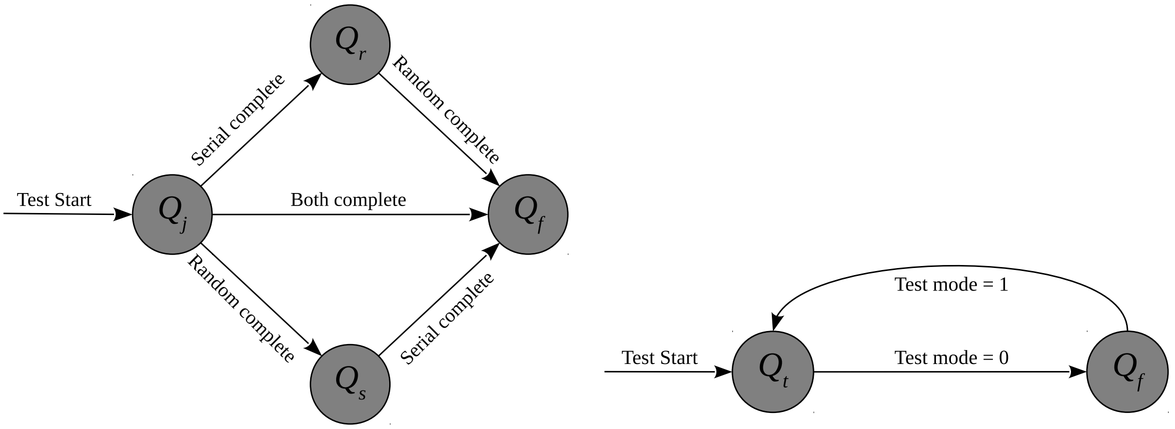
Appending for *P-random* Pattern: For *P-random* the pattern appending procedure differs from that of *P-Serial*. The basic procedure to write a scan flip flop in *P-random* differ. First a row is enabled to read the response, then a column address for the target flip-flop is applied, thereafter the stimuli bit is applied for write. These three steps are essential. Accordingly, the pattern has to be formatted. One of the possible format is $< coladdr,\ bit >$, where *coladdr* is address of target flip-flop in an enabled row and *bit* is stimuli to be written. The pattern for appending has to be in the given format of $< coladdr,\ bit >$. The last column to be written in last row of *P-random* scan grid is chosen as the desired pattern for appending. In case when no column is to be written in last row, the first column is considered as the last column to be written and don't care bit is considered as stimuli *bit* to be written.

Example 4.3. Considering the scenario depicted in Table 4.1, let *P-random* be configured as two rows and three columns. The test set $T_0(S_0^r, R_0^r)$ does not need appending since it's $maxcycle - loadtime(S_0^r)$ is zero. However, $T_1(S_1^r, R_1^r)$ need one pattern appending. The appending pattern in this scenario would be $< 0\ 0\ 0,\ X >$, where [0 0 0] is the address of 0^{th} column and X is the stimuli bit.

In this way all the patterns are aligned by appropriate appending. The implication of alignment process is that it equalizes the load/unload time of the stimuli/response. As stated earlier the pattern alignment results in saving of hardware overhead due to test control logic.

4.3.4 Test Control Mechanism

Test control mechanism in the proposed *2M-JScan* architecture is simplified compared to the controller used in *4M-JScan*. The mode control uses only one test mode pin whereas the *4M-JScan* uses two pins. The state diagram of test control mechanism of *4M-JScan* is shown in Fig. 4.3(a). This saving of one additional pin contribute in minimization of test time. The controller now does not need to keep track of completion of load/unload operation in either *P-Serial* or *P-random* unlike in *4M-JScan*. This reduces additional circuitry. The controller now have only two states: test (Q_t) and functional (Q_f) (shown in Fig. 4.3(b)), whereas *4M-JScan* have four: joint (Q_j), random (Q_r), serial (Q_s), and functional (Q_f) [207].



(a) *4M-JScan*: four state controller

(b) *2M-JScan*: two state controller

Figure 4.3: State transition machine of mode control in *4M-JScan* and *2M-JScan*

4.3.5 Functionality

The proposed *2M-JScan* architecture functions in two modes of operation: 1.) functional mode, and 2.) test mode. Note that both the modules, *P-Serial* and *P-random*, are operated concurrently.

Functional mode

The functional mode controls two primary operations: 1.) normal function, and 2.) response capture. Normal function is when the circuit perform desired functional operation in normal mode. The test control logic keep serial scan enable (*SSE*) signal at low to operate *P-Serial* scan flip-flops as regular flip-flops. Similarly row address shift register and column driver are disabled to operate *P-random* scan flip-flops as regular flip-flops.

Response capture is part of test procedure performed in functional mode. Once the test stimuli is launched the mode of operation is changed to functional mode ($test\_mode = 0$), and after a delay of one or more clock cycles (called dead cycles) the response is captured by applying a functional clock pulse. Once completed the mode is again changed to test mode ($test\_mode = 1$) to load next test vector.

Test Mode

Test mode controls three primary test operations: 1.) loading/unloading of stimuli/response, 2.) launching of stimuli, and 3.) shift out of response from *MISR*. The test mode is enabled by holding $test\_mode = 1$. Loading/unloading operation in *P-Serial* and *P-random* takes place simultaneously. The *SSE* signal is kept high during this mode. In *P-Serial* the test stimuli are scanned in through *scan-in* lines and responses are compacted using *MISR*. The load/unload operation in *P-random* are performed row by row. First a row is enabled by signal generated from row address shift register, then the column driver activates the desired flip-flops in enabled row for to write test stimuli bit one by one.

Once a row is completed the same procedure is repeated for next row until the last row. Recall that the test patterns are aligned in such a way that both, *P-Serial* and *P-random*, will complete the load/unload operation simultaneously. Once load/unload is completed the test stimuli is launched and test mode is changed to functional mode (*test_mode* $= 0$) for capture. Once all the test sets are applied and responses are compacted in *MISRs* the compacted response is shifted out for comparison. This completes entire test procedure.

4.3.6 Clustering of Scan Flip-flops

Grouping scan flip-flops is another important challenge, because it affects test time, data volume, and test power the grouping algorithm must take these parameters into consideration. To minimize test data volume, the flip-flops for which most of the test pattern contain don't care bits are to be grouped to *P-random* and those flip-flops which contain care bits for most of the test pattern are to be grouped with *P-Serial.* Also the appending pattern has to be considered because it adds up to the data volume. Test time is largely affected by maximum number of write operation being performed for any test pattern in *P-random* and the length of scan chain in *P-Serial.* Therefore, both these two parameters play important role. Test power in Joint-scan is primarily consumed and dominated by *P-Serial* chain. Power consumed due to *P-random* is negligible compared to that of *P-Serial.* Therefore, maintaining length of scan chains is important. We propose here a *1D data clustering* algorithm for grouping of scan cells. Following are the problem statement and algorithm.

Problem: Given a set of flip-flops and corresponding number of care and *don't-care* bits, find two clusters such that the difference between two centroids, i.e., distance, will be optimum. The optimum distance provides the best possible minimization of test time, data volume, and test power. The centroid and distance are computed as follow:

$$centroid = \frac{\sum_{i=0}^{k-1} x_i}{k},\ distance = d_i - d_j \tag{4.2}$$

where x_i is don't care bit count, k is cluster size, and d_i is centroid of cluster i.

First, the flip-flops will be sorted in increasing value of don't care bits. Starting with initial two clusters where cluster-1 will have first flip-flop and cluster-2 will have remaining $N-1$ flip-flops from sorted list. We set a pivot pointer to progressively update the cluster population with new flip-flop(s) from sorted list, update of clustering can be done with varying number of flip-flops. Iteratively, centroid and distance will be computed and compared with the previously computed cluster distance. At the last iteration the best clustering will be identified with optimum pivot point. Following diagram demonstrate the algorithm.

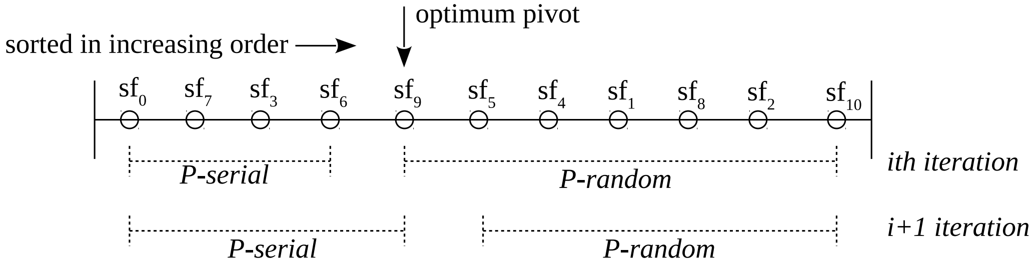


Figure 4.4: Clustering of scan cells using $1D$ clustering algorithm

4.3.7 Computation of Test Parameters

The test time (t_{time}), test data volume (v_{data}), and test power (both peak power (p_{peak}) and average power (p_{avg})) for the proposed architecture are computed as follow. The Following notations are used henceforth.

N	Total number of test sets (T_0 to T_{N-1})
n	Total scan cells ($n_1 + n_2$)
n_1	Total scan cells in *P-random*
n_2	Total scan cells in *P-Serial*
k	Number of scan-in pins for *P-Serial* scan
l	Maximum length of scan chains in *P-Serial*$(= \lceil \frac{n_2}{k} \rceil)$
w	Number of write operation in *P-random*
c	Number of columns in *P-random*
r	Number of rows in *P-random*
a	Number of column address pins for *P-random* $(= \lceil log_2 c \rceil)$
I	Total number of test pins in *2m-jscan* $(= k + a + 2)$
P_{io}	Number of primary *I/O pins*
m	Size of *MISR* (*MISR* in *P-Serial* + *MISR* in *P-random*)

Since the *P-random* and *P-Serial* are operated concurrently the over all test time (in #clock cycle) is computed as follow.

$$t_{time} = (maxcycle * (N + 1)) + N + m \tag{4.3}$$

Where *maxcycle* is load/unload cycle for each test pattern (the detail description of *maxcycle* is provided in Section 4.3.3).

The test data volume, v_{data}, is computed as follows.

$$v_{data} = vs_{data} + vr_{data} + (N * P_{io}) + m \tag{4.4}$$

Where $vs_{data} = N * n_2 + k + vs_{apnd}$ is data volume in *P-Serial* and $vr_{data} = (N * c * r) + (w * c) + w + c + vr_{apnd}$ is data volume in *P-random.* vs_{apnd} and vr_{apnd} are data volume due to pattern appending.

The test power dissipation is computed using the weighted transition metric method which approximate the power by computing the toggles arising in scan flip flops during

Table 4.2: Circuit specification of *S-ISCAS89* benchmark

Circuits	*#PI/PO*	*#Flip-flop*	*#Faults*	*#Testpat*	*%TestCov*
$ss13207$	62/152	9575	$0.22E6$	99	99.08
$ss15850$	77/150	9612	$0.31E6$	129	99.11
$ss5378$	35/49	14857	$0.51E6$	1729	100
$ss9234$	36/39	34815	$0.89E6$	608	99.23
$ss38584$	38/304	39928	$1.61E6$	408	99.31
$ss38417$	28/106	49080	$1.38E6$	418	100

shift operation. For *2M-JScan* the peak and average power are computed as follows:

$$\begin{aligned} p_{avg} &= \frac{toggle_{ps} + toggle_{pr}}{t_{time}} \\ p_{peak} &= Max(instantaneous\ toggle) \end{aligned} \tag{4.5}$$

Where $toggle_{ps}$ and $toggle_{pr}$ are total toggle in *P-Serial* and *P-random* respectively. The instantaneous toggle is the per-cycle toggles activated from *P-Serial* and *P-random*.

4.4 Experimental Result and Discussion

Experiments are performed on scaled *ISCAS89* benchmark circuits, which we abbreviate as *S-ISCAS89*. The scaled up benchmark circuits are recreated from the existing *ISCAS89* circuits. The objective is to get the large number of scan flip-flops without changing the properties of circuit. Therefore, we replicate the *ISCAS89* circuits into a sufficient number of modules and connect these modules with a random glue logic which consist of basic gates like *AND*, *OR*, *NAND*, *NOR* and *INV*. Each design in *S-ISCAS89* is created by replicating its own design from *ISCAS89* benchmark suites.

Design Compiler® and TetraMax® are used for synthesis and test pattern generation respectively. Proposed architecture is evaluated for test time, test data volume, and test power. Results are compared with our earlier *JScan* (referred here as *4M-JScan*)

Table 4.3: Peak and Average Power in comparison with *MSS*

Benchmark Circuit	Peak toggle			Average toggle		
	MSS	*2M-JScan*	%age reduction	*MSS*	*2M-JScan*	%age reduction
*ss*13207	4915	1491	69.66	2729.40	879.84	67.70
*ss*15850	5091	1438	71.75	2774.40	813.04	70.70
*ss*5378	7627	3083	59.50	4036.28	1685.03	58.25
*ss*9234	17940	5132	71.39	9502.72	2836.88	70.14
*ss*38584	20705	2981	85.60	10746.63	1657.86	84.55
*ss*38417	24845	3010	87.88	11864.43	1518.11	87.20

architecture [210], *PRAS* [23], and multiple sequential scan (*MSS*) [146]. Test time, test data volume, and test power are computed using the equations stated in Section 4.3.7. For unbiased comparison the required number of test pins for all the four architectures, *PRAS*, *MSS*, *4M-JScan*, and *2M-JScan*, are kept equal considering *PRAS* as the baseline for pin count. For robust evaluation the size of scaled *ISCAS89* circuits with respect to total flip-flops are varied from 9000 to $49,000$. For all the circuits the full scan design is considered. The *ATPG* (automatic test pattern generation) for stuck-at fault is performed to generate test patterns with *don't-care* bits (*Xs*). The nature of benchmark circuits is provided in Table 4.2.

Experiments for test time and data volume are performed for varied number of pins, understanding the cost of available pin, the motivation has been to design the architecture with given pin constraints. The results are reported in Table 4.4 and Table 4.5. The second column of Table 4.4 and Table 4.5 reports the optimum partition (*PRAN* and *PSER*) for a given number of test pins. Table 4.4 shows that *2M-JScan* reduces test time compared to the earlier architectures. Results show up to 50% reduction in test time compared to the *PRAS* and the *MSS* for sufficient number of test pins. Test data volume compared to the *PRAS* and the *MSS* is reduced by around $40-50\%$. The small quantity of negative reduction compared to the *JScan* is expected because of pattern

Table 4.4: Test time in contrast with *JScan* [210], *PRAS* [23] and *MSS*

Ckts	*#SFF* in	#	Pattern loading time in cycle				
	2M-JScan	*SI*	*2M-*	*4M-*			%red. wrt
	PRAN/PSER	Pin	*JScan*	*JScan*	*PRAS*	*MSS*	JScan/PRAS/MSS
*ss*13207	6705/2865	9	95599	NA	109229	135439	NA/12.4/29.4
		12	57439	NA	101500	94852	NA/43.4/39.4
*ss*15850	6915/2697	9	123893	141847	152105	177253	12.6/18.5/30.1
		12	87749	94706	141697	124108	7.3/38.1/29.2
*ss*5378	8945/5912	9	2834564	3147710	4289836	3670674	9.9/33.9/22.7
		12	1873240	2117236	4107416	2569304	11.5/54.3/27.1
*ss*9234	24722/10093	10	1697079	1912887	2018751	2646024	11.2/15.9/35.8
		13	1227703	1293867	1920956	1924939	5.1/36.1/36.2
*ss*38584	34204/5724	10	921450	969363	1085826	2036744	4.9/15.1/54.7
		13	594642	646350	1015826	1481051	7.9/41.4/59.8
*ss*38417	43300/5780	10	1270007	1335900	1413720	2564856	4.9/10.1/50.4
		13	1128723	1128854	1334182	1865127	0.0/15.4/39.5

appending (this is described in Section 4.3.3). Whereas, the same for test time is not true because the *2M-JScan* uses only one test mode pins unlike the *JScan* which uses two pins.

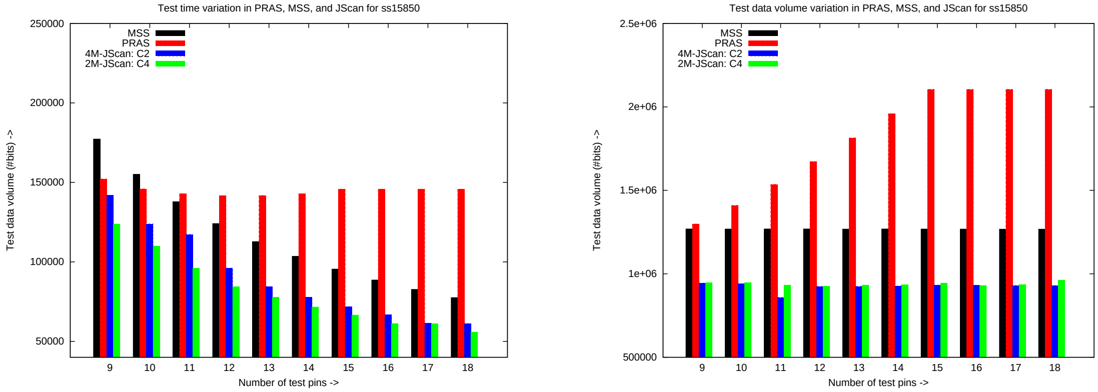
It is expected that in *2M-JScan* the power consumption is dominated by the activity generated from *P-Serial*, the activity from *P-random* is negligible. Table 4.3 reports the peak and average toggle activity considering the same partition as shown in second column of Table 4.4. The computation of average and peak power follows Equation 4.5 in Section 4.3.7. From the table we can observe that the proposed architecture reduces the power by large margin compared to *MSS*, wherein it is to be noted that *PRAS* is the lower bound in power consumption.

Table 4.5: Test data volume in contrast with *JScan* [210], *PRAS* [23] and *MSS*

	#*SFF* in	#	Data volume in bits				
Ckts	*2M-JScan*	*SI*	*2M-*	*4M-*			%red. w.r.t.
	PRAN/PSER	Pin	*JScan*	*JScan*	*PRAS*	*MSS*	JScan/PRAS/MSS
*ss*13207	6705/2865	9	741156	529000	950730	968524	NA/22.1/23.4
		12	628306	599913	1222386	968626	NA/48.6/35.1
*ss*15850	6915/2697	9	946941	944427	1298562	1268851	−0.2/27.1/25.3
		12	976414	952732	1672092	1268725	−2.4/41.6/23.1
*ss*5378	8945/5912	9	20051681	19862981	34342177	25827809	−0.9/41.6/22.3
		12	18929643	18887025	45417963	25820896	−0.2/58.3/26.6
*ss*9234	24722/10093	10	13885425	13683089	18372028	21208872	−1.4/24.4/34.5
		13	13277928	13050185	23440197	21213131	−1.7/43.3/37.4
*ss*38584	34204/5724	10	7698932	7396300	10144322	16430168	−4.0/24.1/53.1
		13	7121708	6974947	12734360	16426499	−2.1/44.1/56.6
*ss*38417	43300/5780	10	11135560	11288391	13073576	20571460	−1.3/14.8/45.8
		13	11880188	11357339	16568044	20567701	−4.6/28.3/42.2

4.4.1 Analysis on Different Possible Configurations

We perform analysis on different possible configurations of *Joint-scan* architecture. The analysis is performed to study the variations in test time and test data volume across the different configuration of Joint-scan architecture. The test power is kept under the limits of less than 40% consumption compared to the baseline multiple sequential scan architecture (*MSS*). The test power is reported in Table 4.3. The area due to the proposed architecture is approximated by the number of scan flip-flops that are grouped to *P-Serial* and *P-random*. Some of our initial results on post-layout routing wire length are reported by Ahlawat et al. [14], which show that the area due to proposed architecture would be around 30 − 40% less compared to the fully random access scan like *PRAS*. The analysis that we report here is for the *ss*15850 design for which we consider the



(a) Test time vs test pin count

(b) Data volume vs test pin count

Figure 4.5: Test time and data volume variation in *MSS*, *PRAS*, and *JScan* with respect to increasing test pin count in $ss15850$ design

partition of flip-flops as 6915 : 2697 for *P-random*:*P-Serial* respectively.

We examine the four possible configurations of the *Joint-scan* architecture. The configuration is based on the usage of available number of test pins. The total available test pins in the *Joint-scan* architecture are distributed/used in three blocks: 1.) *P-Serial*, 2.) *P-random*, and 3.) Test control logic. The four possible configurations are $2-1-p$, $2-1-k$, $1-1-p$, and $1-1-k$. First two configurations are of type *4M-JScan* and the last two are of type *2M-JScan*. The detail description of each configuration is shown in Table 4.6. In each configuration the last number, p and k, indicates whether the column address uses a p address line with column address shift register or a fully parallel k address lines. The possibles address configurations are shown in Figure 4.7(a), 4.7(b), and 4.7(c) respectively for fully serial with single column address shift register, serial with two column address shift register and fully parallel k address lines. Based on the availability of pins the intermediate configurations with p, where $p = 1$ *to* $k-1$, address lines and shift registers are also possible.

The analysis is being carried out in three different aspects of the proposed architecture. The first one is to study the scalability of the proposed architecture which is plotted in Figure 4.5(a) and 4.5(b). The second is to study the effect of address configuration

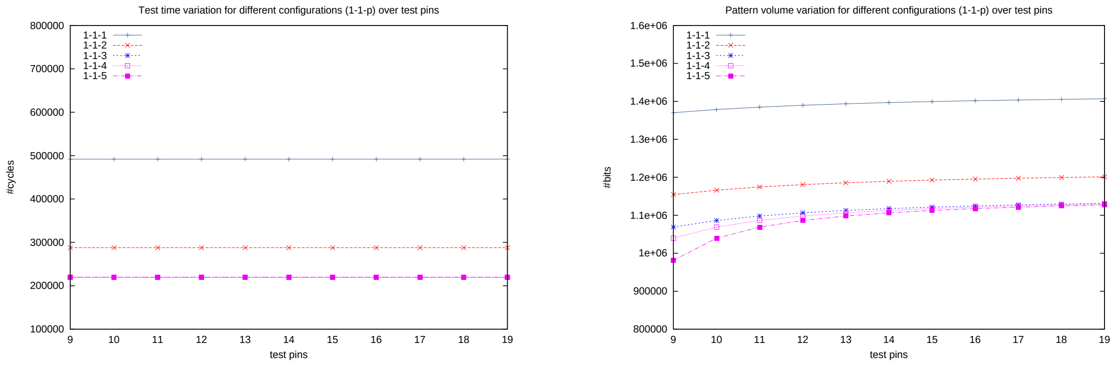
Table 4.6: Four possible configurations of *Joint-scan*

Config	Description	Class
$2-1-p$ (*C1*)	2 mode pins, 1 scan I/O pin, p column address pins	*4M-JScan*
$2-1-k$ (*C2*)	2 mode pins, 1 scan I/O pin, k column address pins	*4M-JScan*
$1-1-p$ (*C3*)	1 mode pins, 1 scan I/O pin, p column address pins	*2M-JScan*
$1-1-k$ (*C4*)	1 mode pins, 1 scan I/O pin, k column address pins	*2M-JScan*

on test time and data volume which is plotted in Figure 4.6(a) and 4.6(b). And, the last analysis we perform is to study the effect of test time and data volume for different dimension of *P-random* and *P-Serial*, which is plotted in Figure 4.8(a) and 4.8(b). In the following sections we explain the behaviour of graphs for address configurations and dimensions with respect to variation in test pin count.

4.4.2 Configurations by Different *P-random* Addressing

The *Joint-scan* architecture consists of partial serial scan architecture (*P-Serial*) and a partial random access scan (*P-random*). For the given *P-Serial* and *P-random*, which are multiple serial scan (*MSS*) and progressive random access scan (*PRAS*), there are different possible configurations which affect the test time and data volume significantly. The first configuration, shown in Figure 4.7(a), with single column address pin and a serial shift register is the most economical in terms of test pin count. For a given number of test pins this configuration dedicate more number pins to *P-Serial* and only 2 pins to *P-random* for $1-1-1$ configuration. Therefore, in this configuration the test time in *P-Serial* could be relatively less than the test time in *P-random*. For circuit $ss15850$, this is evident from the plot in Figure 4.6(a). By increasing the column address pins in



(a) Test time in $1-1-p$

(b) Data volume in $1-1-p$

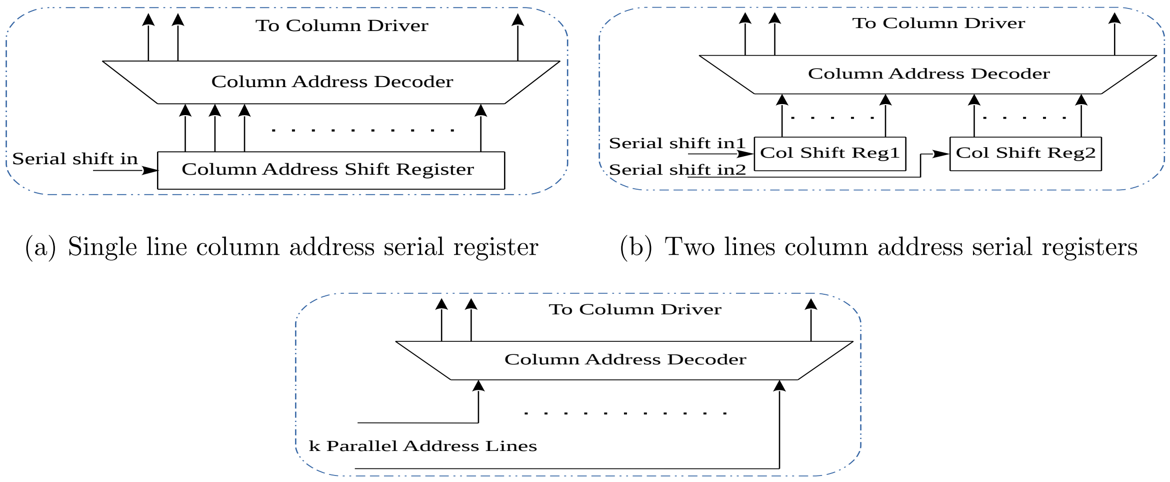
Figure 4.6: Test time and data volume for different configurations in $1-1-p$

P-random the other possible configurations like $1-1-2$, $1-1-3$, $1-1-4$, and $1-1-5$ could be obtained. The last configuration would be with fully parallel column address lines as shown in Figure 4.7(c). Test time and data volume for each of the configurations from $1-1-1$ to $1-1-5$ are plotted in Figure 4.6(a) and 4.6(b). The plots reveals that test time and data volume reduces across the configuration from $1-1-1$ to $1-1-5$. However, it is also to be noted that for each configuration the test time remain same across the varying test pin count. The reason is that the test time contributed by *P-random* is higher than the test time by *P-Serial*. Therefore, the increase in pin count only further reduces the test time in *P-Serial* whereas it has no impact on *P-random*.

Further, when we observe the plot for data volume in Figure 4.6(b) we find the similar behavior like test time plot across the configuration. However, across the test pin the pattern volume increases unlike the test time. The reason for increase is the appended pattern due to widening test time between *P-Serial* and *P-random*. As the test time in *P-Serial* decrease, a more number of bits are required for pattern appending in *P-Serial*.

4.4.3 Alteration in Dimension of *P-Serial* and *P-random*

Next analysis we perform is to find out the optimal dimension of *P-random* and *P-serial*. The dimension of *P-serial* is defined by number of scan chains and maximum length of



(a) Single line column address serial register

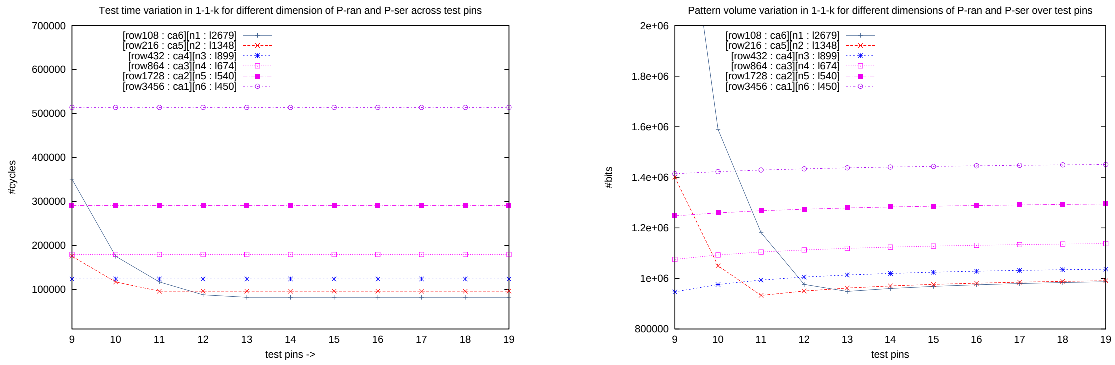
(b) Two lines column address serial registers

(c) Parallel address lines

Figure 4.7: Serial shifting of column address: from single line to $log_2 c$ number of lines, where c is number of columns in *P-random*.

scan chain, and the dimension of *P-random* is defined by number of rows and number of columns. For a fixed number of test pins the dimension of *P-random* is varied in such way that it gives rise to one address pin which can then be used as one of the *scan-in* pin in *P-serial*. For example, if we have total of 9 test pins and we consider the $1-1-k$ configuration of *2M-JScan* then we can have a configuration where just one pin is used for *P-serial* chain, 2 are used for mode control and scan I/O and remaining 6 pins are used for parallel addressing (this means there are total of 2^6) columns in *P-random*). Now, the dimension of *P-serial* and *P-random* can be modified by reducing the total column to half i.e., 2^5 which can now be addressed with only five pins. The one additional pin now can be used for *P-serial scan-in* to form one more scan chain in parallel with the existing chains. Therefore, the dimension of *P-serial* and *P-random* got modified. By the similar process the dimension of *P-serial* and *P-random* can be modified to the extreme where the *P-serial* can be formed with maximum possible scan chains and *P-random* column could be reduced to just one. We plot the scenario in Figure 4.8(a) and 4.8(b) for test time and data volume respectively.

In Figure 4.8(a) and 4.8(b) each graph is plotted for a given configuration which are



(a) Test time variation

(b) Test data volume variation

Figure 4.8: Test time and data volume variation in $1-1-k$ configuration as the change of *P-random* and *P-Serial* dimension

indicated in top right corner of the figure. The configuration $[row108 : ca6][n1 : l2679]$ indicates that in *P-random* there are 108 rows and 2^6 columns and in *P-serial* the number of scan chain is 1 having length of 2679. The top two configurations are to be noted as they reduces the test time and pattern volume compared to rest of the configurations. These two configurations also behave differently from the other configurations as the test pins increases. Such behavior is due to the dominance of test time and pattern volume by *P-serial*. Notice that the test time as well as pattern volume for these configurations reduce quickly as test pins increase. This is because the scan chain length in *P-serial* reduces as rectangular hyperbolic function up to the test pin 13. Thereafter the test time remain constant where as the pattern volume increases linearly with test pins. The constant test time is due to phenomenon that the test time of *P-random* dominates the t_{time} and its behaviour remains unchanged even for increasing number of test pins. However, the test power is affected by increasing number of test pins. This is because of the reduction of test time in *P-serial*, which requires additional bits appending for *P-serial* test patterns. Therefore, the *Joint-scan* architecture could be configured accordingly to obtain the best results for test time and pattern volume while keeping other parameters like routing congestion, area and test power at reasonable value.

4.4.4 Diagnosis, Delay Test and Debugging

Diagnosis and testability for delay defects are two mandatory requirements for any scan based architecture. The standard serial scan and random access scan architecture are proven to be satisfying these requirements [23, 36, 120]. However, the *Join-scan* being a new architecture it brings a fresh challenge on this part. We have not addressed these issues exclusively here in this work, however we assume that since the *Joint-scan* architecture combines serial and random access scan, the established methodology with additional modification could solve these problems. Similarly, the problem of trace-based debugging could be addressed with the *Joint-scan* architecture as it has been proven for the random access scan [116].

4.5 Scan Cell Design for *Joint-scan* Architecture

In this section we propose a new scan cell design for the *Joint-scan* test architecture. In *JScan* test architecture, both the *P-serial* and *P-random* parts are operated simultaneously during test mode. In *P-serial* part the test vector is loaded serially by repetitive application of clock pulse. On the other hand in *P-random* part the test pattern is written by holding the clock signal at a logic high level and enabling a specific cell through row-column address decoder. Because of the conflicting clock requirements *P-serial* and *P-random* can not be operated simultaneously until unless they have separate clock signals or they uses different scan cells.

To overcome this problem we have designed a special scan cell that can be used as a common scan flip-flop in *JScan* test wherein it can be used as a serial scan cell as well as a Random Access Scan (*RAS*) cell [10]. The proposed scan cell does not require separate clock signals and can be operated simultaneously in *P-serial* and *P-random* part as serial scan cell and *RAS* cell respectively. Furthermore, we implemented the *JScan* test architecture using the proposed scan flip-flop. The experimental results show a promising reduction in interconnect wire length compared to the state-of-the-art random access scan and multiple serial scan implementations [14]. The reduced interconnect wire length could help in overcoming the routing congestion which impedes practical implementation of *RAS* architecture.

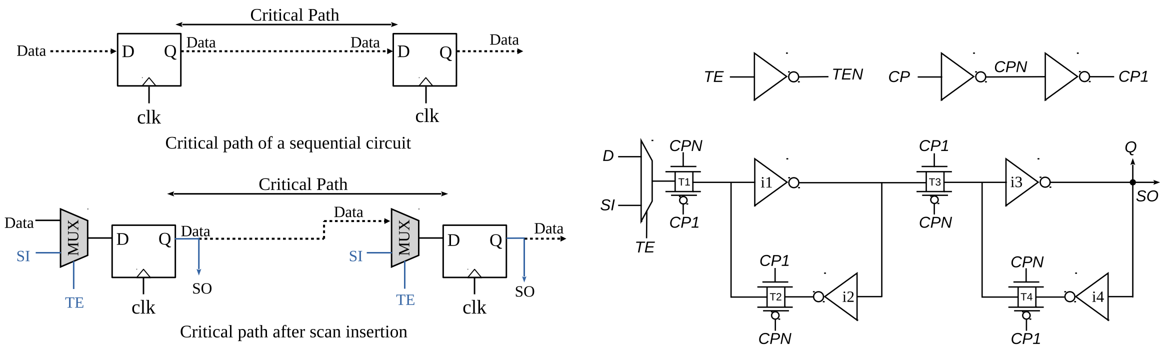
Moreover, the proposed scan cell design eliminates the performance overhead of serial scan design. The performance overhead of serial scan is due to the scan multiplexer [36, 191]. The scan multiplexer falls into each clocked path and adds performance penalty of approximately two gate-delays. A circuit without scan design and with scan design is shown in Figure 4.9(a). As it is observable in Figure 4.9(a), the critical path of a sequential circuit without scan insertion is decided by the longest combinational path between two flip-flops. However, in a scan inserted sequential circuit (see Figure 4.9(a)) the same critical path is elongated by a scan multiplexer at the end of the combinational path. The scan design also adds an extra fanout at the output of a flip-flop. Both of these factors increase the critical path delay, hence reduces functional clock speed by 5% to 10% [36]. This makes it necessary to eliminate the performance overhead of the scan multiplexer. Our proposed scan cell design removes the scan multiplexer from the functional path. The proposed design can help in improving the functional frequency of performance critical designs. It can be used as a regular scan cell in a conventional multiple serial scan design. The major advantages of the proposed scan flip-flop design are as follows:

1. It can be used both as a serial scan cell as well as a *RAS* cell, in the *Joint-scan* test architecture.

2. It eliminates the performance penalty of the serial scan by removing scan multiplexer from the functional path.

3. The proposed design does not introduce any extra control signal and uses the test control signal as a quasi-sequential or low-frequency scan clock.

4. The new scan flip-flop is capable of applying all the tests that can be applied with a conventional scan flip-flop. The proposed design fully complies with the existing industry design and test flow.



(a) Performance overhead of scan design (b) Schematic design of a conventional scan cell

Figure 4.9: Conventional scan cell and performance overhead of scan design

4.5.1 Conventional Scan Cell

A large variety of scan flip-flop implementations are available in the literature [36, 197, 241]. A conventional scan flip-flop design is shown in Figure 4.9(b). This scan cell is a master-slave latch based positive edge triggered muxed input D type flip-flop. The transmission gate $T1$, and the inverter pair connected back to back via transmission gate $T2$ forms the master latch. The slave latch comprises of transmission gate $T3$ and the inverter pair connected back to back via transmission gate $T4$. The multiplexer at the input of master latch selects between functional input (D) and scan_input (SI) depending upon the value of test control signal *test_enable* (TE). In test mode, when TE is high (1), SI is selected and is connected to master latch's input. When the clock signal (CP) is low (0), the value of SI propagates to the master latch. In the meantime, slave latch retains the value from previous clock cycle. The value latched into the master propagates to slave latch when CP turns to high (1), and to the output Q of scan flip-flop. Similarly, when the *test_enable* signal (TE) is set to 0, functional input D is selected, and the circuit operates in functional mode. The conventional scan cell can not be used in a *JScan* test environment because of conflicting operational requirement between *P-serial* and *P-random*. Hence a new scan cell is required which support the *JScan* architecture. We have proposed a scan cell which support the *JScan* architecture. The detailed working of the proposed scan cell is explained in the following subsections.

4.5.2 Proposed Scan Cell Design

This section discusses the working of the proposed scan flip-flop in different modes of operation. The proposed scan flip-flop's schematic design is shown in Figure 4.10. Instead of a multiplexer at master latch's input, the proposed design uses a separate path for loading test vector values into the master latch. Furthermore, the proposed scan flip-flop uses a low-cost dynamic slave latch for shifting of test vectors in the test mode.

In functional mode, functional slave latch's output Q drives the combinational circuit inputs. The master latch of the proposed scan flip-flop is formed by transmission gate $T1$, and inverter pair $(i1, i2)$ connected back to back via transmission gate $T2$. Similarly, the slave latch is formed by transmission gate $T3$, and inverter pair $(i3, i4)$ connected back to back via transmission gate $T4$. The dynamic slave latch comprises transmission gate $T7$ and inverter $i7$. The test mode path is formed by adding transmission gate $T5$, $T6$, buffer $i5$, and inverter $i6$ to the master latch structure. It should be noted that the extra gates added to the master stage to form the test mode input path are not on the functional path. This extra circuitry remains disabled during the functional mode, and

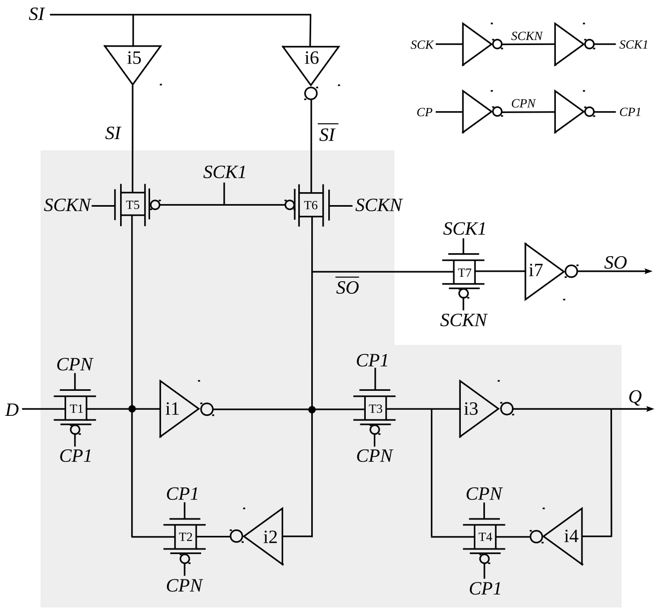


Figure 4.10: Schematic design of the proposed scan flip-flop

the proposed scan flip-flop acts as a regular flip-flop. The master latch and the slave latch are controlled by functional clock signal *CP*. The test mode input path is disabled by the *test_enable* cum scan clock signal *SCK*. Note that, the *SCK* signal in the proposed scan cell is functionally equivalent to the *test_enable* signal *TE*, however, in contrast to the conventional scan design in which *TE* is a purely combinational signal, *SCK* is a low frequency or quasi-sequential signal.

The *SCK* signal is used both as test control as well as a low-frequency scan clock signal in the proposed scan design. Since the scan operation is performed at a much lower frequency, typically at 10MHz to 50MHz, compared to the system or functional clock frequency [171], the routing of *SCK* as a slow frequency scan clock signal will not introduce much overhead in terms of area and power. The routing area and power overhead of *SCK* is analyzed in result and analysis section (see Table 4.10). The details of the working of the proposed design in different modes of operation are explained in the following subsections:

Functional mode

The proposed scan flip-flop works as a regular flip-flop in functional mode. In functional mode, scan clock signal *SCK* is kept at constant logic high (1) level. As long as *SCK* is at constant high (1) level the transmission gate $T5$, and $T6$ remain disabled. This disconnects the test mode input path from the master structure and the proposed scan flip-flop functions as a regular flip-flop. The scan clock signal (SCK) held at constant high (1) level indicates functional mode operation. During the functional mode operation, the transmission gate $T7$ always remains enabled. This keeps the dynamic slave latch always transparent during the functional mode and makes the scan output (SO) toggle every time whenever there is a change in master latch's state. However, that is not of any concern as far as the functional mode operation is concerned because the scan output (SO) drives only the scan path which feeds the scan input (SI) of the successive scan flip-flop.

The scan input path remains disconnected from the master structure during the functional mode of operation. The toggling of scan output SO will create switching activity in the scan path which also happens in the conventional scan design. Because in case of conventional scan cell the combinational load, as well as the scan path, is driven by the Q output of the scan cell. So, in case of conventional scan cell during functional mode, whenever there will be a toggling on the Q output, it will propagate in both the combinational logic as well in the scan path. Also, in conventional scan cell, the scan multiplexer which falls in the scan path would dissipate redundant power in both the modes. In functional mode, the master latch of proposed scan cell gets it's input from the functional data input D. When clock CP is low, the value of functional input D propagates into the functional master latch. When CP turns to high, the value latched into the master propagates to functional slave latch, and to output Q of the scan cell. We verify the said functionality using post-layout simulation.

Test mode

While keeping the functional clock CP held at constant high (1) level, consecutive application of scan clock SCK makes the proposed scan flip-flop to function in test mode. As the functional clock CP is kept high (1), the transmission gate $T1$ always remains disabled in test mode. This disconnects the functional input D from the master latch. During test mode, the master latch gets its input from *scan_input SI*. The consecutive application of scan clock SCK loads the test values into the scan flip-flops. As it can be observed in Figure 4.10, when SCK gets to logic low (0), $T5$ and $T6$ get enabled, and the value of SI is written into the master latch in a similar way to memory write operation.

It should be noted that in test mode since CP is always high (1), the feedback path transmission gate $T2$ always remains enabled. This makes the master latch always trying to retain its previous value. However, it can be observed from Figure 4.10, the test mode input path circuit force writes the SI value simultaneously at both input and output nodes of inverter $i1$ via buffer $i5$ and inverter $i6$ respectively. This makes the write operation faster as far as logical fighting is concerned. When the scan clock SCK gets high (1), the dynamic slave latch transmission gate $T7$ gets enabled, and the master latch starts driving both dynamic slave latch inverter $i7$, and functional slave latch inverter $i3$. This propagates the test value latched into the master during the negative clock cycle, to dynamic slave latch, and to *scan_output SO* of the scan cell.

When scan clock SCK gets to logic low (0), $T7$ gets disabled, and the input parasitic capacitance of inverter $i7$ drives the successive scan cell's *scan_input SI*. Due to the very high impedance of the inverter, the parasitic capacitance does not discharge immediately and takes a long time. The parasitic capacitance discharge time decides the minimum scan clock frequency at which scan shifting can be done. The parasitic capacitance discharge time mainly depends upon two factors: total input capacitance of inverter $i7$, and the charge leakage rate. Hence, for a particular fabrication process technology with well-characterized leakage rate, the discharge time can be optimized by controlling the total input capacitance which in turn depends upon the size of inverter $i7$. The size of inverter $i7$ can be scaled as per the required minimum scan frequency. However, a very low shift frequency is undesirable as it increases the test time, which in turn increases the test cost [23, 36, 171, 210].

It should be noted that in test mode the transmission gate $T3$ always remains enabled. This keeps the functional slave latch always transparent during test mode and makes the output (Q) toggle every time whenever there is a change in master latch's state. Every master latch in scan chain gets its scan input from preceding scan flip-flop's SO output, except the very first master latch in the scan chain which gets its test input from a primary input pin. The scan output SO of the last flip-flop of the scan chain is connected to a primary output pin. The shifting of test vectors into the scan chain is done using the dynamic slave latch. Once the scan chain is loaded, the test vector is launched via the functional slave latch. The test application process is elaborated in detail in the next section.

4.5.3 Application of Test Vectors

The proposed scan flip-flop allows applying all kind of test vectors that can be applied using a conventional scan flip-flop. Before applying any test vectors, scan chain integrity is verified by exercising scan flush test. Scan flush test is applied by propagating an all transition pattern, like 1100, through the scan chain without any response capture cycle in between. The scan clock *SCK* is always kept high during functional mode. When functional clock *CP* is high (1), falling edge on *SCK* switches the circuit from functional mode to test or scan mode. The functional clock *CP* is always kept high (1) during scan shift operation. On arrival of the negative edge of *SCK*, the value of *SI* propagates into master latch via test input path. Next, the rising edge on *SCK* transfers the master latch value to dynamic slave latch and to the scan output node *SO*. By repetitive application of scan clock, the flush patterns are propagated through the scan chain and observed at the primary output pin. The observation of correct input sequence at primary output pin verifies the integrity of the scan chain or scan path.

Note that for a sequential circuit element like scan cell, the faults are modelled at their inputs and output terminals. In the proposed scan cell, there are separate inputs and outputs for test mode and functional mode which forms the respective scan path and functional path. The scan integrity test covers all possible faults on the scan path which comprises faults of test input *SI*, faults of test output *SO*, and faults of the scan path. The scan integrity test does not cover input/output faults of the functional path, i.e., faults of input D of the functional master latch and faults of the output Q of the functional slave latch. As we will see in the next subsection, these faults are covered during stuck-at fault test application.

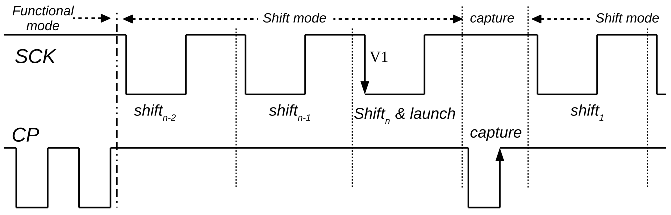


Figure 4.11: Launch and capture in *stuck-at* test

Stuck-at fault test

When clock CP is high (1), falling edge on scan clock SCK indicates the start of scan shifting. The *stuck-at-fault* test is applied by first loading the test vector via scan shifting path and then launching the test vector via functional slave latches. As explained earlier the functional slave latch always remains transparent during scan shifting process. So during the last *shift-cum-launch* cycle when negative edge at scan clock SCK comes, the test vector is applied via output Q of the functional slave latches. It should be noted that the test vector is launched in last shift cycle at the negative edge of the scan clock SCK. After the launch of the test vector, the scan clock SCK is kept high (1) to disable the scan input path. In order to capture the test response, the functional clock CP is clocked once. When the functional clock CP gets low, the functional response is latched into the master latch via functional input D. On arrival of a positive edge on the functional clock CP, the response is propagated to the functional slave latch as well as to the dynamic slave latch. Once the test response is captured, functional clock CK is kept at logic high (1) level. This disconnects the functional input D from the master latch. Now, falling edge on scan clock SCK switches the circuit operation from capture to shift mode. At the same negative edge on SCK, functional response stored in the slave latch gets transferred to the functional master latch of next scan cell via the scan input path. The unloading of test response is done with the simultaneous loading of next test vector.

The timing diagram for *stuck-at-fault* test application is shown in Figure 4.11. As stated earlier, input/output faults of the proposed scan cell's functional path are not covered by the scan chain integrity test. These faults are covered by *stuck-at-fault* test as the test vectors are launched via functional output Q. So the functional output faults (i.e., faults of output Q), if there exists any, will manifest during the test vector launch cycle and will cause launch of wrong test values. The functional input faults (i.e., faults of input D), if there exists any, will manifest during response capture cycle, and will cause capture of wrong test response values. Hence, as a result, the functional input and output faults will result in wrong test vector response. In this way, these faults will be detected at the end of the test response unloading process. Since the same set of test

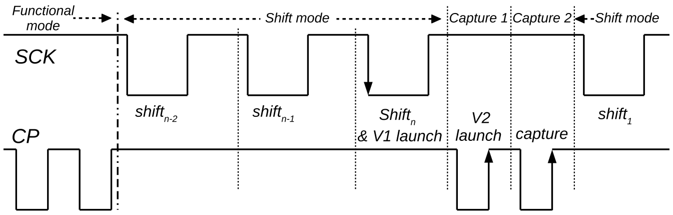


Figure 4.12: Launch of V_1, V_2, and capture in LOC test

vectors are shifted and applied during the *stuck-at-fault* test, the same set of faults are covered as in conventional scan design based test process. It should be noted that the detection of functional input and output faults will be detected as a by-product of the stuck-at fault test and hence no extra test vectors are required for that.

Launch-on-capture test

In *launch-on-capture* (LOC) testing, a test vector pair ($V1$, $V2$) is applied to the CUT. The first vector $V1$ initializes the circuit and the second vector $V2$ launches the transition. The application of initialization vector $V1$ and its response capture is performed in a way similar to *stuck-at-fault* test. As we know that vector $V2$ is the functional response of vector $V1$, response capture of $V1$ acts as the launch of transition vector $V2$. The response of vector $V2$ is captured by applying an *at-speed* functional clock cycle. The loading/unloading of test/response is done in a way similar to *stuck-at-fault* testing. The timing diagram for the *Launch-on-capture* test application is shown in Figure 4.12.

Launch-on-shift test

In *launch-on-shift* (LOS) testing, transition vector $V2$ is a one-bit shift of initialization vector $V1$. In LOS test, the response of $V1$ is not captured. As explained earlier $V1$ is launched at the negative edge of scan clock SCK. As $V2$ is a one-bit shift of $V1$, $V2$ is launched at the next negative edge of SCK. In order to capture the response of $V2$ *at-speed*, the scan clock SCK needs to be clocked at functional clock speed. To apply LOS, SCK must be timing closed. However, it should be noted that SCK in the proposed design is an exact functional equivalent of the *test_enable* signal TE in the conventional scan flip-flop. In contrast to the conventional scan design, the *SCK* in the proposed scan design is used both as test control as well as a low-frequency scan clock. In order to make *SCK* timing closed at the functional frequency, which is also a global signal like the functional clock signal *CK*, it needs to be synthesized like a clock tree which is a very costly task. Application of *LOS* test even in conventional scan design is not possible without timing closed *TE* signal.

The *LOS* based transition delay fault test, without a fast *SCK* (or timing closed *TE* in case of conventional scan design) can be exercised by using the *AND-OR-INVERT* (*AOI*) circuitry proposed by Gefu et al. [138, 240]. However, that also has some associated cost in terms of area and power. Despite the fact that *LOS* offers slightly better test coverage compared to *LOC*, most of the industrial designs support only *LOC* based delay test because of the high implementation cost of *LOS*. Hence, neither conventional nor the proposed design have the capability of exercising *LOS* based delay test without modification. Also, in case of a mixed mode test architecture, the *RAS* inherently lacks *LOS* test application capability. The proposed design does not introduce any new test control signal or testing constraint and can be easily integrated into the existing *DFT* flow.

4.5.4 Post Layout Simulation Results

To evaluate the proposed design, physical layout has been done for the proposed scan cell along with the conventional scan cell [241], and the *PRAS* cell [23]. For conventional scan cell, *IBM's* commercial muxed input *D* flip-flop implementation has been used,

Table 4.7: Layout Area and Leakage Power

Scan cell →	Conventional [241]	PRAS [23]	Proposed
# *Transistors*	32	24	38
Area (um^2)	65.63	51.56	81.80
Leakage power (nW)	10.67	6.40	14.29

which is based on *Power PC* 603 *MS* latch [241]. The layout has been done in standard cell fashion by using Cadence's Virtuoso Layout Suite and *UMC's* 65nm technology library. For symmetric output drives, the *NMOS* to *PMOS* W/L ratio of 2.5, i.e., $(W/L)_p = 2.5(W/L)_n$ has been used. The static or leakage power along with area numbers are given in Table 4.7 for all the three scan cell designs. As it can be observed from Table 4.7, the proposed scan cell has comparatively large area and leakage power among all the three designs. On the other hand, the *PRAS* cell design has minimum area and leakage power. It should be noted that in the mixed mode scan test the proposed scan cell is synthesized both as a serial scan cell as well as *PRAS* cell, so the overall area overhead and leakage power dissipation because of the proposed scan cell will not be much. A chip level implementation of the mixed mode scan test using the proposed scan cell, however, will give a better idea of the overall area and leakage power dissipation overhead.

Furthermore, the standard cell library has been developed for the proposed scan cell using Cadence's Virtuoso Characterization Suite [37], which can easily handle multi-clock and multi-bit sequential cell with complex functionality. The library characterization is done for low leakage and regular threshold voltage (V_t) fabrication process corner. Also, typical n-channel and p-channel *MOSFET* device models have been used for characterization. A nominal temperature of 25^{o}C has been used for standard cell library characterization.

The post-layout timing simulation of the proposed scan flip-flop has been carried out at an operating voltage of 1.2V for frequencies ranging from 2MHz to 1GHz. The post-layout timing diagram at a clock frequency of 500MHz has been used as a reference in Figure 4.13. The timing diagram verifies the efficacy of the proposed design. In order to verify the functionality of the test mode dynamic slave latch the timing simulation of the proposed scan cell is also carried out at very low frequencies. The timing simulation verifies the proper functionality of the dynamic test mode slave latch up to a frequency of 2MHz which is relatively lower than the frequencies at which the internal scan chains of a commercial chip operates.

The internal scan chains of complex commercial chips typically operate at frequencies ranging from $10MHz$ to $50MHz$ due to power dissipation and scan path timing constraints [171]. Furthermore, it is desirable to exercise the scan test at maximum possible scan frequency because it directly impacts the test time and hence the overall test cost. Also, as discussed in the introduction section, the mixed mode scan design can reduce the test power up to $80\% - 90\%$. So, in a mixed mode scan test architecture, the test frequency will not be constrained by test power dissipation, and the scan test can be performed at a relatively much higher frequency. At higher frequencies, the dynamic

Table 4.8: Post Layout Timing Simulation Results at $500MHz$

Functional Mode					
Parameter → Scan cell ↓	*Clock to Q* (t_{cq})	*Setup time* (t_{setup})	*Hold time* (t_{hold})	$t_{cq} + t_{setup}$ (t_{pd})	*Time gain*
Conventional	$0.332ns$	$0.400ns$	$0.0ns$	$0.732ns$	$+64ps$
Proposed	$0.378ns$	$0.290ns$	$0.0ns$	$0.668ns$	
Test Mode					
Conventional	$0.332ns$	$0.400ns$	$0.0ns$	$0.732ns$	$-103ps$
Proposed	$0.284ns$	$0.545ns$	$0.0ns$	$0.829ns$	

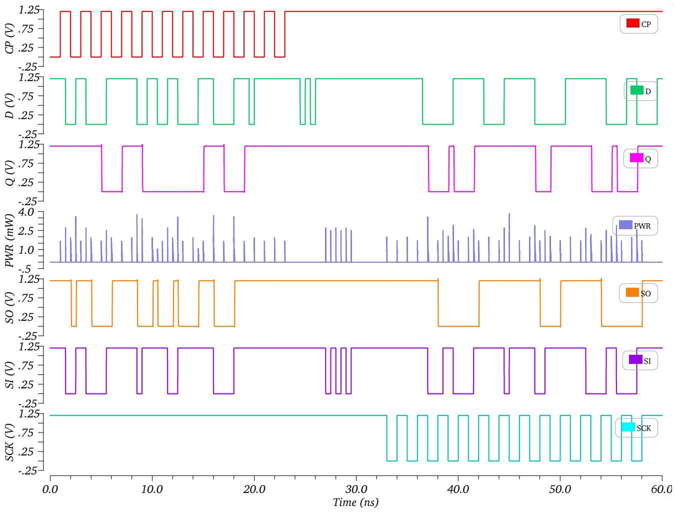


Figure 4.13: Post layout timing waveform at $500MHz$

nature of the test mode slave latch will not be of any concern. Nevertheless, the dynamic slave latch inverter $i7$ can always be sized appropriately to meet even a very low scan test frequency requirement. Hence, the proposed scan cell can be used in any commercial design without any issue as far as the dynamic nature of the test mode slave latch is concerned. Also, it can be observed from the timing diagram that the dynamic power dissipation in both the functional mode as well as in the test mode is almost equal and is below $4mW$.

The post-layout timing simulation results are listed in Table 4.8. In functional mode, clock to Q (t_{cq}) delay, and setup time (t_{setup}) of the proposed design is found to be $378ps$, and $290ps$ respectively. It can be observed from Table 4.8, that t_{cq} of the proposed flip-flop is slightly higher than t_{cq} of the conventional flip-flop. This is due to the extra capacitive loading caused by the dynamic slave latch. There is a considerable decrease in t_{setup} of the proposed scan flip-flop due to the elimination of input multiplexer. Overall the proposed design offers a time saving of $64ps$. In test mode, propagation delay (t_{pd}) of the proposed scan flip-flop degrades by approximately $103ps$. However, test mode performance degradation is not of any concern as the test shifting is done at a frequency much lower than the functional frequency [171]. Overall the time saving offered by the proposed design is approximately equal to 3-4 inverters delay, where typical inverter delay in $65nm$ technology is approximately 15-18ps.

4.6 JScan Implementation using Proposed Scan Cell

This section explores the use of the proposed scan flip-flop to implement *Joint-scan* architecture. In a *Joint-scan* architecture, one set of the *CUT* flip-flops are used to form *P-serial* scan and rest of the flip-flops form *P-random* architecture. Both the serial scan test architecture and *RAS* test architecture are operated concurrently [207]. In the rudimentary implementation of the *Joint-scan*, the flip-flops that are included in serial scan architecture are replaced by a serial scan flip-flop and the flip-flops that are included in *RAS* architecture are replaced by *RAS* cell. In *JScan* design the clock needs to be kept high throughout the test mode to perform *RAS* cell read/write operation. Hence, the serial scan part cannot be operated in concurrent to *RAS* using the conventional serial scan cell. The proposed scan cell overcomes this problem by using test control signal as a slow frequency scan clock and allows to operate both serial and *RAS* design in parallel.

4.6.1 Proposed Scan Cell as a *RAS* Cell

The proposed scan design eliminates the need for two separate scan cell libraries for implementing serial scan part and *RAS* part. It provides a common scan cell that can be used both as a serial scan cell as well as a *RAS* cell. Schematic design of area and performance efficient, progressive random access scan cell proposed by Baik et al. [23] is shown in Figure 4.14. This *PRAS* cell is a modified design of a regular master-slave based positive edge triggered D type flip-flop. The grey part in Figure 4.14 depicts the *PRAS* cell, and the remaining circuit is part of *RAS* test architecture. The grey part of the proposed scan flip-flop shown in Figure 4.10 can be used as a base scan cell. The grey box of the proposed flip-flop maps to the basic *PRAS* cell design except the access pass transistors are replaced by transmission gates ($T5$, $T6$). Both of these basic cells are functionally equivalent. Therefore, the same basic cell of the proposed design can be used as a *PRAS* cell without modification. Note that the proposed scan cell has one extra output node *SO* which remains unconnected in *RAS*. The base scan cell can be synthesized as a *PRAS* cell by mapping the base scan cell in/out signals with corresponding *PRAS* in/out signals to use in *RAS* test architecture. Similarly, the base scan cell can be synthesized as serial scan cell with the full logic shown in Figure 4.10. It is worth to note that with a minor change in synthesis process the proposed scan flip-flop can be synthesized as both serial scan cell and *PRAS* cell. Furthermore, another major advantage of the proposed scan cell is that both serial scan design and *RAS* design can be synthesized with a common clock tree. As discussed in the previous sections the clock is kept at a constant high level to operate the proposed scan cell in test mode, and *test_enable* signal is used as scan clock. This can make the *Joint-scan* test architecture

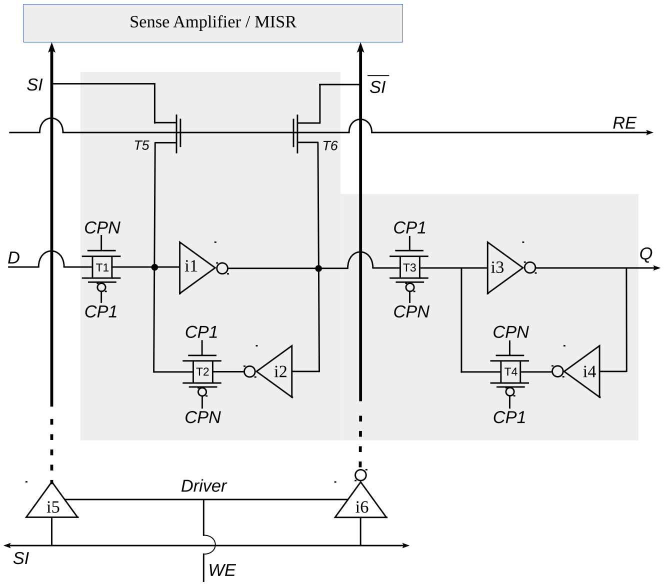


Figure 4.14: Progressive Random Access Scan (*PRAS*) Cell [23]

implementation very efficient and smooth.

4.6.2 Implementation, Experimental Results and Discussion

To validate the efficacy of the proposed scan flip-flop in *JScan* design environment, scaled *ISCAS89* benchmark circuits have been used. The benchmark circuits are up-scaled in terms of size by replicating the same circuit module multiple times. The circuits are synthesized using *Synopsys Design Compiler* for all the three *DFT* architectures, i.e. multiple serial scan (*MSS*) design, *PRAS* design, and *JScan* design. The total routing wire length (*WL*) computed by the tool for all the three test schemes is reported in Table 4.9. As it can be observed from Table 4.9, the full multiple serial scan architecture sets the lowest bound on the total and average wire length. On the other hand, the *RAS* architecture has the highest wire length because of which it has routing congestion issue. The *JScan* test architecture makes use of the best of both multiple serial scan and *RAS* architecture. The *JScan* architecture provides a way of trade-off between hardware overhead and test time, test data volume, and test power.

The routing wire length has been calculated with two different number of test pins.

Table 4.9: Routing wire length (*WL*) in *JScan*, *PRAS*, and *MSS*

Ckts	Layout Area (μm^2)	Arch.	Total *WL* (*mm*)	Avg. WL/net (μm)	%red
*S*38417	65149.30	*PRAS*	279.54	45.02	0.00
		JScan	221.28	35.29	20.84
		MSS	167.63	28.37	40.03
	75929.06	*PRAS*	281.59	45.32	0.00
		JScan	232.11	36.96	17.57
		MSS	179.17	30.71	46.37
*S*5378	7400.92	*PRAS*	29.52	28.72	0.00
		JScan	26.74	26.60	9.42
		MSS	24.00	25.87	18.69
	8616.95	*PRAS*	29.43	28.69	0.00
		JScan	27.30	27.16	7.40
		MSS	22.86	25.6	22.48
*S*838	1612	*PRAS*	4.45	15.72	0.00
		JScan	4.05	15.55	9.12
		MSS	3.15	14.85	29.20

The first row results correspond to the *S38417* and *S5378* benchmark circuits implemented with eleven *scan-in* pins. On the other hand, the second row results correspond to circuit implementation with ten *scan-in* pins. It can be observed from Table 4.9 that the interconnect routing wire length also improves with the higher number of test pins. With a higher number of test pins available at the chip level the number of internal scan chain increase which in turn reduces the scan chain length or in other words the number of scan cell in a scan chain. The decrease in scan chain length decreases the scan path length. As a result, the overall wire length reduces as the number of test pins increases. The maximum percentage reduction in total wire length is approximately

Table 4.10: Routing cost of scan clock SCK in terms of buffer/inverter area and power

Ckts	Parameter	CP	SCK	CP	SCK	CP	SCK	CP	SCK
↓	↓	$1GHz$	$100MHz$	$1GHz$	$50MHz$	$1GHz$	$10MHz$	$1GHz$	plain
$S9234$	#buff/inv	124	22	122	21	101	19	105	4
	buff-area	814.1	144.6	798.7	140.8	669.4	133.1	687.4	22.1
	D-power	14.99	0.71	14.75	0.35	13.58	0.078	13.16	0.02
$S13207$	#buff/inv	130	56	134	49	123	50	137	21
	buff-area	1158.0	586.2	1183.7	509.8	1122.2	535.4	1204.5	92.8
	D-power	43.66	3.322	43.05	1.544	44.99	0.3723	41.27	0.08
$S38417$	#buff/inv	576	245	589	260	622	237	576	92
	buff-area	3901.4	1689.6	3993.6	1783.0	4188.2	1644.8	3875.8	476.6
	D-power	160.7	12.03	154.5	5.923	168.3	1.329	147.2	0.28
$S5378$	#buff/inv	90	18	90	18	82	17	93	4
	buff-area	590.1	120.3	590.1	120.3	545.3	113.9	610.6	24.3
	D-power	11.33	2.68	11.26	0.56	12.09	0.12	9.95	0.03

21% for $S38417$ benchmark circuit. Note that the percentage of reduction is computed with respect to the *PRAS* based implementation. Also for all the circuits, the *JScan* routing length is relatively lesser than *PRAS* wire length. The *JScan* test technique can effectively alleviate the routing congestion problem of *PRAS* design.

The improvement in wire length depends upon the number of scan cells included in the serial scan part of the *Joint-scan* architecture. Across all the benchmark circuits used in the experiment, 30% to 40% of the total scan cells go to the serial part, and 60% to 70% go to *RAS* part. As explained in the Introduction section, routing congestion is a serious issue which needs to be resolved for the practical implementation of *RAS*. The routing congestion in *RAS* comes from the routing of two extra global signals SI and $\overline{SI}$ in addition to the routing of regular signals of serial scan architecture i.e., clock and *test_enable* (see Figure 4.14). The *Joint-scan* test architecture improves the routing congestion by making use of the multiple serial scan and achieves a significant reduction in test time and test data volume by using *RAS*. The multiple serial scan sets a lower bar on the wire length whereas the *RAS* has the highest wire length. The *JScan* test architecture has a wire length which is somewhere in between the two test architectures *MSS* and *RAS*.

The proposed *JScan* test architecture is capable of exercising both *stuck-at* fault test as well as the *launch-on-capture* (*LOC*) based transition delay fault test. However, as the *RAS* design inherently lacks *launch-on-shift* (*LOS*) test, *JScan* design cannot be used to exercise *LOS* test.

As explained in Section II, in *Joint-scan* scan architecture, the *SCK* signal is used both as test control as well as a low-frequency scan clock. Hence, routing of *SCK* as a global quasi-sequential signal will introduce some area and power overhead. In order to analyze the effect of scan frequency over the routing area and power overhead *SCK* routing, we carry out place and route on a few benchmark circuits with *SCK* frequency varying from $10MHz$ to $100MHz$. Also, the frequency of functional clock *CP* for routing was kept constant at $1GHz$. The results are presented in Table 4.10. We report the results on the number of buffers/inverters inserted in clock tree, total buffer/inverter area, and total dynamic power dissipation (D-power) by all the buffers/inverters. It should be noted that these numbers do not give the total routing area and power dissipation. The total routing area and power depend upon various factors such as types of routing metal layers used, the total number of *VIA's*, the total length of metal wire used, and metal wire length used for a particular routing metal layer.

It can be observed from the Table 4.10, that the area and power overhead due to the routing of *SCK* as a low frequencies signal is slightly higher than the routing of *SCK* as a pure combinational signal. However, it is much lower than the routing overhead of the functional clock signal *CP*. Therefore, synthesizing and routing the SCK as a low-frequency clock signal is cost effective. Also from the table, we observe that there is a slight variation in buffers/inverters count for *CP* signal due to the stochastic nature of routing and placement.

4.7 Conclusion

We have proposed a framework for new *Joint-scan DfT* architecture. The *4M-JScan* is reviewed and a new *2M-JScan* is proposed. The proposed architecture is shown to be effective in minimizing the test time, test data volume, and test power compared to the *MSS* and *PRAS*. Also the *2M-JScan* is compared with *4M-JScan*. New test control mechanism is proposed which enables the architecture to function in similar way the standard *DfT* architecture does. Procedure for test set alignment is described. We also developed a $1D$ clustering algorithm for efficient grouping/clustering of scan flip-flops. We anticipate that *2M-JScan* can be well scaled for billion gates design by appropriately guiding the grouping algorithm to provide optimum size for *P-serial* and *P-random*. The proposed architecture is limited to *stuck-at* fault and transition fault testing with *launch-on-capture* strategy.

Further, we have proposed a new scan flip-flop for *Joint-scan* architecture which can be used both as a serial scan cell as well as a *RAS* cell. The *2M-JScan* architecture implemented with proposed scan flip-flop shows a promising reduction in interconnect wire length, test data volume, and test application time. Moreover, the proposed scan flip-flop design eliminates the performance penalty of the serial scan by removing scan multiplexer from the functional path. The proposed scan cell can be used for contemporary serial scan architecture to eliminate scan performance overhead.

— ∗ — ∗ —

Chapter 5

Scan Chain Diagnosis

Scan based diagnosis plays a critical role in failure mode analysis for yield improvement. However, as the logic circuitry associated with scan chains constitute a significant fraction of a chip's total area the scan chain itself can be subject to defects. In some cases, it has been observed that scan chain failures may account up to 50% of total chip failures. Hence, scan chain testing and diagnosis have become very crucial in recent years. In this chapter we propose a hardware-assisted low complexity and area efficient scan chain diagnosis technique. The proposed technique is simple to implement and provides maximum diagnostic resolution for *stuck-at* faults. The proposed technique can be further extended to diagnose the timing faults of scan chain at the cost of slightly diminished diagnostic resolution.

5.1 Introduction

Almost every complex circuit today employ scan-based Design-for Testability (*DfT*) architecture to enhance testability and diagnostic capabilities. The effectiveness of these techniques rely upon the proper functioning of the scan design i.e., the scan chain itself is fault free. However, it has been reported in the literature that the chip area consumed by the scan path along with the scan control signals may range from 15% to 30%. Furthermore, it has been observed that 10% to 30% of the total defects may cause the scan chain to fail [99]. A faulty scan chain hinders the chip failure mode analysis process for yield enhancement. The presence of a fault in scan chains can be easily detected by performing a simple flush test, however, identifying the exact location of the fault in the scan chain is a tedious task. Several techniques have been proposed in the literature for diagnosing scan chain faults. These techniques can be broadly classified into three main categories: 1.) simulation-based [84, 117, 195], 2.) tester-based [60, 194, 196], and 3.) hardware-assisted [64, 67, 145, 181, 236]. The simulation-based techniques make use of the failure logs of scan tests from the tester and use inject-and-evaluate approach to identify the defective scan cell [84]. Unfortunately, due to limited failing buffer size capacity of the tester, not all the failing pattern/cycles data can be recorded. The limited availability of failing log data from the tester may result in false identification or reduced diagnostic resolution. The simulation based techniques do not have any hardware overhead, however, the diagnostic resolution is comparatively poor compared to hardware-assisted diagnosis techniques. Also, the simulation-based techniques are very complex and time consuming.

In tester-based approach, a physical failure analysis (*PFA*) device is used in conjunction with the tester. While the scan patterns are shifted in through the scan chain by the tester, *PFA* is used to observe and analyze the defective response of the scan cells at different physical locations. In one such approach [60], E-beam is used to detect the toggles in a scan chain while a stream of alternating $0's$ and $1's$ are shifted in the scan chain by the tester. The toggles disappear at the location of the *stuck-at* fault. The tester based techniques can accurately locate the defect site when probable physical defect location is very small, however, the diagnosis time and associated *PFA* cost is prohibitively high.

A good review of simulation-based and tester-based diagnosis techniques is provided in a recent work by Huang et al. [99]. These techniques do not use any extra circuitry for diagnosis of scan chain faults and hence no area overhead. However, these techniques have some serious drawbacks like poor diagnostic resolution, long diagnosis time, and prohibitively high instrumentation cost. The hardware-assisted techniques, on the other hand, have a much better diagnostic resolution as compared to simulation-based and tester-based techniques. These techniques often use custom scan cell design or add extra circuitry in the scan path to facilitate scan chain fault diagnosis. The extra circuitry added to the scan design is used either to *set*/*reset* every scan cell in the scan chain or flip its content.

Edirisooriya et al. [67] insert a two-input *XOR* gate between the scan cells. The *XOR* gate is used as an inverter to invert the content of a scan cell before shifting it into the next scan cell. By inserting a *XOR* gate between every pair of scan cells a maximum resolution of 1 can be achieved. In another hardware-based technique, the authors [181] use partner scan chains to diagnose scan chain faults. These partner scan chains are connected with each other through extra routing wires such that during diagnostic mode, the content of bad scan chain can be observed by the good partner scan chain. Narayan and Das [145] use extra circuitry to *set*/*reset* the scan out port of the scan cell for effective diagnosis of scan chain faults. In another such approach [236], Wu uses custom scan cells with *set*/*reset* capability to locate the exact position of a fault in the scan chain. The technique in [236] is capable of diagnosing both *stuck-at* and *hold-time* faults. However, there is a trade-off between diagnostic resolution and hardware overhead. In a recent work [64], Dounavi et al. use a modified scan architecture that uses charging/discharging of a global diagnosis-line to locate the faulty scan cell. This technique has a high diagnostic resolution, however, the static power consumption is very high because of a direct path formation between power supply node and the ground node.

In spite of having much better diagnostic resolution compared to simulation and tester based techniques, most of the hardware-based techniques are unacceptable in practical designs. The hardware-based techniques have some practical issues like power consumption due to extra circuitry during functional mode, testing of extra circuitry, and area overhead. In this Chapter, we propose a new hardware-assisted scan chain diagnosis technique. In the proposed technique, a custom scan cell design is used to achieve *set*/*reset* capability for enhancing the diagnostic capability. The proposed technique can be used to diagnose both stuck-at and timing faults in the scan chain. The major advantages of the proposed technique are as follows:

1. The proposed scan cell design eliminates the need for separate *set* and *reset* control signal for diagnosis and has minimum area overhead compared to the existing hardware-assisted diagnosing techniques.

2. The proposed diagnosis technique has the maximum diagnostic resolution (to be precise, 1) for *stuck-at* faults and hence can locate the exact position of the faulty scan cell.

3. The proposed technique can be extended to diagnose hold time faults at the cost of slightly diminished diagnostic resolution.

4. The new scan cell has little performance overhead compared to conventional scan cell design.

The remainder of the chapter is organized as follows: Section 5.1.1 describes the fault type identification procedure. Section 5.2 elaborates on the implementation of the proposed scan cell. Further, this section describes diagnosis of stuck-at scan chain faults. Section 5.3 explains the extension of the proposed technique to diagnose hold time faults in a scan chain. Section 5.4 compares the merits of the proposed technique with the existing hardware-based techniques. This section also discusses the post layout timing simulation results. The chapter is concluded in Section 5.5.

5.1.1 Preliminaries

In scan inserted circuits, the scan chain is used to load/unload the test stimuli/ response for test and diagnosis purpose. However, the presence of defects in the scan circuitry may cause the scan chain to fail and invalidate the test and diagnosis process. Many factors contribute to the presence of defects in a scan chain. Some of these factors, such as faulty fabrication line, technology process variation effects, resistive short or open interconnect, may manifest in a number of ways. The presence of a short or open interconnect at the scan cell's input/output ports can result either as a *stuck-at-0* or a *stuck-at-1* fault. A

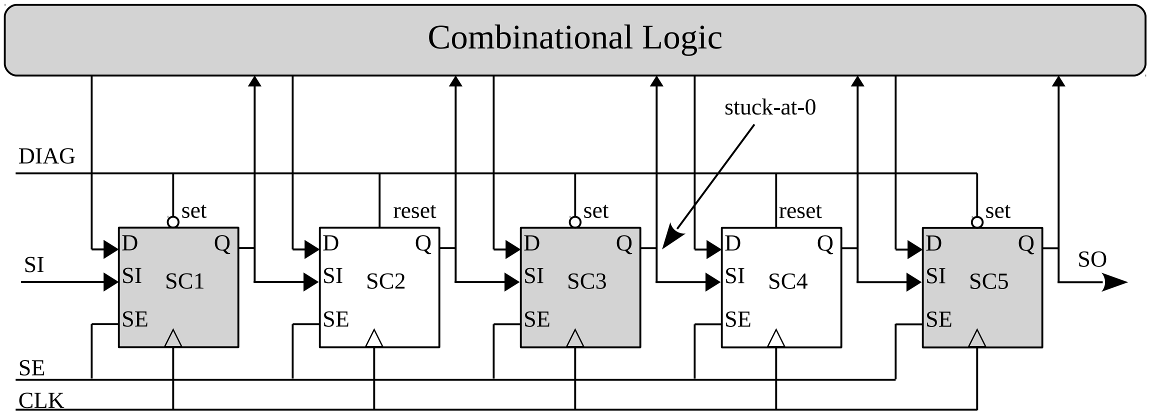


Figure 5.1: Example Scan chain with a single $sa0$ fault

faulty scan cell with a stuck-at-0 (1) fault will change all the bits shifting through it to 0 (1). As a result, the shifted out sequence will consist all $0's$ (1) no matter what sequence is shifted in.

Hold time fault is another commonly observed scan chain fault. The hold time violation is generally caused by clock skew, which could be a result of process variation or improper clock tree design. There are three types of scan chain hold time faults which have been observed in practice [236]. In a type-I hold time fault, a scan cell captures faulty value only in case its scan input SI has a rising transition i.e., 0 to 1 transition. The type-I fault is caused by the faster rise time of the Q of the preceding scan cell that feeds the faulty scan cell with clock skew. The clock skew at the faulty scan cell is large enough that the rising transition of the preceding scan cell from the present cycle overwrites the valid data at the SI input before it gets captured in the faulty cell. However, for falling transition the clock skew is small enough such that before the arrival of the falling transition at scan input the valid data gets captured. Similarly, in a type-II fault, the faulty scan cell fails only if its SI input has a falling transition i.e., 1 to 0 transition. In case of type-III fault, the faulty scan cell has a large enough clock skew such that it fails for both rising and falling transitions at the SI input.

The impact of hold time faults on scan chain is different than stuck-at faults. A hold time fault allows a proper shifting of sequences consisting all $0's$ or all $1's$. However, if a transition exists in the sequence the faulty scan cell acts as a shadow copy of its preceding scan cell. As a result, the faulty scan cell captures the same value as its preceding scan cell. This makes the bit following the problematic transition appears one cycle earlier than expected at the scan out port of the scan chain. In another word, the scan chain appears effectively one bit shorter because of a hold time violation. For example, in the case of a good scan chain, the input sequence 0001011100 will be observed as it is on the scan out port *SO*. However, in case of a type-I, type-II, and type-III hold time faults the shifted out response will be 0001111100, 0000001100, and 0000101110 respectively.

The scan chain failure detection and fault type identification are relatively very simple tasks, however, finding the location of the faulty scan cell (i.e., scan chain diagnosis) is a very tedious process. In this work, extra circuitry is added to the conventional scan cell which can be used to *set*/*reset* the scan cell during scan/diagnose mode.

5.1.2 Fault Type Identification

The scan chain failure detection and fault type can be identified by applying a simple flush test. To explain fault type identification and scan chain diagnosis process, an example scan chain of length five is shown in Figure 5.1. To identify the *stuck-at* fault a flush test comprising a sequence of all $0's$ or all $1's$ can be used. To identify a *stuck-at-1* fault a stream of $0's$ can be shifted in by applying the clock five times. After five clock cycle, if a 1 appears at the *scan_out* port then there is a *stuck-at-1* fault. In case a 0 appears at the (*SO*) port after five clock cycles then that means the scan chain is free from *stuck-at-1* faults. Similarly, a sequence of $1's$ can be used to verify the scan chain for *stuck-at-0* faults. Once the type of fault is identified the scan chain diagnosis techniques are used to identify the exact location of the faulty scan cell.

To identify a hold time fault in a scan chain the input sequences as proposed in [236] can be used. These sequences can identify the presence as well as the type of hold time faults. The sequences for the hold type faults are as follows: (A) Sequence 1: 1111100000 (B) Sequence 2: 0000011111 (C) Sequence 3: 0000010000 (D) Sequence 4: 1111101111

By observing the responses of all the above four sequences the type of hold time fault can be identified. The presence of extra $1's$ in the response of sequence 1 with no extra $0's$ in the response of sequence 2 indicates a type-I fault. The number of extra $1's$ in the observed response gives the number of faults present in the scan chain. Similarly, the presence of extra $0's$ in the response of sequence 2 with no extra $1's$ in the response of sequence 1 indicates a type-II fault. Again, the number of extra $0's$ in the observed response gives the number of faults present. If there is one-bit shift in the observed responses for both sequence 3 and sequence 4 then there is a type-III fault. The number of extra shifts in the response gives the number of faults present in the scan chain.

5.2 Diagnosis of Stuck-at fault in scan chain

Consider a *stuck-at-0* fault at the output port (Q) of 3^{rd} scan cell $(SC3)$ in the example scan chain shown in Figure 5.1. As explained in the previous subsection, a *stuck-at-0* fault can be identified by simply shifting a stream of $1's$ through the scan chain. The steps involved in fault diagnosis process and the values of all the scan cells during fault identification and diagnosis procedure are listed in Table 5.1.

Table 5.1: Scan chain state in different diagnosis phases

scan cell states						
cycle no. ↓	*SC1*	*SC2*	*SC3*	*SC4*	*SC5*	phase ↓
initialization	X	X	X	X	X	
$clk - i_1$	1	X	X	X	X	
$clk - i_2$	1	1	X	X	X	
$clk - i_3$	1	1	0	X	X	
$clk - i_4$	1	1	0	0	X	
$clk - i_5$	1	1	0	0	0	← *detection*
cell *set*	1	1	0	1	1	
$clk - d_1$	1	1	0	0	1	
$clk - d_2$	1	1	0	0	0	← *diagnosed*

The initial values in all the scan cells are represented by *X's* as the values in the scan cells are not known when the scan chain is switched from functional mode to test or shift mode. The cycle-wise values of the scan cells during the shift operation can also be seen from Table 5.1. The fifth cycle $clk - i_5$ in the table represents the fault identification cycle. Let us assume that all the scan cells in the scan chain have *set* capability and all the cells can be *set* to 1 together by using a global control signal. After the detection of a *stuck-at-0* fault in the fifth clock cycle, the control signal can be asserted to *set* all the scan cells to 1. During the *set* operation, the clock either needs to be kept inactive or at a constant logic high/low level depending upon the requirement of the *set*/*reset* circuitry. Once the cells are *set* to 1, the scan chain contents are shifted out by applying the scan clock. At the first clock cycle, a 1 will be observed at the *SO* port. However, on the second clock cycle, a 0 will be observed due to the *stuck-at-0* fault that exists at the output port of the third scan cell SC_3. The second scan/diagnose cycle $clk - d_2$ shown in Table 5.1 represents the cycle in which the fault is diagnosed. Observation of a faulty 0 value after two clock cycles locates the position of the faulty scan cell (i.e., SC_3 in the example scan chain).

Similarly, the scan chain can be diagnosed for *stuck-at-1* fault by following the same sequence of steps. Instead of a sequence of all $1's$, a sequence of all $0's$ needs to be used to identify a *stuck-at-1* fault. Also, all the scan cells must have *reset* capability. The number of clock cycles after which the faulty value is observed at the *SO* port tells the location or index of the faulty scan cell. In order to diagnose a scan chain for both *stuck-at-0* as well as *stuck-at-1* faults, all the scan cells must have both *set* and *reset* capability. However, to integrate both set and reset capability in the scan cell most of the existing hardware-assisted scan chain diagnosis techniques use two separate global control signals. The control signal routing and associated circuitry make the area overhead prohibitively high to implement these techniques in practical designs for diagnosis purpose. We propose a very low-cost scan cell design that has both *set* and *reset* capability and uses only a single global control signal.

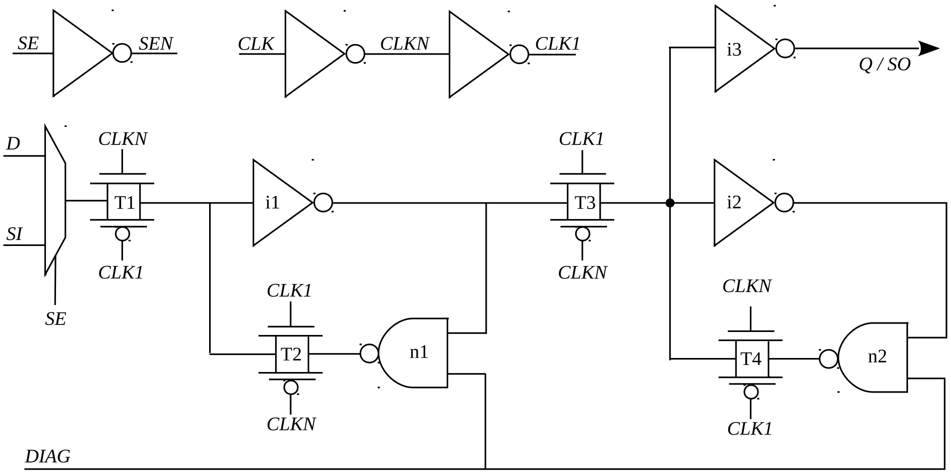


Figure 5.2: Proposed scan cell design with *set/reset* circuitry

Proposed Scan Cell

The schematic design of the proposed scan cell is shown in Figure 5.2. As it can be observed from the schematic design, in the proposed scan cell the feedback path inverter is replaced by a *NAND* gate in both master and slave latch. The inputs of *NAND* gate n_1 are driven by inverter i_1 and diagnosis control line called *DIAG*. Similarly, inputs of *NAND* gate n_2 are driven by inverter i_2 and *DIAG*. The logical operation performed by *NAND* gate is represented by $c = \overline{a.b}$. So, if one of the *NAND* gate input is permanently tied to logic 1 value, it imitates as a logical inverter. Therefore, during the functional mode and the test mode, *DIAG* is permanently kept at a logic high level and the proposed

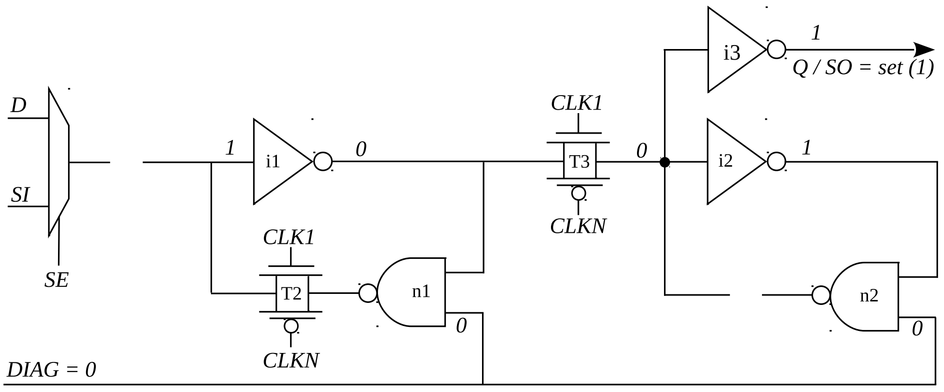


Figure 5.3: *Set* operation in proposed scan cell

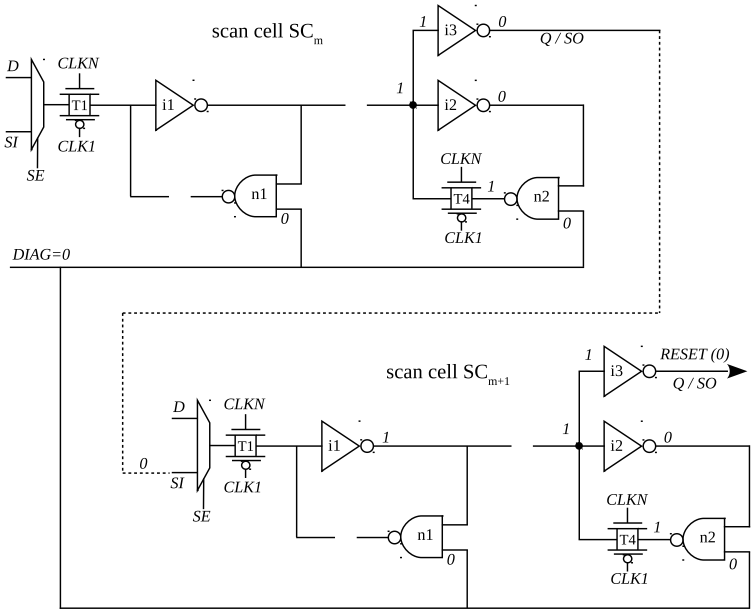


Figure 5.4: *Reset* operation in proposed scan cell

scan cell operates as a regular scan cell. In diagnostic mode of operation, *DIAG* line will set or reset the scan cell depending upon whether the clock signal *CLK* is at the positive logic level or negative logic level. It should be noted that the *set/reset* operation can not be performed during the functional mode of operation.

Set operation

The *set* operation of the proposed scan cell is explained in Figure 5.3. To set the scan cell, a negative pulse of width t_{set} at *DIAG* line is applied when the scan clock *CLK* is high (1). When the *CLK* changes from low (0) to high (1), transmission gate $T1$ gets disabled and isolates the master latch from the scan input *SI*. Now as soon as *DIAG* is pulled low (0), the output of *NAND* gate n_1 is forced to logic 1. Also, a high value at *CLK* enables the transmission gates $T2$ and $T3$. This allows the *NAND* to drive inverter i_1 and force its output to logic 0. This, in turn, forces the outputs of inverter i_2 and i_3 to logic 1. This sets the *scan-out* port *SO* of the scan cell to 1. Note that transmission gate $T4$ remains disabled as long as *CLK* is 1, the output of slave latch's *NAND* gate n_2 remains disconnected.

The timing requirement for the *set* operation is shown in Figure 5.5. It should be noted that the $DIAG$ signal must be pulled to 0 only after the clock CLK gets 0 to 1. The $DIAG$ must be pulled back to 1 before the clock CLK gets to 0. Thus, the minimum pulse width time for which the $DIAG$ signal needs to be pulled down for proper set operation must satisfy the condition $t_{set} < 0.5 * t_{CLK}$. The t_{set} is decided by the feedback path delay of the master latch. This timing requirement is easily satisfiable as the scan clock is generally supplied by the tester. Also, in the test mode, the scan clock frequency is kept low due to the power constraint. In case t_{set} is higher than half of the scan clock period the clock can be kept inactive at a logic high (1) level until the scan cell is properly *set*. Once the scan cell is *set*, the CLK can be again activated to shift out the scan chain values for diagnosis. So, there is no impact of the *set* operation timing requirement on the scan shift clock frequency. The *set* operation can be used to diagnose the scan chain *stuck-at-0* faults.

Reset **operation**

The *reset* operation of the proposed scan cell is shown in Figure 5.4. Application of a low pulse of width t_{reset} on the $DIAG$ line while the CLK is low (0) will *reset* the scan cell. Figure 5.4 shows two successive scan cells with index number SC_m and SC_{m+1} of a scan chain. When the CLK gets 1 to 0, transmission gate $T1$ and $T4$ gets enabled. Also, a low (0) CLK signal disables the transmission gate $T3$ and isolates the slave latch from the master latch.

The *reset* operation is performed by using slave latch's $NAND$ gate n_2 while the clock CLK is low (0). While CLK is 0, pulling down the $DIAG$ signal to 0 will force the output of the $NAND$ gate n_2 to 1. This, in turn forces, the output of inverter i_3 (i_2) to 0 and *reset* the output port SO. However, as the CLK turns from 0 to 1, the value stored in the master latch gets transferred to the slave latch. This may overwrite the *reset* value of the scan out port SO. To ensure that the SO port of the scan cell remains *reset*, the clock period must be such that the SO port *reset* (0) value gets enough time to propagate through the scan path and get latched into master latch of the succeeding

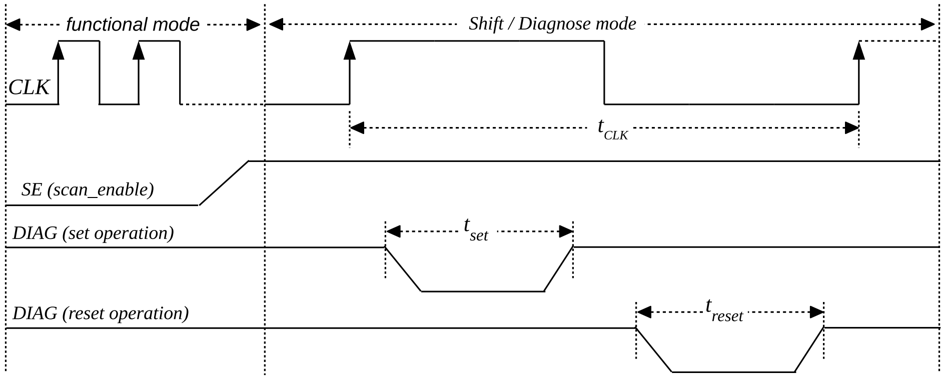


Figure 5.5: Timing requirement for *set/reset* operation

scan cell, while the CLK is 0. For example, while the CLK is low (0), the *reset* (0) value at the SO port of SC_m cell must get enough time to propagate to master latch of the succeeding cell SC_{m+1} and get latched into it. Similarly, at the same time master latch of scan cell SC_m will latch the *reset* value (0) coming from its preceding scan cell SC_{m-1}. This will ensure that when CLK turns high, the scan cells remain *reset.*

It should be noted that since the first scan cell in the scan chain gets its value from the primary input pin *scan_input*, the value at the primary input pin must supply a 0 during the *reset* cycle. Note that transmission gate $T2$ remains disabled as long as CLK is 0, the output of slave latch's $NAND$ gate n_1 remains disconnected. The timing requirement for the *reset* operation is shown in Figure 5.5. The minimum pulse width t_{reset} required to *reset* the scan cell is decided by the feedback path delay of the slave latch. The minimum pulse width t_{set} and t_{reset} will be equal for both *set* and *reset* operations. The other timing constraints for *reset* operation comes from the propagation time taken by the *reset* value of a scan cell to get latched into the master latch of succeeding scan cell. However, to avoid any performance penalty on the scan shift frequency the clock can be kept inactive at a constant 0 level during the *reset* operation. Once the scan cells are properly *reset*, scan clock can be applied to scan out the scan chain states for diagnosis. By using the *reset* capability, the scan chain can be diagnosed for *stuck-at-1* faults.

The proposed scan cell is capable of diagnosing *stuck-at* faults, however, it can not be used to diagnose hold time faults. To diagnose both *stuck-at* and hold time faults we

Table 5.2: Hold time fault diagnosis

Type-I fault diagnosis						
cycle no. ↓	*SC1*	*SC2*	*SC3*	*SC4*	*SC5*	phase ↓
initialization	X	X	X	X	X	
$clk - d_1$	1	0	1	0	1	cell *set-reset*
$clk - d_2$	0	1	1	1	0	
$clk - d_3$	1	0	1	1	1	
$clk - d_4$	0	1	1	1	1	← *diagnosed*
Type-II fault diagnosis						
initialization	X	X	X	X	X	
$clk - d_1$	1	0	1	0	1	cell *set-reset*
$clk - d_2$	0	1	0	1	0	
$clk - d_3$	1	0	0	0	1	
$clk - d_4$	0	1	0	0	0	
$clk - d_5$	1	0	0	0	0	← *diagnosed*
Type-III fault diagnosis						
initialization	X	X	X	X	X	
$clk - d_1$	1	0	1	0	1	cell *set-reset*
$clk - d_2$	0	1	1	1	0	
$clk - d_3$	1	0	0	1	1	
$clk - d_4$	1	1	1	0	1	← *diagnosed*

propose a revised version of the proposed scan cell. The revised version of the proposed scan cell is more area efficient, however, it has a slightly diminished diagnostic resolution. The hold time fault diagnosis technique is explained in detail in the next section.

5.3 Diagnosis of Hold Time Fault in Scan Chain

The presence and type of hold time fault can be identified by using the procedure as explained in Subsection 5.1.1. Once the fault type is identified the exact location of the fault needs to be found out. To locate the fault in the scan chain, a sequence of alternative 1 and 0 bits can be used. Assume that the scan chain shown in Figure 5.1 has scan cells at odd index number with set capability and scan cells at even index number with reset capability. Thus by using the $DIAG$ signal the scan cells SC_1, SC_3, SC_5 can be set and scan cells SC_2, SC_4 can be reset simultaneously. Further, assume that the scan cell SC_3 has a type-I hold time fault at its SI input and the output of scan cell SC_5 is directly driving the *scan-out* port of the scan chain..

The steps involved in diagnosing the type-I fault are listed in upper part of Table 5.2. The scan chain is initialized with unknown values in all the scan cells. In the first diagnose cycle d_1, all the odd cells get *set* while all the even cells get *reset*. As scan cell SC_5 gets *set*, a 1 value is observed at *scan-out* port during the first diagnose cycle d_1. In the second clock d_2, a fast rising transition occurs at the Q output of SC_2 (i.e., SI input of SC_3). Due to the type-I fault, the faulty scan cell SC_3 behave as a shadow of scan cell SC_2 and captures the same value. However, all the other scan cells capture the right values and 0 is observed in the second diagnosis cycle d_2. In the third cycle d_3, since there is a falling transition at the SI input of SC_3 the fault remains inactive. All the scan cells shift the correct values in cycle d_3 and 1 is observed. In the fourth cycle d_4 the scan cell again captures the wrong value due a fast rising transition at its SI input. Because of the faulty value captured during the second cycle, 1 is observed at the *scan-out* in the fourth cycle d_4.

Observation of the faulty value in fourth diagnose cycle d_4 locates the position of the type-1 fault, i.e., between third and fourth scan cell from the output side. Hence the type-I faults can be diagnosed with a maximum diagnostic resolution of 1 (precise diagnosis). However, assuming a type-II fault at the same location and following the diagnosis steps listed in middle part of Table 5.2, the error is observed at the fifth clock cycle d_5. Therefore, the probable fault location is either between SC_3 and SC_2 or SC_2

and SC_1. Hence, the diagnostic resolution for type-II faults is 2. Similarly, type-III faults can also be diagnosed using the similar steps which are listed in lower part of Table 5.2. The diagnostic resolution for type-III faults is also 1. A similar procedure can be used to diagnose the stuck-at fault. The steps involved in diagnosing a *stuck-at-0* and *stuck-at-1* fault at the output of scan cell SC_3 are listed in Table 5.3. Note that in the case of *stuck-at-1* fault the diagnostic resolution is 2 as the diagnostic value at the fault site is same as the fault value. The overall diagnostic resolution for the proposed scheme is $1 - 2$.

The proposed scan cell shown in Figure 5.2 is further modified for diagnosing hold time faults. To diagnose the hold time violation faults we propose a revised version of the proposed scan cell implementation. The schematic diagram of the variant of the proposed scan cell is shown in Figure 5.6. As it can be observed from the schematic diagram, instead of an inverter in the feedback path the master latch has a *NOR* gate. It should be noted that to set the cell in positive clock cycle the $\overline{DIAG}$ signal must be

Table 5.3: Stuck-at fault diagnosis

stuck-at-0 fault diagnosis						
cycle no. ↓	*SC1*	*SC2*	*SC3*	*SC4*	*SC5*	phase ↓
initialization	X	X	X	X	X	
$clk - d_1$	1	0	0	0	1	cell *set-reset*
$clk - d_2$	0	1	0	0	0	
$clk - d_3$	1	0	0	0	0	← *diagnosed*
stuck-at-1 fault diagnosis						
initialization	X	X	X	X	X	
$clk - d_1$	1	0	1	0	1	cell *set-reset*
$clk - d_2$	0	1	1	1	0	
$clk - d_3$	1	0	1	1	1	
$clk - d_4$	0	1	1	1	1	← *diagnosed*

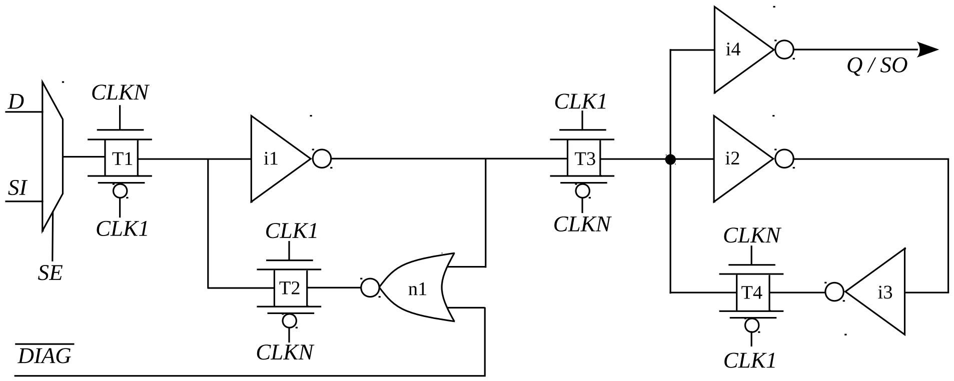


Figure 5.6: Revised version of proposed scan cell with *reset*

pulled high (1). This can be done by using the inverted value of the $DIAG$ signal. Note that the cell can only be reset, however, to diagnose hold time faults some cells need to be reset and other cells need to be set. To achieve set capability instead of using a *NOR* gate in the master latch's feedback path if a *NAND* gate is used the cell will have set capability. By using two separate scan cells one with set feature and another with reset feature and connecting them in an alternative fashion in a scan chain both stuck-at and hold time faults can be diagnosed.

Table 5.4: Comparison with existing diagnosis techniques

Parameters of comparison				
Diagnosis Technique ↓	*transistor count*	*control signal(s)*	*maximum resolution*	*fault type(s)*
Edirisooriya et al. [67]	16	1	1	*sa*
Schafer et al. [181]	6 – 8	1	1	*sa*
Narayan et al. [145]	12	1 – 2	1 – 2	*sa*
Yuejian Wu [236]	6 – 8	1 – 2	1	*sa, hold*
Dounavi et al. [64]	4	2	1	*sa*
Proposed design 1	4	1	1	*sa*
Proposed design 2	2	1	1 – 2	*sa, hold*

5.4 Experimental Results

The post layout timing simulation of the proposed scan cell design has been carried out using *UMCs* 65nm technology at operating voltage of 1.2V. The post layout timing simulation results are listed in Table 5.5. The propagation delay t_{pd} of the proposed scan cell with both set and reset features degrades by 10ps. Whereas the propagation delay of proposed scan cell design with only reset feature degrade by 19ps as compared to the conventional scan cell. This can be minimized by placing the *NOR* based scan cell with reset capability only in the non-critical functional paths. The scan design implementation in [67, 236] loads the output node of each scan cell with the parasitic capacitance of *XOR* gate or multiplexer respectively.

In terms of area overhead, the proposed scan cell is highly efficient compared to the existing hardware-based diagnosis techniques. The proposed design uses only four extra transistors compared to conventional scan cell. Note that the revised design of the proposed scan cell uses only two extra transistors per scan cell. Comparison of the existing and the proposed scan chain diagnostic techniques in terms of transistor count, number of control signals, maximum diagnostic resolution, and type of faults diagnosed is done in Table 5.4. It can be observed that the proposed technique uses least numbers of extra transistors. The number of extra transistors used in [67, 145, 181, 236] is 3 times to 8 times higher. The technique in [64] uses four extra transistors and a global diagnose line. The static power consumption during the test mode in [64], is prohibitively high because of a direct path formation between power supply node and the ground node. Furthermore, it can only diagnose *stuck-at* faults.

The proposed technique uses only one global diagnose control signal for both the proposed approaches. The proposed technique also offers the maximum diagnostic resolution for both *stuck-at-0* and *stuck-at-1* faults. By extending the proposed scan cell design both hold time and *stuck-at* faults can be diagnosed. The technique by Wu et al. [236] can diagnose both stuck-at and hold time faults, however, it uses 3 to 4 times extra transistors compared to the proposed technique. In terms of area overhead, the revised design of the proposed scan cell is highly efficient. The proposed scan cell design

Table 5.5: Post layout timing simulation results at 500MHz

<table>
<tr><th colspan="3">Proposed scan cell with both set and reset feature</th></tr>
<tr><td>Mode ↓ Parameter →</td><td>$t_{cq} + t_{setup} = t_{pd}$</td><td><i>Time gain</i></td></tr>
<tr><td>Functional / Test mode</td><td>$401ps + 354ps = 755ps$</td><td>$-10ps$</td></tr>
<tr><th colspan="3">Proposed scan cell with only set feature</th></tr>
<tr><td>Functional / Test mode</td><td>$401ps + 353ps = 754ps$</td><td>$-9ps$</td></tr>
<tr><th colspan="3">Proposed scan cell with only reset feature</th></tr>
<tr><td>Functional / Test mode</td><td>$400ps + 369ps = 769ps$</td><td>$-24ps$</td></tr>
<tr><th colspan="3">Conventional scan cell</th></tr>
<tr><td>Functional / Test mode</td><td>$397ps + 348ps = 745ps$</td><td>$--$</td></tr>
</table>

does not have any special timing and test requirements and complies with the existing industrial design and test flow.

5.5 Conclusion

In this Chapter, we have proposed a hardware assisted scan chain fault diagnosis technique. The proposed technique is very simple to implement and is capable of diagnosing both *stuck-at* and timing faults in scan chains. The proposed technique has the maximum diagnostic resolution for *stuck-at* faults and hence can locate the exact position of the faulty scan cell. Furthermore, the proposed technique is capable of diagnosing hold time faults with slightly diminished diagnostic resolution In addition to that, the proposed design incurs insignificant area overhead and has minimal performance overhead.

$- * - * -$

Chapter 6

Scan Flip-flop Design

The central element in scan based *DfT* architecture is a scan cell. Various scan cell designs are available in literature which are motivated either by performance or test power [36, 138, 185, 191, 197, 203, 241, 248]. In this Chapter, we target three issues through an efficient scan cell design. The design of scan cell is motivated by the following three objectives:

1. Elimination of scan performance overhead,
2. Elimination of unnecessary combinational power dissipation in test mode, and
3. Enabling *launch-off-shift* (*LOS*) based delay test using slow scan enable signal.

The rest of the chapter is organized as follows: Section 6.1 gives details on the existing scan cell design based approaches on eliminating scan performance overhead along with the proposed scan cell. It further explains the compatibility of the proposed scan cell design with the existing industry test flow. In Section 6.3, a scan cell design which allows exercising *LOS* test with slow scan_enable test control signal is explained. Section 6.2 explains the scan cell based technique to eliminate unnecessary switching activity in combinational logic during test mode. Finally the Chapter is concluded in Section 6.4.

6.1 Elimination of Performance Penalty of Scan

The demand for high performance system-on-chips (SoC) in communication and computing has been growing continuously. To meet the performance goals, very aggressive circuit design techniques such as the use of smallest possible logic depth are being practiced. Replacement of normal flip-flops with scan flip-flops adds an additional multiplexer delay to critical path. Furthermore as the combinational depth decreases, the performance degradation caused by scan multiplexer delay become more critical. Elimination of the scan multiplexer delay off the functional path has become crucial in maintaining the circuit performance.

Traditionally partial-scan has been the alternative approach to alleviate the performance penalty of full scan. Partial-scan provides a trade-off between the ease of testing and the cost associated with the scan design by selecting a subset of the flip-flops for inclusion in the scan chain. Existing partial-scan methods can be categorized as: structure based [19, 40, 47, 85, 119], testability measures based [34, 107, 177, 237, 238], and test-generation based [7, 94, 131, 132, 187]. The partial-scan techniques can effectively overcome the scan performance overhead issue, however, they face some severe issues. The partial-scan techniques do not comply with the existing industry design flow and also incapable of insuring the quality of full-scan. Moreover, Increasing complexity of integrated circuits has forced the industry to abandon partial scan, which necessitates a computationally demanding and unaffordable sequential $ATPG$ (or combinational $ATPG$ with time frame expansion), and to rather adopt full scan despite its costs.

Another approach to tackle the scan performance issue is to use high performance scan cell designs. In recent past the author in [191] has used a transformation technique to move multiplexer from input side to output side of scan cell. This technique looks promising, however, it fails to eliminate the penalty when the scan cell is in critical path from both the sides. We propose a new transistor level scan cell design to eliminate the scan multiplexer off the functional path [9]. The proposed scan cell uses separate master latch for functional and test mode where as the slave latch is same in both the modes. Our proposed scan flip-flop fully comply with the conventional test flow. Post layout experimental results justify the effectiveness of the proposed scan cell design in eliminating the performance penalty of scan, and thus in improving the timing performance of integrated circuits.

6.1.1 Preliminaries and Proposed Scan Cell

There are many types of scan cell implementations available in literature. In this work we have considered *IBM's* commercial multiplexed input *D-type* flip-flop implementation as reference, which is a very robust scan cell. This scan cell is based on Power PC 603 *MS* latch [197], which has also been studied in [244, 245]. This scan cell is a master slave latch based positive edge triggered multiplexed input *D-type* flip-flop. The *MOSFET* based transistor level implementation of the reference scan cell used in this work is shown in Figure 6.1. The circuit highlighted by the dashed line (red colored) rectangle is the multiplexer that select either functional input (D) or scan input (SI) depending upon

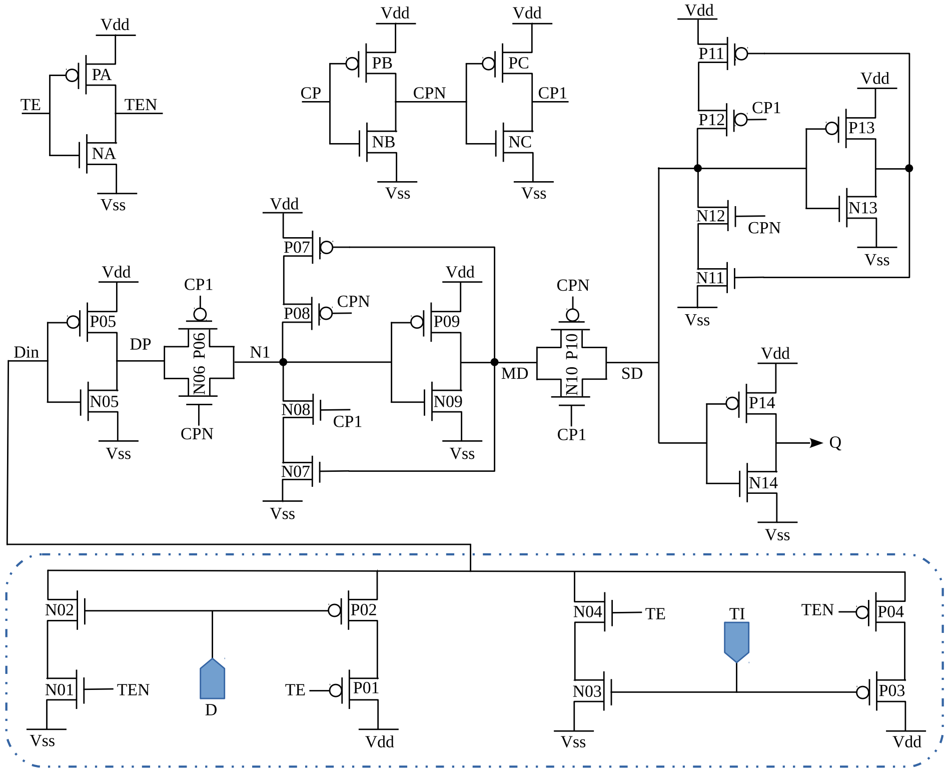


Figure 6.1: Transistor level implementation of the reference scan flip-flop [197, 245]

the value of control signal scan enable (SE). The circuit between the nodes DP and MD is master latch and the circuit between the node MD and the output (Q) forms the slave latch. When the scan enable signal (SE) is set to high (1), SI is selected and the circuit operates in test mode. The value of SI then propagates into the master latch when clock (CP) is low. Meanwhile, the nodes in the slave latch retain the values from the previous clock cycle. When CP turns to high, the signal stored in the master latch propagates into the slave latch and to the output of the scan cell. In the same way when SE is set to 0, D is selected and the circuit operates in functional mode.

Proposed Scan Cell Design

In this section we discuss the working of proposed scan flip-flop in different modes of operation. The transistor level implementation of the proposed scan flip-flop is shown in Figure 6.2. The proposed design uses separate master latches for test mode and functional mode, and a common slave latch in both modes of operation. The MOS transistor implementation of the proposed scan flip-flop is shown in Figure 6.2. The scan enable (SE) signal decides which master latch will drive the slave latch during different modes of operation. The functional master latch (shown in blue colored dotted line rectangle) is a back to back connected gated inverters circuit. The feedback inverter is gated using the clock (CP) signal and the output inverter is gated using the scan enable (SE) signal. During test mode the gated output inverter keeps the master latch isolated from the slave latch. In test mode the slave latch gets its input from the shadow master latch. The shadow master latch (shown in red colored dashed line rectangle), is also a back to back connected inverter circuit in which the feedback inverter is gated using the clock (CP) signal and the output inverter is a static inverter. During the functional mode the shadow master latch latch is isolated from the slave by the transmission gate connected between shadow latch output and slave latch input. In functional mode the slave latch get its input from functional master latch. The operation of the proposed scan flip-flop in functional mode and test mode is explained in the following subsections:

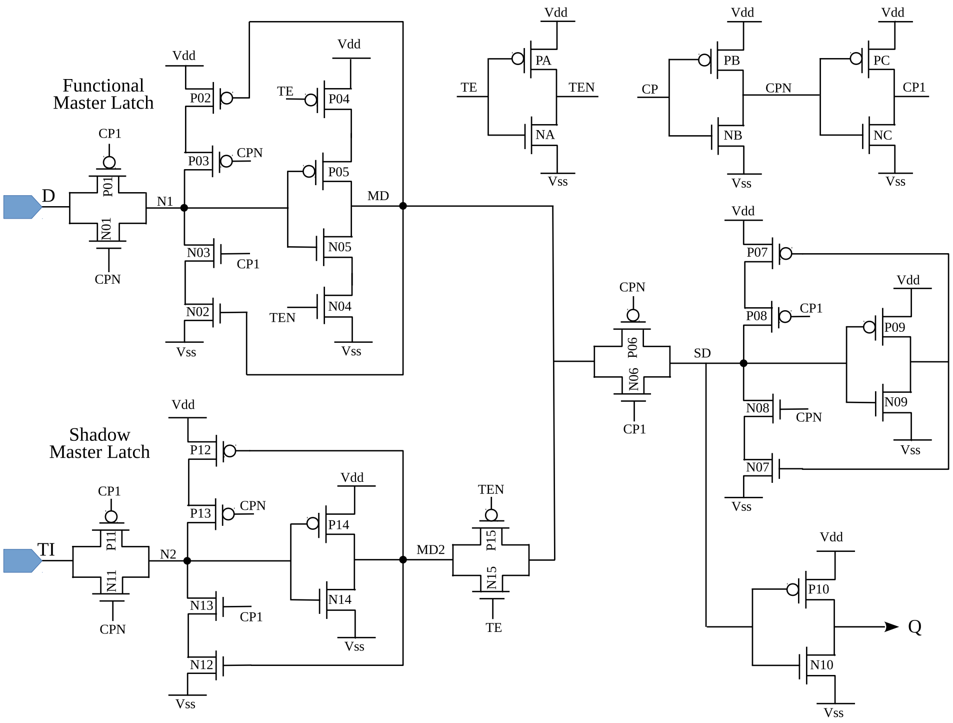


Figure 6.2: Proposed Scan Flip-Flop Implementation

Functional mode

In functional mode the scan flip-flop functions as a normal flip flop. The scan enable (SE) signal switches the scan flip-flop between functional mode and test mode. When SE is low (0) the scan flip-flop operates in functional mode and when SE is high (1), it operates in test mode. When the clock signal is inactive, a falling edge on the SE (indicating the starting of the functional mode) switch the circuit operation from test mode to functional mode. The SE and TEN signals get to logic low (0) and logic high (1) levels respectively. The gated output inverter of the functional master latch get enabled because the upper $PMOS$ transistor $P04$ and the lower $NMOS$ transistor $N04$ turns ON. At the same time the transmission gate connected between the shadow master latch and the slave latch gets *open* because the $PMOS$ transistor $P15$ and the $NMOS$ transistor $N15$ turns OFF. This keeps the shadow master latch isolated from the slave latch during functional mode. The slave latch gets its input from the functional master latch. When clock (CP) is low the value of D propagates into the functional master latch. Meanwhile, the nodes in the slave latch retain the values from the previous clock cycle. When *CP* turns to high, the signal stored in the functional master latch propagates into the slave latch and to the output of the scan cell.

Test mode

When the clock signal is inactive, rising edge on the *SE* (indicating the beginning of shift cycles) switches the circuit operation from functional mode to test mode. As the circuit enters into test mode the *SE* and *TEN* signals get to logic high (1) and logic low (0) levels respectively. The gated inverter of the functional master latch get disabled because the upper *PMOS* transistor $P04$ and the lower *NMOS* transistor $N04$ turn *OFF*. This makes the functional master latch's output node floating and disconnects it from the slave latch during test mode. At the same time the *PMOS* transistor $P15$ and the *NMOS* transistor $N15$ turn *ON*, which makes the transmission gate, connected between the shadow master latch and the slave latch, gets *open*. This makes the slave latch to get its input from shadow master latch during test mode. The shadow master latch gets its input from the preceding scan cell output, except the shadow master latch of the very first scan cell in the scan chain which is fed by primary input pin scan input (*SI*).

The test mode is further sub-categorized in shift mode and capture mode. During shift mode, at first shift cycle the value of *SI* propagates into the shadow master latch, when clock (*CP*) is low. Meanwhile, the nodes in the slave latch retain the values from the previous clock cycle. When clock (*CP*) turns to high, the signal stored in the shadow master latch propagates into the slave latch and to the output of the scan cell. The shadow master latch of the succeeding scan cells get their inputs fed by the preceding scan cell. With the repetitive application of the shift cycle, the test values can be loaded in the scan chain in the same way as in case of a conventional scan flip-flop based scan chain. In the last shift cycle also called the launch cycle the values loaded into the scan cells along with the primary inputs, are applied to the combinational logic. When the clock signal is inactive a falling edge on the *SE* (indicating the beginning of capture cycle) switch the circuit operation from test mode to capture/functional mode. As the circuit enters into capture/functional mode, the functional master latch get enabled and the shadow master latch get disconnected from the slave latch. The functional response value propagates to the functional master latch when the capture pulse comes (CP turns to high). Subsequent to the capture window, before the arrival of first shift cycle of the next test vector loading session, when the clock signal is inactive, a rising edge on the SE (indicating the beginning of shift cycles) switches the circuit into test mode again. This makes the functional master latch disabled and the shadow master latch enabled. The arrival of first shift cycle (CP turns to high) propagates the functional response stored into the slave latch to the shadow master latch and to the output of the scan cell.

In test mode, at every negative clock cycle the functional master latch inputs fed by the combinational logic will set the $N1$ node to a logic level as per the functional logic value but will not propagate to the slave latch input MD. However during positive clock cycle the clocked inverter feedback network of the functional master latch, which make the $N1$ node retains its logic value during normal/functional mode, may force the $N1$ node to switch because the MD node is not driven by the functional master latch. However other than some extra switching activity in the functional master latch it has no effect on the scan shift process. The switching of $N1$ node could have been avoided by using transmission gate to gate the output of the functional master latch instead of domino style gated inverter. However, domino style gated inverter has advantages over transmission gate based output gating. The first advantage is that during functional mode the internal nodes between $P04$ and $P05$, and $N04$ and $N05$ will be always at VDD and $GROUND$ level respectively. This improves the speed of the inverter or decrease the *clock* to Q delay of the slave latch. On the other hand, in transmission gate based output gating, whenever there will be a switching the parasitic capacitance of the transmission gate will have to be charged/discharged. The second advantage is that the domino style gated inverter is more resilient to self loading effect. To avoid switching of $N2$ node during functional mode we used the transmission gate at the output of the shadow master latch.

6.1.2 Test Quality Considerations

The proposed scan flip-flop comply with the conventional design and test flow and inherit all test application capabilities of the conventional scan flip-flop. In scan based testing first of all the integrity of the scan chain is ensured by scan flush test. Scan flush test is applied by running a few patterns through the scan chain without any capture operation in between. In the proposed scan cell, all these patterns are shifted through the shadow master latch and traverse through the scan chain. This way, all the faults on the scan path are covered as in the conventional scan flush testing. As the scan path does not traverse through the functional master latch, the faults on their input and output are not covered during scan flush testing, however, these faults can be covered in stuck-at fault test which is discussed below.

Stuck-at fault test

In stuck-at fault testing, when the clock signal is inactive, a rising edge on the *SE* disables the functional master latch and opens the shadow master latch. Before the first shift cycle, dead clock time is inserted to ensure that the *SE* signal gets enough time to switch from low (0) to high (1) value. This dead time also ensures the propagation of correct *SI* signal value to the slave latch input node. The loading/unloading and response capture is carried out in the same manner as explained in the test mode section. As stated earlier, the scan chain integrity test can not detect the faults at the input and output of functional master latch. These faults if there exist any, will manifest during stuck-at-fault capture cycle. When the capture pulse comes the combinational response captured into the functional master latch will propagate to the slave latch. The arrival of next shift cycle (*CP* turns to high) to load next test vector, will propagates the functional response stored into the slave latch to the shadow master latch of the next scan cell and to the output of the scan cell. This way the faults at input and output node of functional master latch will be detected at the end of the response unloading process. Also, as the same patterns are loaded and applied, and the same responses are unloaded and observed, the same set of faults in the combinational logic is covered as in conventional testing.

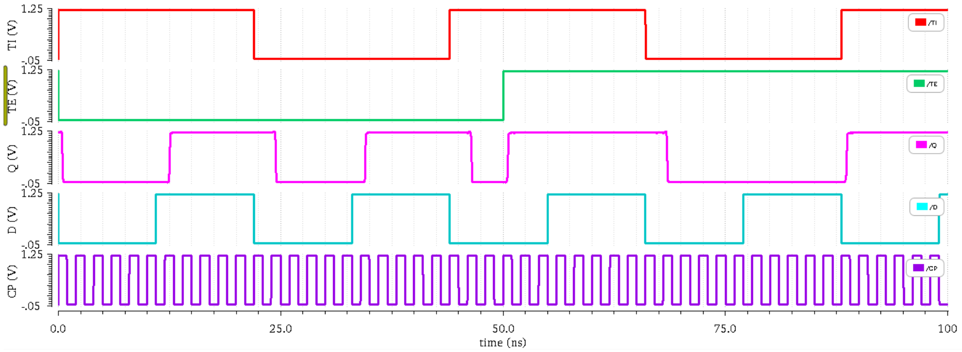


Figure 6.3: Post layout timing diagrams at clock frequency of $500 Mhz$

Launch-on-capture test

In launch-on-capture testing, upon the loading of the first test pattern into scan chain, the initialization vector is launched from the scan cells and response is captured in all the scan cells using functional master latches. After the first capture operation (transition launch) the transition responses are captured in all the scan cells using the functional master latch. The loading/unloading can be done in the same way as explained in the stuck-at faults application section.

Launch-on-shift test

In launch-on-shift testing, upon the loading of the test pattern into the scan chain, initialization vector is launched from the scan cells, however as the functional master latch get enabled only after the launch of second transition, no capture happens. As the second vector is a one bit shift of the first vector we need not to take care of the first vector's response. After the second vector (transition launch) is launched from the scan cells, the response is captured in all scan cells using functional master latch. Again the unloading/loading can be done in the same way as explained in the stuck-at faults application section. Thus, the proposed scan flip-flop retain both the test quality and the timing requirements intact. As the functionality of the proposed scan flip-flop is identical to the conventional scan flip-flop, it works perfectly with all type of the standard test flow. Furthermore, as the proposed scan flip-flop do not use any extra control signal.

6.1.3 Experimental Results

To validate the efficacy of the proposed scan flip-flop design, post layout timing simulation in both functional and test mode of operation have been carried out. The layout of both reference scan flip-flop [244] and proposed scan flip-flop was done using Cadence's Virtuoso layout editor suit using 65nm technology library. For the sake of fare comparison, transistors placement were done manually in standard cell fashion in an identical manner for both the implementations. After placement, Virtuoso's automatic routing function was used to do the interconnection routing. All the MOS transistors used were of minimum size with a $NMOS$ to $PMOS$ W/L ratio of 2, i.e. $(W/L)_p = 2 * (W/L)_n$. We did the parasitics extraction from the layout and simulated the circuit using Mentor Graphics's Calibre physical verification tool at clock frequency of 500MHz to 1GHz.

The timing simulation results of the proposed scan flip-flop at clock frequency of 500MHz have been put for reference in Figure 6.3. These results verify the functionality of the proposed design. As shown in Figure 6.3, when SE signal get to logic high (1) in test mode, the scan cell output node Q follows the input SI. When signal SE get to logic low (0), output node Q follows D. In functional mode, we have measured clock (CL) to Q (t_{cq}) propagation delay, setup time (t_{setup}) and hold time (t_{hold}) for both the implementations. As shown in Table 6.1, propagation delay t_{pd} of the proposed scan

Table 6.1: Post Layout Simulation Results at 500MHz

Parameter → *Scan cell* ↓	*Clock to Q* (t_{cq})	*Setup time* (t_{setup})	*Hold time* (t_{hold})	t_{pd} ($t_{cq} + t_{setup}$)	*Time gain* *w.r.t* [244]
Functional Mode					
Reference [244]	0.458ns	0.425ns	0.0ns	0.883ns	+51ps
Proposed	0.502ns	0.330ns	0.0ns	0.832ns	
Test Mode					
Reference [244]	0.458ns	0.425ns	0.0ns	0.883ns	−66ps
Proposed	0.509ns	0.440ns	0.0ns	0.949ns	

flip-flops in functional mode is found to be $0.832ns$. The propagation delay t_{pd} of the reference scan flip-flop is $0.883ns$, in functional mode. It can be observed from Table 6.1, that the clock to Q delay of the proposed flip-flop is higher than the clock to Q delay of the reference flip-flop. The higher clock to Q delay of proposed scan flip-flop is due to the use of a gated inverter at the output of the master latch that drives the input of the slave latch. On the other hand the setup time of proposed flip-flop is less than the setup time of reference flip-flop due to the elimination of input multiplexer in proposed design. Overall the proposed design offers a time saving of $51ps$. The time saving that proposed design offers is approximately equal to the propagation delay of $2-3$ inverters, where typical inverter delay in $65nm$ technology is approximately $15-18ps$.

In test mode, the proposed scan flip-flop has higher clock to Q delay (t_{cq}) and setup time with respect to reference scan flip-flop. The reason for higher (t_{cq}) and (t_{setup}) is the extra transmission gate used to isolate the shadow master latch from the slave latch during functional mode. Overall the performance of the proposed scan flip-flop degrades by approximately $66ps$ in test mode. As it is a common practice in industry to run the test application at a frequency less than half of the functional frequency, so the increase in propagation delay in test mode should not be of any concern and is acceptable. Also, if we consider the number of transistor as a rough metric of area estimation, then the proposed scan flip-flop uses two extra transistor as compared to reference scan flip-flop in [244]. The proposed scan flip-flop uses 36 transistors against the reference scan flip-flop which uses 34 transistors. Therefore, the proposed design has approximately 6% area overhead as compared to the reference scan flip-flop. This area overhead could be minimized by using this flip-flop only for critical paths.

6.2 Scan Cell Design for Test Power Minimization

Power dissipation during scan testing of modern high complexity designs could be many folds higher than the functional operation power, which is a well established observation. High test power dissipation can severely affect the chip yield and hence the final cost of the product. This makes it of utmost important to develop low power scan test methodologies. It is reported by Wunderlich et al. that around $70 - 80\%$ of scan shift power is dissipated in combinational logic alone [18, 76, 118]. Thus it is very important to eliminate useless power dissipation in combinational logic during scan shifting. Several techniques have been proposed to reduce the combinational switching activity in a circuit-under-test (CUT) during test.

Another very effective structural technique is to make use of blocking logic at the output of each scan flip-flop which prohibits the propagation of scan ripple values into combinational logic. This technique completely eliminates the redundant switching in combinational logic during entire scan operation. In [76], the authors proposed a $NAND$ or NOR gate as blocking logic, and freezes the combinational inputs to logic 1 or logic 0 during the scan shift using scan enable signal. The gating logic is very simple and test vector independent. However, the blocking logic comes into functional path and deteriorates the timing performance of the circuit. Devanathan et al. use a transmission gate to block the scan ripple propagation [61]. A pull-up or pull-down logic is used to keep the combinational inputs at constant logic 1 or logic 0 values.

In another gating technique the authors have proposed multiplexer as a gating logic [79]. The multiplexer blocks the Q output and holds the combinational inputs at previous states of scan flip-flop. Both these techniques suffer from functional performance degradation. Also the blocking logic consumes significant amount of power in functional mode. To overcome the performance degradation problem Kavousianos et al., and Elshoukry et al. have proposed partial gating [71, 110]. The authors in this technique use blocking logic only for those flip-flops which do not falls into timing critical paths. The partial gating can reduce power to a desired level. However, due to large fan-out cons of non gated flip-flops the power dissipation could be significantly large in some cases.

Bhunia et al. [30] use a different gating approach. The authors implement the gating effect by inserting an extra transistor in VDD to GND supply path for the first level gates at the output's of scan flip-flop. In recent years, researchers have proposed many methodologies to design low power scan flip-flop [185, 203, 248]. In one such proposals by Mishra et al., an extra transmission gate has been used to isolate the slave latch during scan shift operation [138]. The authors in this technique bypass the slave latch and use a dynamic slave latch to propagate the scan values. Extra circuitry is required to generate the control signals to make it work for both LOS and LOC based *at-speed* test. This technique effectively reduces the combinational switching, however, it is very costly in terms of both area and performance overhead.

We propose a modified scan flip-flop design which uses a low cost dynamic slave latch to shift the test vectors and allows the static slave latch to retain the responses from the previous test vector. Through bypassing the slave latch during loading/unloading operation the proposed design eliminates redundant switching activity in combinational logic and hence minimizes test power. Furthermore the proposed scan flip flop design does not use any gating element in functional path, and hence the functional performance overhead is comparatively very less than the previously proposed output gating techniques so far.

6.2.1 Schematic Design of Proposed Scan Cell

This section explains the proposed scan flip-flop design and it's functionality in different modes of operation. The schematic design of the proposed scan flip-flop is shown in Figure 6.4. As compared to conventional scan flip-flop shown in Figure 6.1, the proposed scan flip-flop uses separate slave latches for functional mode and test mode. However, for both modes of operation the proposed scan flip-flop uses only a single master latch. The master latch is formed by transmission gate $T1$ and the inverter pair connected back to back through transmission gate $T2$.

The slave latch is formed by transmission gate $T3$ and the inverter pair connected back to back through transmission gate $T4$, and is referred as functional slave latch. The second slave latch, which is a low cost dynamic slave latch, is formed by transmission gate $T5$ followed by an inverter which is connected between $T5$ and scan output node SO, and is referred as dynamic slave latch. The proposed scan flip-flop uses different clock signals to enable/disable the master latch and the functional slave latch. However,

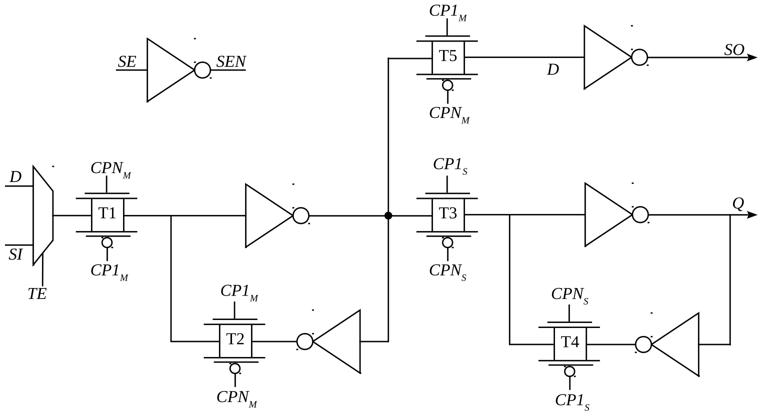


Figure 6.4: Proposed scan flip-flop schematic design

it should be noted that these clock signals are locally derived from the same global clock signal CP and hence they are in synchronization with each other. The transmission gate of master latch and dynamic slave latch i.e. $T1$, $T2$ and $T5$ are controlled by clock signals $CP1_M$ and CPN_M. The transmission gate $T3$ and $T4$, of functional slave latch, are controlled by $CP1_S$, and CPN_S.

All the clock signals, $CP1_M$, CPN_M, $CP1_S$, and CPN_S, are locally derived signals from the global clock signal CP. In functional mode, $CP1_M$, and CPN_M are equivalent to $CP1_S$, and CPN_S respectively. However, in test mode $CP1_S$ and CPN_S remains at constant logic high (1) and logic low (0) levels respectively. The $CP1_M$ and CPN_M clock signal function in normal way in both functional and test mode. The circuit to locally derive these clock signals is shown in Figure 6.5.

The proposed scan flip-flop has two outputs Q and SO. The output Q drives the combinational inputs and output node SO is used for scan shifting. In test mode the functional slave latch is disabled and the shifting of test vector takes place through the dynamic latch. In functional mode the functional slave latch is enabled and the proposed scan flip-flop functions like a conventional scan flip-flop. The detailed operation of the proposed scan flip-flop in functional mode as well as in test mode is discussed in the following subsections:

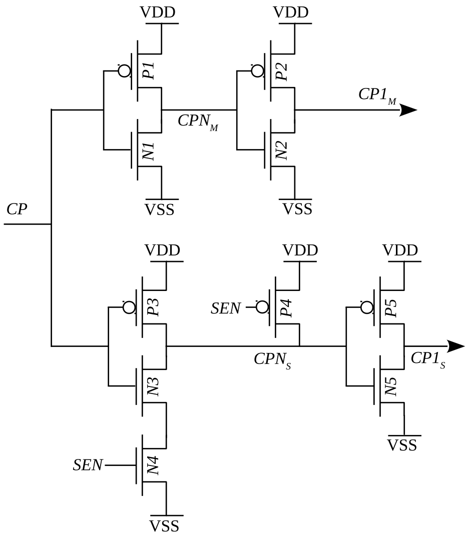


Figure 6.5: Circuit to generate locally derived clock signals

Functional mode operation

The *test_enable (TE)* signal switches the scan flip-flop between functional mode and test mode. In test mode, the *TE* remains at logic high (1), and in functional mode *TE* remains at logic low (0). In functional mode, the *TE* signal gets to logic low (0) and *TEN* signal gets to logic high (1). As it can be observed in Figure 6.5, the pullup *PMOS* transistor *P4* turns *OFF* and the *NMOS* transistor *N4* turns *ON*. This makes master latch clock signal equivalent to functional slave latch clock signal. $CP1_M$, and CPN_M, are in synchronization with $CP1_S$, and CPN_S respectively. The *functional_input (D)* is selected and propagates into the master latch when clock is low (0). At the same time the slave latch is disabled and retains previous value. When the clock signal turns high (1), the value stored into master propagates into the functional slave latch and to the output node *Q*.

Test mode operation

When clock signal *(CP)* is high, a rising transition on *TE* signal switches the scan flip-flop from functional mode to test mode. As *TE* gets to logic high (1), the pullup *PMOS* transistor $P4$ turns *ON* and *NMOS* transistor $N4$ turns *OFF*. This inhibits the functional slave latch clock signal CPN_S and $CP1_S$ to toggle, and freeze them at constant logic high (1) and logic low (0) levels respectively. This keeps the transmission gate $T3$ disabled, as long as the circuit is in test mode, and disables the functional slave latch. Since the functional slave latch remains disabled during the whole scan shift operation there is no redundant switching in the functional logic. Meanwhile, the low cost dynamic slave latch is used to serially shift the test vectors. When the clock gets high, the transmission gate $T5$ is enabled, and the value latched into the master latch start driving the input parasitic capacitance (at node D) of the dynamic latch inverter. When the clock gets to logic low, the transmission gate $T5$ is disabled and the parasitic capacitance at node D drives the next scan flip-flop's *test_input* (*SI*). Due to high impedance of the inverter, the parasitic capacitance do not discharge immediately. The parasitic capacitance discharge time puts a lower bound on the frequency of shift operation. However a lower shift frequency results into higher test time and higher test cost. Therefore, a very low scan shift frequency is undesirable. The major advantages of the proposed design in terms of test power and functional timing performance are as follows:

1. When the proposed scan flip-flop switches from capture mode to scan or shift mode the functional slave latch retains the test response value of the previous test vector. This completely eliminates the redundant switching in the combinational logic. The additional advantage of the proposed design is that it does not settle to some particular value in the cycle following the capture cycle which is the case in previously proposed output gating techniques.

2. Also there is a reduction in switching activity in the scan flip-flop itself, as the functional slave latch remains disabled during scan shift operation. However, the dynamic slave latch inverter dissipate some additional power. Since this inverter drives only the *test_input (SI)* of next scan flip-flop's multiplexer, it could be made weak. As a whole the power dissipation in the proposed scan flip-flop will be much lower as compared to other output gating techniques.

3. The test frequency is decided by the maximum power dissipation and *test_enable*

(TE) signal routing delay. In designs where the test frequency is constrained by former, the proposed design offers a significant reduction in power reduction. In that case the test frequency can be increased significantly by properly routing the *test_enable (TE)* signal.

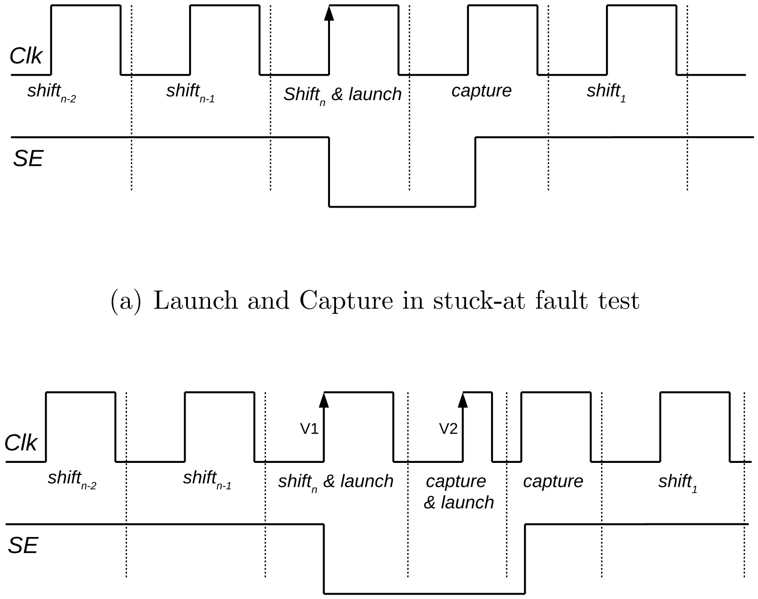
4. The proposed design does not use any gating logic that comes into the functional path. Hence the functional performance degradation is very less as compared to previously proposed output gating techniques, which suffers from the functional path timing degradation.

6.2.2 Test Quality Consideration

The proposed scan flip-flop inherits all the test application capabilities of a conventional scan flip-flop. The scan chain integrity is ensured by applying the flush test. As no capture is required in scan flush test, these patterns are shifted through the dynamic slave latch. This covers all the faults on the scan path. However, the scan path do not traverse through the functional slave latch. The faults on the input and output of the functional slave latch are not covered by the scan chain integrity test. These faults can be covered in *stuck-at* and *at-speed* test. The application of these tests is discussed in the following subsections:

Stuck-at fault test

During scan shift operation the *test_enable (TE)* signal is kept at logic high (1) level. The timing diagram for *stuck-at* fault testing is shown in Figure 6.6(a). While the shifting of test patterns takes place, the functional slave latch remains disabled. In the last *shift-cum-launch* cycle, when the clock is low (0), the test values are loaded into the master latch. When the clock gets to high (1), *TE* should get down to low. This enables the functional slave latch. The test value stored into the master latch gets propagated to the functional slave latch and to the output of the scan flip-flop. This completes the test vector launch operation. The *test_enable* in general is not a timing closed signal.



(a) Launch and Capture in stuck-at fault test

(b) Launch and Capture in LOC testing

Figure 6.6: Timing requirements for scan enable in *stuck-at* and *LOC* test

The *test_enable* should get down within half clock cycle. To ensure the proper launch through the functional slave latch the clock frequency must be adjusted only for launch and capture cycles as follows:

$t_{clk} \geq t_{test\_enable} + t_{cq} + t_{comb} + t_{setup}$

where, $t_{clk} = clock\ period$

$t_{test\_enable} = test\ enable\ routing\ delay$

$t_{comb} = combinational\ path\ delay$

$t_{cq} = clock\ to\ q\ delay$

$t_{setup} = setup\ time$

$t_{test\_enable} \leq t_{clk}/2$

However, this does not put any constraint on the shift frequency. Instead of adjusting the clock cycle for launch and capture cycle, the clock can be kept high for a few cycles after the arrival of positive edge (i.e., when clock gets to high (1)) of launch cycle and after the positive edge of capture cycle. This is a common practice in industry and is known as dead cycle insertion. However, the dead cycles are inserted before the launch cycle and after the capture cycle. During this period the clock is kept low instead of high. This provides sufficient time for the *TE* signal to change from one level to other.

Following the launch cycle, when clock gets low (0), the response is captured into the master. When the clock gets high (1), the responses captured into master gets transferred to the dynamic as well as functional slave latch. Following the capture cycle, when the next shift cycle comes to load the next test vector, the response stored into the dynamic slave latch will propagate to the master latch of the next scan flip-flop and to the output of the scan flip-flop. As stated earlier, the scan chain integrity test does not cover input output faults of the functional slave latch. The *stuck-at* fault test covers the functional slave latch input/output faults. Since the test vector is launched from the functional slave latch, such fault will manifest themselves at the end of response unloading process.

Launch-on-capture test

The timing or transition delay faults are tested by applying a test vector pair(V_1, V_2). The first vector V_1 initializes the circuit and the second vector V_2 launch a transition. In *launch-on-capture* (*LOC*) transition delay fault test, the initialization vector V_1 is loaded and launched into similar manner as *stuck-at* vector. The combinational response of vector V_1, which also acts as vector V_2, is captured keeping the *TE* signal low. The response of V_2 is captured *at-speed* or functional frequency. After the *at-speed* capture, *TE* is pulled up high (1) again to apply the next test. The timing diagram for *launch-on-capture* is shown in Figure 6.6(b).

Launch-on-shift test

In *launch-on-shift* (*LOS*) transition delay fault test, vector V_2 is a one bit shift of vector V_1. In order to apply *LOS* test, the *test_enable* (*TE*) signal must be timing closed. Since the *TE* signal is also a global signal just like clock. To make *TE* signal timing closed is very costly task. In general, the *LOS* test is not exercised into industry due to its very high implementation cost. To apply *LOS* test even in conventional scan flip-flop without fast *TE* signal is not possible. *LOS* test without fast *TE* signal can be exercised by using extra *AOI* (*AND-OR-INVERTER*) circuitry proposed by Gefu et al. [240]. Therefore, both conventional as well as the proposed scan flip-flop lack the capability to apply *LOS* based transition delay fault test. The proposed scan-flip-flop does not impose any extra testing constraints and works perfectly well with the standard test flow. The proposed design neither uses any extra control signal nor it has special test application and timing requirements. It can be easily integrated into the existing industrial test flow.

6.2.3 Experimental Results

Post layout timing simulation of the proposed scan flip-flop has been carried out. The layout of both conventional scan flip-flop and proposed scan flip-flop has been done using Cadence's Virtuoso physical layout suit. The technology library used for post layout simulation was of 65nm technology. Schematic driven placement of transistors was done in standard cell fashion in an identical manner for both the implementations. For the sake of fair comparison, the routing was done using Virtuoso's automatic routing feature. The sizes for all the *MOS* transistors were kept minimum with a *NMOS* to *PMOS* W/L

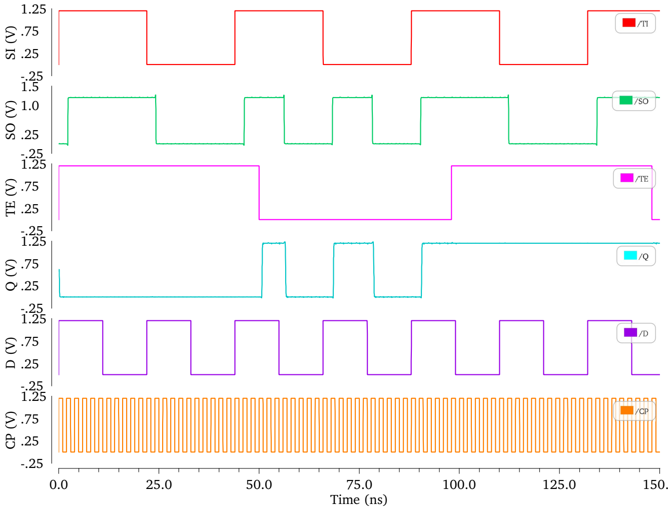


Figure 6.7: Post layout timing waveform at 500MHz

Table 6.2: Post Layout Simulation Results at 500MHz

Functional Mode				
Parameter → Scan cell ↓	*Clk to Q* (t_{cq})	*Setup Time* (t_{setup})	t_{pd} ($t_{cq} + t_{setup}$)	*time Overhead* *w.r.t conventional* [79]
Conventional [79]	0.332ns	0.400ns	0.732ns	−−
Proposed	0.380ns	0.420ns	0.800ns	9.2%
Mishra et al. [138]	−−	−−	−−	24.6%
Test Mode				
Conventional [79]	0.332ns	0.400ns	0.732ns	−−
Proposed	0.320ns	0.460ns	0.780ns	6.5%

ratio of 2, i.e. $(W/L)_p = 2 \times (W/L)_n$. After extracting the parasitics, timing simulation was done using Mentor Graphics's Calibre physical verification tool at clock frequencies of upto 1GHZ.

The post layout timing simulation results at a clock frequency of 500MHz are shown in Figure 6.7. The timing diagram verifies the functionality of the proposed scan flip-flop design. As it can be observed in Figure 6.7, when the flip-flop is in test mode, the functional slave latch output (Q) freezes at the last value latched in functional mode. The timing waveforms also verifies the proper scan shifting of test vectors through the dynamic slave latch. When $TE = 1$, the scan output (SO) of the dynamic slave latch follows the *test_input* (SI). When the *test_enable* signal $TE = 0$, the proposed scan flip-flop operates in functional mode. During functional mode both the functional slave latch output Q and dynamic slave latch output SO follow the functional input D.

The post layout timing measurement results for the proposed and conventional implementations are shown in Table 6.2. The functional propagation delay t_{pd} of the proposed design and the conventional design are found to be 800ps and 732ps respectively. This minor increase in propagation delay is due to the extra capacitive loading on the master latch caused by the dynamic slave latch input capacitance. However, the overall 9.2% increase in functional mode propagation delay is much smaller compared to 24% increase as reported by Mishra et al. [138]. In test mode, the propagation delay is found to be 780*ps*, approximately 6.5% higher. This decrease in propagation delay in test mode is due to the disabling of the functional slave latch, which decrease the capacitive load on the master latch output. As a rough metrics of area estimation, the proposed design uses five extra inverters and an extra transmission gate. As compared to [138] the proposed design has very less area and timing overhead. Because, in [138] the authors use a dynamic slave latch along with an extra global slave disable signal. Therefore, with a minimal area overhead, the proposed design effectively eliminates redundant test power dissipation, without compromising much on functional performance.

6.3 Enabling *LOS* Test with Slow Scan Enable

The probability of delay defects has increased in deep sub-nanometer technology due to process variation effects. Such defects can be detected using *Launch-off-Capture* (*LOC*) and *Launch-off-Shift* (*LOS*) based delay test techniques. In terms of delay test coverage and test set size *LOS* is more effective compared to *LOC*. However, to exercise *LOS* based delay test a high speed *scan_enable* signal is required. The cost of implementing a high speed global *scan_enable* signal is prohibitively high. In practice, most of the commercial design employing full scan design supports only *LOC* based delay test. We propose a new scan flip-flop design that is capable of exercising both *LOS* and *LOC* based delay test with a slow *scan_enable* signal. The proposed design can achieve much higher delay fault coverage by exercising both *LOS* and *LOC* test.

6.3.1 Introduction

Timing related failures have become more common in complex *VLSI* chips fabricated in deep *sub-nanometer* technology. In order to ensure shipment of high quality level chips at-speed testing has become very important. In the past, functional tests were used to ensure the correct *at-speed* operation of high performance chips. However, with ever growing design complexity functional test has become unpractical in terms of test set size and timing fault coverage. Presently, scan based *TDF* testing is the only practical solution. As the delay defects can only be activated and observed by passing a signal transition through the fault site, a test vector pair $\langle V_1, V_2 \rangle$ is required. The first vector V_1 is scanned-in into the scan chain and used to initialize the circuit under test *CUT* to a particular state and is called initialization vector. Once the *CUT* is initialized to a known state, second vector V_2 is used to launch a transition at the target node and is therefore called launch vector. The transition must reach one of the observable circuit outputs within the functional timing specification. Due to the design limitation of scan chain not all combinations of $\langle V_1, V_2 \rangle$ are possible. Depending upon how transition vector V_2 is generated the delay fault test technique is categorized as *LOC* or *LOS*. In the case of *LOC*, the second vector V_2 is functional response of initialization vector V_1. The functional dependency puts a restriction on possible $\langle V_1, V_2 \rangle$ combinations. This results in a moderate delay fault coverage for *LOC* technique. Moreover, the *LOC* requires a complex sequential *ATPG* because it needs to consider two sequential states to generate valid $\langle V_1, V_2 \rangle$ pair.

In *LOS* the second vector V_2 is a one bit shift over V_1. The constraints to generate valid $\langle V_1, V_2 \rangle$ test vector pair in case of *LOS* is bit relaxed. Due to which *LOS* has comparatively higher test coverage and significantly smaller test set size. However, to exercise *LOS* test the *scan_enable* signal must be timing closed and need to be routed like a clock. The design cost associated with a timing critical global *scan_enable* signal is prohibitively high. Despite having significantly higher delay test coverage and much smaller test set size most scan based designs do not employ *LOS*. Consequently, there is a growing need for low cost scan designs that can support *LOS* based delay test.

Several solutions have been proposed to support *LOS* delay test without using high speed *scan_enable* signal. In one such approach scan enable signal is distributed in a pipeline manner [16]. This technique eliminates the requirement for a fast *scan_enable* signal. However, the local *scan_enable* signal still needs to be timing closed. Furthermore, the routing of these fast local timing critical signals complicates the layout. In another approach, Prasanna et al. [62] avoids the use of a fast *scan_enable* by operating a subset of flip-flops in LOS mode during launch and capture operation. This technique allows LOS for a subset of flip-flops by compromising the observability of test response for those flip-flops. This approach increases the TDF coverage compared to LOC test alone. However, the coverage remains well below the LOS test. Furthermore, the technique has an overhead of an extra slow speed global *scan_enable* signal.

In Hybrid Delay Scan [223] the observability of the subset of flip-flops is restored by using a fast *scan_enable* signal which operates in LOS mode. The rest of the flip-flops are operated in LOC mode. This improves the coverage at the expense of distributing the fast *scan_enable* signal for the subset of flip-flops. Enhanced scan is another classical scan delay test technique which removes the restriction on the $\langle V_1, V_2 \rangle$ combinations [62, 223]. There are various variants of enhanced scan available in the literature, however, the area overhead of all these designs are prohibitively very high. In a recent work, Gefu et al. [240] proposed a new Delay Test Scan flip-flop ($DTSFF$). The $DTSFF$ uses a local clock alignment logic to properly align the slow *scan_enable* signal to the clock edge during the launch cycle to support LOS tests. The $DTSFF$ can effectively apply LOS test however it can not be used to apply LOC based test. In a mixed LOC/LOS test the delay fault coverage can be further improved upto 95% [240]. To apply mixed LOC/LOS, full LOC and full LOS tests the $DTSFF$ uses two extra scan test select signals. The area overhead in such scenario could be very high.

An alternative DfT approaches to serial scan is RAS which is highly efficient in terms of test power and test time. However, this technique also lacks LOS based delay test capability [23, 210]. We propose a new scan flip-flop design that can be used to implement both full LOS or full LOC tests using a slow scan_enable signal. The major advantages of the proposed scan flip-flop design are as follows:

1. The proposed design eliminates the need for a fast scan enable signal to apply LOS.

2. The proposed design does not use any extra control signal and has a minimal area overhead compared to conventional scan flip-flop.

3. The new scan flip-flop is capable of applying both LOS and LOC tests.

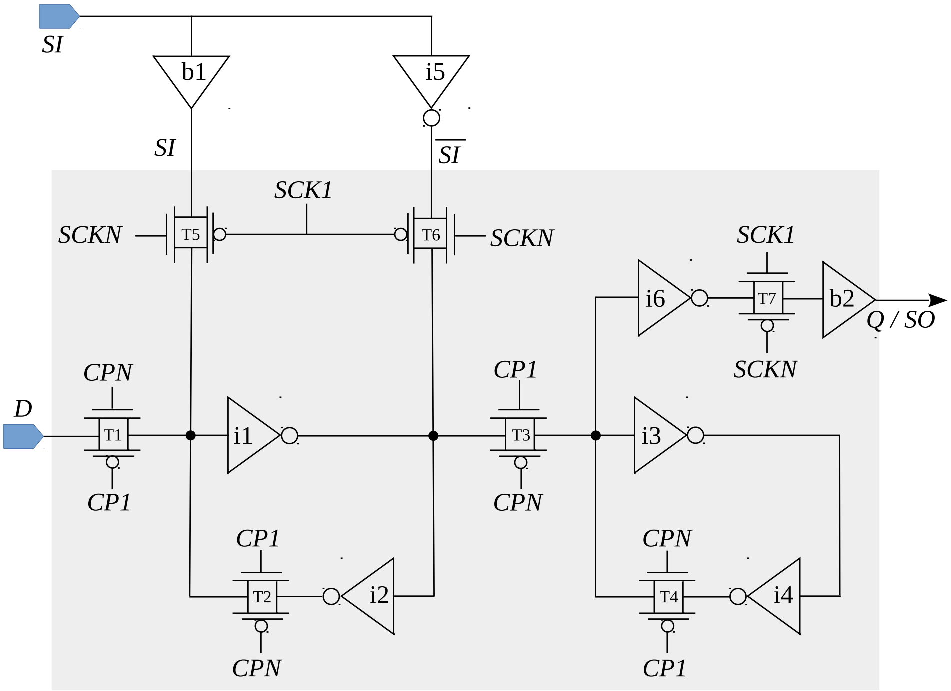


Figure 6.8: Proposed scan flip-flop design

The next subsection describe the proposed scan flip-flop design and elaborates on the details of test application process respectively.

6.3.2 Proposed Scan Flip-Flop Design

The schematic design of the proposed scan flip-flop is shown in Figure 6.8. It uses separate functional input (D) and test input (SI) paths. Transmission gate $T1, T2$, and back to back connected inverter pair ($i1, i2$) form the master latch. The slave latch is formed by transmission gate $T3, T4, T7$, and back to back connected inverter pair ($i3, i4$) along with inverter $i6$ and buffer $b2$. Transmission gate $T5$, $T6$, buffer $b1$, and inverter $i5$ form the test input path. It can be observed that the extra gates used to form the test input path are not on the functional path. During functional mode of operation, the test input path is disabled by the *scan_enable* cum scan clock signal SCK. The operational details of the proposed design are explained in the following subsections:

Functional mode

In functional mode of operation the proposed scan flip-flop works as a regular flip-flop. During functional mode, the scan clock signal SCK is permanently held at logic high (1) level. This makes transmission gate $T5$, and $T6$ OFF and disconnects the test mode input path from master structure. On the other hand a constant high (1) SCK signal keeps the transmission gate $T7$ always ON during functional mode. The value of functional input D gets latched into the master latch when clock CP is low and propagates to slave latch when CP turns high. The transmission gate $T7$ followed by buffer $b2$ falls into the functional path and increase the clock to Q delay of the scan flip-flop. However, it should be noted that the proposed design does not have a scan multiplexer on its functional path. This decreases the setup time of the proposed design. Hence the increase in the clock to Q delay is compensated by the decrease in setup time. In overall the proposed scan flip-flop has same functional mode timing performance compared to conventional scan flip-flop. The post layout timing simulation results given in the experimental section validates the functional performance of the proposed scan flip-flop.

Test mode

In test mode the functional clock CP is held at a constant logic high (1) level. This makes transmission gate $T1$ OFF and disables the functional input D during the test mode. Also, it should be noted that transmission gate $T3$ stays ON during test mode and became a part of the master latch along with inverter $i6$. The test mode slave latch is formed by transmission gate $T7$ and output buffer $b2$. The test mode slave latch is dynamic in nature as the feedback path transmission gate $T4$ remains OFF during scan shift mode. In test mode, when scan clock SCK is at logic 0, transmission gate $T5$ and $T6$ turn ON and the test value SI is written into the master latch. The test input value is written into the master latch via buffer $b1$ and inverter $i1$ simultaneously in a memory write operation fashion. As the scan clock signal SCK turns high (1), transmission gate $T7$ turns ON and the value stored in master latch gets transferred to the dynamic slave latch.

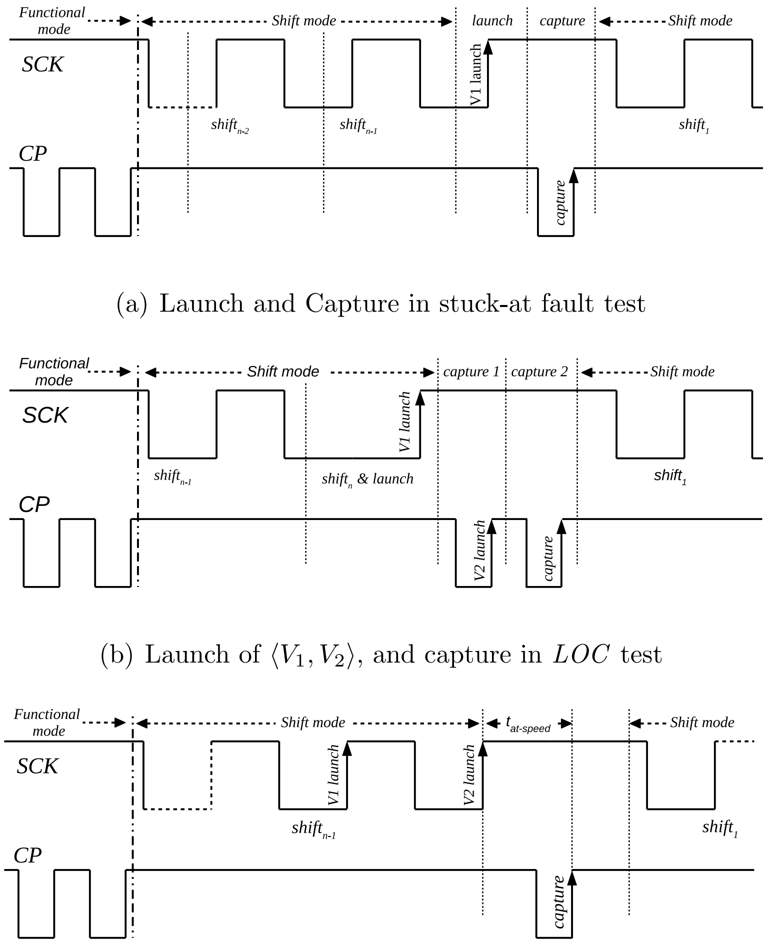
During the time when SCK is at logic low level (0) and $T7$ is OFF, the input parasitic capacitance of output buffer $b2$ keeps holding the test value into the dynamic slave latch and drives the output Q/SO of the scan flip-flop. Due to a very high impedance of the buffer, it takes a long time for the parasitic capacitance to get discharged. The discharge time puts a minimum frequency at which the scan shift operation can be performed. However, as the shift frequency decides the test time and hence test cost, a very low shift frequency is highly undesirable. Application of static (*stuck-at-fault*) and timing (*transition delay fault*) tests are elaborated in detail in the next section.

6.3.3 Application of Test Vectors

The main advantage of the proposed scan flip-flop over conventional scan flip-flop is that it is capable of applying LOS timing test with slow scan enable signal. Before exercising static or timing test, the scan chain integrity is verified by scan flush test. Keeping the functional clock CP high (1), an all transition pattern (1100) is shifted through the scan chain by repetitive application of scan clock signal SCK. The scan chain integrity is verified by observing the flush test pattern at the primary output pin. Application of static and timing test are explained in detail in the next subsections.

Stuck-at-fault test

The timing diagram for *stuck-at-fault* test is shown in Figure 6.9(a). First, the functional clock is pulled to logic high (1) level. This disables the functional input D. The falling edge on the scan clock SCK enables the test input path. The shifting and launching of test vector are done by successive application of scan clock SCK. Once the test vector is launched the scan clock SCK is kept at logic high (1) level. The response is captured by applying the functional clock CP once. The functional clock CP is kept high after capturing the response. The unloading of test response is done with simultaneous loading of next test vector.



(a) Launch and Capture in stuck-at fault test

(b) Launch of $\langle V_1, V_2 \rangle$, and capture in *LOC* test

(c) Launch of $\langle V_1, V_2 \rangle$, and capture in *LOS* test

Figure 6.9: Timing requirements for scan enable in *stuck-at* and delay test

Launch-off-Capture test

The initialization vector V_1 is shifted in and launched using the slow scan clock signal *SCK* in a way similar to *stuck-at-fault* test. The functional response of vector V_1 is captured by applying functional clock *CP* once which also acts as the launch of transition vector V_2. The functional response of transition vector V_2 is captured by applying an *at-speed* functional clock *CP* pulse. The timing diagram shown in Figure 6.9(b) illustrates the timing requirements associated with the execution of *LOC* test. The loading/unloading of test stimuli/response is again done using the slow scan clock *SCK*.

Launch-off-Shift test

The proposed design has the capability to apply *LOS* based delay test using a slow *scan_enable*. The initialization vector V_1 is shifted in and launched using the slow *scan_enable* cum scan clock *SCK* in a way similar to *LOC* delay test. The transition vector V_2, which is a one bit shift over V_1, is also launched using slow scan clock *SCK*. The clock to Q delay $t_{cq(SCK)}$ of *SCK* will be different as compared to clock to Q delay $t_{cq(CP)}$ of functional clock *CP*. This difference is because of two reasons. First, since *SCK* is nothing but the slow *scan_enable* signal so, it has higher rise and fall time. Second, in the case of scan clock *SCK*, the launch takes place from transmission gate $T7$ wherein in case of functional clock *CP* the launch takes place from transmission gate $T3$. In functional clock to Q ($t_{cq(CP)}$) path there is an extra inverter ($i6$) delay compared to test clock to Q ($t_{cq(SCK)}$) path. This difference in t_{cq} needs to be considered while applying the *at-speed* functional clock *CP* to capture the functional response of V_2. The timing requirement associated with the execution of *LOS* test is shown in Figure 6.9(c). For proper *at-speed* response capture the following timing requirements must be met:

$$t_{at-speed} = t_{cq(CP)} - \Delta + t_{comb} + t_{setup(CP)}, where$$

$$t_{setup(CP)} = functional\ mode\ setup\ time$$

$$t_{comb} = combinational\ delay$$

$$\Delta = t_{cq(CP)} - t_{cq(SCK)}$$

As can be seen from Table 6.3, the $t_{cq(CP)}$ is $20ps$ larger than $t_{cq(SCK)}$. As we know that the functional clock *CP* is a free running clock originated from on-chip *PLL* so, the position of the capture edge can not be changed. However, the slow scan clock *SCK* is provided by the Automatic Test Equipment (*ATE*). Hence, the V_2 launch edge of *SCK* can be easily positioned with respect to capture edge of *CP*. So, *at-speed* capture timing requirements given by the above equation can easily be met. Alternatively, the positioning can be done by inserting an on-chip delay element in input path of *SCK*.

Table 6.3: Post Layout Timing Simulation Results at 500MHz

Proposed Scan Flip-flop					
Parameter → Mode ↓	*Clk to Q* (t_{cq})	*Setup time* (t_{setup})	*Hold time* (t_{hold})	t_{pd} ($t_{cq} + t_{setup}$)	*Time gain w.r.t.* *Conventional*
Functional	0.100ns	0.022ns	0.0ns	0.122ns	$+1ps$
Test	0.080ns	0.207ns	0.0ns	0.287ns	$-164ps$
Conventional Scan flip-flop					
Functional	0.058ns	0.065ns	0.0ns	0.123ns	$--$

6.3.4 Experimental Results

To verify the efficacy of the proposed design post layout timing simulation has been carried out using *UMC's* 65nm technology at operating voltage of 1.2V for frequencies ranging from 500MHz to 1GHz. The post layout timing simulation results are listed in Table 6.3. It can be observed that in functional mode propagation delay t_{pd} of proposed flip-flop is almost same as t_{pd} of the conventional flip-flop. In test mode, (t_{pd}) degrades by approximately 164ps. However, test mode performance degradation is not of any concern as the scan shifting is done at a much lower frequency.

6.4 Conclusion

In this chapter we have proposed scan cell design based techniques to overcome three issues related to scan design. First, we have proposed a new transistor level design of scan flip-flop which can significantly enhance the functional performance of scan based designs by eliminating the performance penalty of scan. The scan cell can be utilized to improve the speed of performance critical circuits. Second, we have proposed a scan flip-flop design which can exercise *LOS* based delay test with a slow scan enable signal. The new scan flip-flop can achieve significantly higher delay fault coverage by applying both *LOS* and *LOC* based delay test. The proposed design have a minimal area overhead in terms of transistors count compared to existing solutions. Finally, we proposed a scan flip-flop design which suppresses the redundant/useless switching in functional logic during scan shift operation. The proposed design also reduces power dissipation inside the flip-flop itself by disabling the slave latch during serial scan of test vectors. The proposed scan flip-flop provides a way of combinational switching suppression with comparatively very less functional performance degradation compared to existing output gating techniques. All the three proposed techniques fully complies with the conventional scan design and test flow, and can be easily integrated into the existing Design-for-Test (DfT) flow. Furthermore, the proposed designs has the capability of exercising all conventionally used structural tests.

$- * - * -$

Chapter 7

Conclusion and Future Scope

The advancements in chip fabrication has made it possible to design systems with highly complex functionality, which are being used in applications such as autonomous vehicles, personnel healthcare, smart home and cities, deep neural network, and many more *IoT* based emerging applications. Security and reliability are at the forefront of design of such systems. In this thesis, we studied security and testability issues in modern day *VLSI* chips. Because of orthogonal objectives of security and testability the problem has become more severe.

In this thesis, we have proposed techniques to perform scan test of cryptographic chips in a secure manner. The proposed techniques ensures security of scan architecture against scan attacks without compromising on its testability aspects. Another problem that we addressed in this thesis is the testability issues in scan test such as test data volume, test time, and test power. We have proposed an efficient implementation of an alternative Joint-scan architecture that minimizes these issues all togather.

Further, we have explored scan cell design based approach to resolve issues like, scan performance overhead, unnecessery switching activity in combinational logic during scan, and scan chain diagnosis. In addition to that we proposed a technique to enable *LOS* based delay test with slow scan enable signal.

7.1 Contributions

Chapter 3 presented test protocol countermeasure based techniques to secure the scan design of cryptographic chip. The proposed techniques are based on encryption key masking, test restriction and test data encryption. The proposed techniques secure the scan design against all the know scan-based side-channel attacks on *AES*. The effectiveness of the proposed techniques is evaluated in terms of security, testability, and design cost. We synthesized the scan inserted *AES* architecture with proposed scan security features using 65*nm* technology nodes for area estimation. All of our proposed techniques are comparatively area efficient compared with their existing counterparts. The key points about the proposed techniques are given below:

∗ the first technique masks the encryption key during scan test

  - an *all-0* or *all-1* key is used as pseudo key to carry out test

∗ the second technique is based on test restriction principle

  - *LFSR* and *MISR* based test authorization schemes are used to restrict access to scan architecture; uses separate test key for authorization;
  - plain-text restriction during scan test is used to restrict attacker to apply differential input pairs; no separate test key needed; no test time and data volume overhead; supports all kinds of scan test and no loss in fault coverage

∗ the third proposal is based on test data encryption

  - on-chip test vector encryption to restrict the attacker for applying crafted input vectors
  - test response encryption using pipelines *AES* avoids test response analysis by attacker; no separate test key needed, provides $2X$ higher throughput compared to the iterative *AES*

Chapter 4 contributed a framework for a new *Joint-scan DfT* architecture. We named it as *2M-JScan* because it functions in just two modes of operation unlike the earlier implementation of *Joint-scan* which operates in four modes. The proposed architecture is shown to be effective in minimizing the test time, test data volume, and test power compared to the *MSS* and *PRAS*. Results show up to 50% reduction in test time compared to the *PRAS* and the *MSS* for sufficient number of test pins. Test data volume compared to the *PRAS* and the *MSS* is reduced by around $40-50\%$. Experimental results show 12% improvement on test time, compared to the previously proposed Joint-scan architecture, without affecting data volume and test power. Furthermore, we also have proposed a new scan cell design that can be used as a common scan cell in *2M-JScan* architecture wherein it can be used both as a serial scan cell as well as a Random Access Scan (*RAS*) cell. Contributions of this Chapter could be summarized as follows:

∗ Proposed the *2M-JScan* architecture

  – operates in two modes: test and functional

∗ Developed an efficient test control mechanism

  – maintains equilibrium in shift time across all patterns

  – reduced test pin count and minimized test time

∗ Provided design of a new scan cell for *Joint-scan* architecture

  – can be used both as serial scan cell as well as *RAS* cell

  – offers 64ps of time saving in functional mode

We implemented the *Joint-scan* test architecture using the proposed scan flip-flop. The effectiveness of the architecture is experimentally demonstrated on the modified (enlarged *SoC* designs) *ISCAS*89 circuits. The experimental results show a promising reduction in interconnect wire length. The maximum percentage reduction in total wire length is approximately 21% for $S38417$ benchmark circuit. The reduced interconnect wire length could help in overcoming the routing congestion which impedes practical implementation of *RAS* architecture.

Chapter 5 presented a hardware-assisted low cost and low complexity scan chain diagnosis technique. The proposed technique is very simple in operation and provides maximum resolution for stuck-at fault diagnosis. A custom scan cell with both set/reset capability is used to enhance the diagnostic capability. The major advantages of the proposed technique are as follows:

∗ Minimum area overhead compared to the existing hardware-assisted diagnosing techniques
  - uses only one global diagnose control signal

∗ Offers maximum diagnostic resolution for scan chain *stuck-at* faults
  - diagnostic resolution of 1 for both *stuck-at-0* and *stuck-at-1*
  - can locate the exact position of the faulty scan cell

∗ Can be extended to diagnose scan chain hold time faults at the cost of slightly diminished diagnostic resolution.
  - overall diagnostic resolution is $1 - 2$

∗ Modified scan cell design eliminates the need for separate set and reset control signal for diagnosis.
  - small performance overhead compared to conventional scan cell design

Chapter 6 addressed three issues of scan *DFT* architecture through scan cell design based approach. The scan issues being addressed are scan performance overhead, unnecessary switching activity in combinational logic during scan, *LOS* based delay test with slow scan enable signal. The following contributions may be noted:

∗ Elimination of Performance Penalty of Scan

  - removed scan multiplexer off the functional path
  - offers time saving of $55ns$ in functional mode
  - can be used in high speed circuit design

- Test power minimization in combinational logic
  - eliminates redundant switching activity in combinational logic during scan shift operation and hence minimizes test power
  - retain the responses from the previous test vector
  - functional performance overhead (9.2%) is comparatively very less than existing output gating techniques

- Enabling LOS based delay test with slow scan enable
  - capable of exercising both LOS and LOC based delay test with a slow scan enable signal
  - can achieve much higher delay fault coverage by exercising both LOS and LOC test
  - does not use any extra control signal and has a minimal area overhead compared to conventional scan flip-flop
  - no timing performance overhead compared to conventional scan cell

7.2 Future Scope

In this thesis, we addressed security and testability issues in scan *DfT* based testing of present day *VLSI* chips. This work can be extended in various directions. The future directions where this work could be taken forward is discussed in this Section.

7.2.1 Secure Scan DfT Architecture

The proposed secure scan *DfT* architecture is dedicated to test the cryptographic chips, implementing the Advanced Encryption Standard (*AES*), in a secure manner. However, various hardware implementation of public key ciphers like Elliptic Curve Cryptography (*ECC*) has been found vulnerable to scan attacks. The proposed techniques can be extended to securely test the *ECC* crypto chips.

The pipelined architecture based test response encryption scheme has the capability to perform reliable functional operation. It can be extended for detection of intermittent errors during functional mode of operation. Furthermore, this technique can be extended for secure testing of fully pipelined *AES* architecture.

7.2.2 Joint-scan Test Architecture

In the present work, *p-serial* and *p-random* part, are implemented using multiple serial scan and *RAS* respectively. The *p-serial* part can be replaced with well researched serial scan architecture like scan tree [24, 141], scan forest [239], and Illinois scan [63, 95]. Further, the integration of the current industrial serial scan architectures like Embedded Deterministic Test (*EDT*) [159], or DFTMax Compression Architecture [45, 108] with the proposed *JScan* architecture can be studied as a future direction.

Initial experiments on *JScan* implementation using the proposed scan cell shows promising results for improving the overall interconnect wire length. A chip level implementation of the Joint-scan architecture using the proposed scan cell, however, will give a better idea about routing congestion, overall design area, and both test and leakage power dissipation.

7.2.3 Scan Cell Design

The proposed low power scan cell design eliminates combinational switching during scan shift operation. The exact test power saving that it offers can be find out by doing experiments on few benchmark circuits. The second scan cell design which enables *LOS* delay test with slow scan enable signal needs to be implemented using benchmark circuits to know the exact area and power overhead numbers.

– ∗ – ∗ –

Bibliography

[1] M. Abramovici, M. A. Breuer, and A. D. Friedman. *Digital systems testing and testable design*. Computer Science Press, 1990. ISBN 978-0-7167-8179-0.

[2] J. M. Acken and S. D. Millman. Fault model evolution for diagnosis accuracy versus precision. In *Proceedings of Custom Integrated Circuits Conference*, pages 13.4.1–13.4.4, 1992.

[3] R. Adiga, G. Arpit, V. Singh, K. K. Saluja, H. Fujiwara, and A. D. Singh. On minimization of test application time for RAS. In *Proceedings of the 23rd VLSI Design Conference*, pages 393–398, 2010.

[4] R. Adiga, G. Arpit, V. Singh, K. K. Saluja, and A. D. Singh. Modified t-flip-flop based scan cell for RAS. In *Proceedings of the European Test Symposium*, ETS '10. IEEE Computer Society, 2010.

[5] V. D. Agrawal and M. R. Mercer. Testability measures - what do they tell us? In *Proceedings of the International Test Conference*, pages 391–396, November 1982.

[6] V. D. Agrawal and S. C. Seth. *Test generation for VLSI chips*. IEEE Computer Society Press, 1988.

[7] V. D. Agrawal, K. T. Cheng, D. D. Johnson, and T. Sheng Lin. Designing circuits with partial scan. *IEEE Transactions on Design & Test of Computers*, 5:8–15, March 1988. ISSN 0740-7475. doi: 10.1109/54.2032.

[8] V. D. Agrawal, K.-T. Cheng, and P. Agrawal. A directed search method for test generation using a concurrent simulator. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 8(2):131–138, Feb 1989. ISSN 0278-0070. doi: 10.1109/43.21831.

[9] S. Ahlawat, J. Tudu, A. Matrosova, and V. Singh. A new scan flip flop design to eliminate performance penalty of scan. In *2015 IEEE 24th Asian Test Symposium (ATS)*, pages 25–30, Nov 2015. doi: 10.1109/ATS.2015.12.

[10] S. Ahlawat, J. Tudu, A. Matrosova, and V. Singh. A high performance scan flip-flop design for serial and mixed mode scan test. In *2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pages 233–238, July 2016. doi: 10.1109/IOLTS.2016.7604709.

[11] S. Ahlawat, D. Vaghani, and V. Singh. Preventing scan-based side-channel attacks through key masking. In *2017 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pages 1–4, Oct 2017. doi: 10.1109/DFT.2017.8244434.

[12] S. Ahlawat, D. Vaghani, and V. Singh. An efficient test technique to prevent scan-based side-channel attacks. In *2017 22nd IEEE European Test Symposium (ETS)*, pages 1–2, May 2017. doi: 10.1109/ETS.2017.7968241.

[13] S. Ahlawat, D. Vaghani, J. Tudu, and V. Singh. On securing scan design from scan-based side-channel attacks. In *2017 IEEE 26th Asian Test Symposium (ATS)*, pages 58–63, Nov 2017. doi: 10.1109/ATS.2017.23.

[14] S. Ahlawat, J. Tudu, A. Matrosova, and V. Singh. A high performance scan flip-flop design for serial and mixed mode scan test. *IEEE Transactions on Device and Materials Reliability*, 18(2):321–331, June 2018. ISSN 1530-4388. doi: 10.1109/TDMR.2018.2835414.

[15] S. Ahlawat, D. Vaghani, N. Bazard, and V. Singh. Using misr as countermeasure

against scan-based side channel attacks. In *IEEE 16th East-West Design & Test Symposium (EWDTS) 2018*, Sep 2018.

[16] N. Ahmed, C. P. Ravikumar, M. Tehranipoor, and J. Plusquellic. At-speed transition fault testing with low speed scan enable. In *23rd IEEE VLSI Test Symposium (VTS'05)*, pages 42–47, May 2005. doi: 10.1109/VTS.2005.31.

[17] H. Ando. Testing vlsi with random acecss scan. In *Digest of Computer Society International Conference*, pages 50–52, 1980.

[18] K. Arabi, R. Saleh, and X. Meng. Power supply noise in socs: Metrics, management, and measurement. *IEEE Transactions on Design &; Test of Computers*, 24 (3):236–244, 2007. ISSN 0740-7475. doi: http://doi.ieeecomputersociety.org/10.1109/MDT.2007.79.

[19] P. Ashar and S. Malik. Implicit computation of minimum-cost feedback-vertex sets for partial scan and other applications. In *Proceedings of the 31st annual Design Automation Conference*, DAC '94, pages 77–80, New York, NY, USA, 1994. ACM. ISBN 0-89791-653-0. doi: http://doi.acm.org/10.1145/196244.196283.

[20] D. Baik and K. K. Saluja. Test cost reduction using partitioned grid random access scan. In *Proc of the 19th VLSI Design Conference*, 2006.

[21] D. Baik, S. Kajihara, and K. Saluja. Random access scan: A solution to test power, test data volume and test time. In *Proc of the VSLI Design Conference*, pages 883–888, 2004.

[22] D. H. Baik and K. K. Saluja. State-reuse test generation for progressive random access scan: Solution to test power, application time and data size. In *Proceedings of 14th Asian Test Symposium*, pages 272–277, Dec 2005.

[23] D. H. Baik and K. K. Saluja. Progressive random access scan: a simultaneous solution to test power, test data volume and test time. In *Proceedings 2005 IEEE*

International Test Conference, ITC 2005, Austin, TX, USA, November 8-10, 2005, page 10, 2005. doi: 10.1109/TEST.2005.1583994.

[24] S. Banerjee, D. R. Chowdhury, and B. B. Bhattacharya. An efficient scan tree design for compact test pattern set. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(7):1331–1339, July 2007. ISSN 0278-0070. doi: 10.1109/TCAD.2007.895840.

[25] P. H. Bardell, W. H. McAnney, and J. Savir. *Built-in Test for VLSI: Pseudorandom Techniques*. Wiley-Interscience, New York, NY, USA, 1987. ISBN 0-471-62463-2.

[26] C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, A. Ferko, B. Keller, D. Scott, B. Koenemann, and T. Onodera. Extending opmisr beyond 10 times; scan test efficiency. *IEEE Design Test of Computers*, 19(5):65–73, Sep 2002. ISSN 0740-7475. doi: 10.1109/MDT.2002.1033794.

[27] I. Bayraktaroglu and A. Orailoglu. Test volume and application time reduction through scan chain concealment. In *Proceedings of the 38th Design Automation Conference*, pages 151–155, June 2001. doi: 10.1145/378239.378388.

[28] R. G. Bennetts and F. P. M. Beenker. Partial scan: what problem does it solve? In *European Test Conference, Proc. of ETC 93., Third*, pages 99–106, Apr 1993.

[29] S. Bhatia. Will test compression run out of gas? In *International Test Conference (ITC) 2008*, pages 1–1, Oct 2008.

[30] S. Bhunia, H. Mahmoodi, D. Ghosh, S. Mukhopadhyay, and K. Roy. Low-power scan design using first-level supply gating. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(3):384–395, March 2005. ISSN 1063-8210. doi: 10.1109/TVLSI.2004.842885.

[31] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. Present: An ultra-lightweight block cipher. In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded*

Systems - CHES 2007, pages 450–466, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-74735-2.

[32] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch. A gated clock scheme for low power scan testing of logic ics or embedded cores. In *Proceedings of 10th Asian Test Symposium*, pages 253–258, 2001. doi: 10.1109/ATS.2001.990291.

[33] Y. Bonhomme, T. Yoneda, H. Fujiwara, and P. Girard. An efficient scan tree design for test time reduction. *Proceedings of 18th IEEE European Test Symposium (ETS) 2004*, 0:174–179, 2004. doi: http://doi.ieeecomputersociety.org/10.1109/ETSYM.2004.1347657.

[34] V. Boppana and W. K. Fuchs. Partial scan design based on state transition modeling. In *Proceedings of International Test Conference*, pages 538–547, 1996.

[35] M. A. Breuer. A random and an algorithmic technique for fault detection test generation for sequential circuits. *IEEE Transactions on Computers*, C-20(11):1364–1370, Nov 1971. ISSN 0018-9340. doi: 10.1109/T-C.1971.223140.

[36] M. Bushnell and V. D. Agrawal. *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*. Springer, Heildelberger Platz 3, 14197 Berlin, Germany, first edition, 2005.

[37] Cadence Design Systems. Virtuoso Liberate Reference Manual, Product Version 15.1, August 2016, Cadence Design Systems, Inc., August 2016. URL https://www.cadence.com/content/cadence-www/global/en_US/home/tools/custom-ic-analog-rf-design/library-characterization/virtuoso-liberate-characterization.html.

[38] G. R. Case. Analysis of actual fault mechanisms in cmos logic gates. In *Proceedings of the 13th Design Automation Conference*, DAC '76, pages 265–270, New York, NY, USA, 1976. ACM. doi: 10.1145/800146.804823. URL http://doi.acm.org/10.1145/800146.804823.

[39] K. Chakrabarty and S. Goel. *Testing for Small-Delay Defects in Nanoscale CMOS Integrated Circuits*. CRC Press, Boca Raton: CRC Press, first edition, 2013.

[40] S. T. Chakradhar, A. Balakrishnan, and V. D. Agrawal. An exact algorithm for selecting partial scan flip-flops. In *Proceedings of the 31st annual Design Automation Conference*, DAC '94, pages 81–86, New York, NY, USA, 1994. ACM. ISBN 0-89791-653-0. doi: http://doi.acm.org/10.1145/196244.196285.

[41] K. Chakravadhanula, V. Chickermane, D. Pearl, A. Garg, R. Khurana, S. Mukherjee, and P. Nagaraj. Smartscan - hierarchical test compression for pin-limited low power designs. In *2013 IEEE International Test Conference (ITC)*, pages 1–9, Sept 2013. doi: 10.1109/TEST.2013.6651897.

[42] A. Chandra and R. Kapur. Bounded adjacent fill for low capture power scan testing. In *26th IEEE VLSI Test Symposium (vts 2008)*, pages 131–138, April 2008. doi: 10.1109/VTS.2008.47.

[43] A. Chandra, H. Yan, and R. Kapur. Multimode illinois scan architecture for test application time and test data volume reduction. In *Proceedings of 25th IEEE VLSI Test Symposium (VTS) 2007*, pages 84–92, May 2007. doi: 10.1109/VTS.2007.39.

[44] A. Chandra, F. Ng, and R. Kapur. Low power illinois scan architecture for simultaneous power and test data volume reduction. In *Proceedings of Design, Automation and Test in Europe (DATE) 2008*, pages 462–467, March 2008. doi: 10.1109/DATE.2008.4484724.

[45] A. Chandra, S. Chebiyam, and R. Kapur. A case study on implementing compressed dft architecture. In *Proceedings of 23rd IEEE Asian Test Symposium*, pages 336–341, Nov 2014. doi: 10.1109/ATS.2014.68.

[46] U. Chandran and D. Zhao. Ss-ktc: A high-testability low-overhead scan architecture with multi-level security integration. In *2009 27th IEEE VLSI Test Symposium*, pages 321–326, May 2009. doi: 10.1109/VTS.2009.20.

[47] K. T. Cheng and V. D. Agrawal. A partial scan method for sequential circuits with feedback. *IEEE Transaction on Computers*, 39:544–548, April 1990. ISSN 0018-9340. doi: http://dx.doi.org/10.1109/12.54847.

[48] K. T. Cheng, S. Devadas, and K. Keutzer. Delay-fault test generation and synthesis for testability under a standard scan design methodology. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(8):1217–1231, Aug 1993. ISSN 0278-0070. doi: 10.1109/43.238614.

[49] G. Chiu and J. C. Li. A secure test wrapper design against internal and boundary scan attacks for embedded cores. *IEEE Transactions on VLSI Systems*, 20(1): 126–134, 2012. doi: 10.1109/TVLSI.2010.2089071.

[50] R. M. Chou, K. K. Saluja, and V. D. Agrawal. Power constraint scheduling of tests. In *Proceedings of 7th International Conference on VLSI Design*, pages 271–274, Jan 1994. doi: 10.1109/ICVD.1994.282700.

[51] Y. Chunhua and K. K. Saluja. A study of word oriented random-access-scan based BIST for low area overhead and low power. In *Proceedings of Workshop on Impact of Low-Power design on Test and Reliability*, 2008.

[52] A. Cui, Y. Luo, H. Li, and G. Qu. Why current secure scan designs fail and how to fix them? *Integration, the VLSI Journal*, 56:105 – 114, 2017.

[53] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy. Techniques for minimizing power dissipation in scan and combinational circuits during test application. *IEEE Transactions on Computer-Aided Design*, 17(12):1325 – 1333, December 1998.

[54] J. DaRolt, G. D. Natale, M. Flottes, and B. Rouzeyre. Scan attacks and countermeasures in presence of scan response compactors. In *16th European Test Symposium, ETS 2011, Trondheim, Norway, May 23-27, 2011*, pages 19–24, 2011. doi: 10.1109/ETS.2011.30.

[55] J. DaRolt, G. D. Natale, M. Flottes, and B. Rouzeyre. Are advanced dft structures sufficient for preventing scan-attacks? In *30th IEEE VLSI Test Symposium, VTS 2012, Maui, Hawaii, USA, 23-26 April 2012*, pages 246–251, 2012. doi: 10.1109/VTS.2012.6231061.

[56] J. DaRolt, G. D. Natale, M. Flottes, and B. Rouzeyre. A smart test controller for scan chains in secure circuits. In *2013 IEEE 19th International On-Line Testing Symposium (IOLTS), Chania, Crete, Greece, July 8-10, 2013*, pages 228–229, 2013. doi: 10.1109/IOLTS.2013.6604085.

[57] J. DaRolt, A. Das, G. D. Natale, M. Flottes, B. Rouzeyre, and I. Verbauwhede. Test versus security: Past and present. *IEEE Transactions on Emerging Topics on Computers*, 2(1):50–62, 2014. doi: 10.1109/TETC.2014.2304492.

[58] J. DaRolt, G. D. Natale, M. Flottes, and B. Rouzeyre. Thwarting scan-based attacks on secure-ics with on-chip comparison. *IEEE Transactions on VLSI Systems*, 22(4):947–951, 2014. doi: 10.1109/TVLSI.2013.2257903.

[59] B. Davis. *The Economics of Automatic Testing.* McGraw-Hill, London, United Kingdom: McGraw-Hill, first edition, 1982.

[60] K. De and A. Gunda. Failure analysis for full-scan circuits. In *Proceedings of IEEE International Test Conference (ITC)*, pages 636–645, Oct 1995. doi: 10.1109/TEST.1995.529892.

[61] V. Devanathan, C. Ravikumar, R. Mehrotra, and V. Kamakoti. Pmscan : A power-managed scan for simultaneous reduction of dynamic and leakage power during scan test. In *Proceedings of IEEE International Test Conference (ITC) 2007*, pages 1–9, Oct 2007. doi: 10.1109/TEST.2007.4437598.

[62] N. Devta-Prasanna, A. Gunda, P. Krishnamurthy, S. M. Reddy, and I. Pomeranz. A novel method of improving transition delay fault coverage using multiple scan enable signals. In *ICCD*, pages 471–474, 2005.

[63] S. Donglikar, M. Banga, M. Chandrasekar, and M. S. Hsiao. Fast circuit topology based method to configure the scan chains in illinois scan architecture. In *Proceedings of International Test Conference (ITC) 2009*, pages 1–10, Nov 2009. doi: 10.1109/TEST.2009.5355661.

[64] H. M. Dounavi and Y. Tsiatouhas. Stuck-at fault diagnosis in scan chains. In *Proceedings of the 9th IEEE International Conference on Design Technology of Integrated Systems in Nanoscale Era (DTIS)*, pages 1–6, May 2014. doi: 10.1109/DTIS.2014.6850663.

[65] C. Dufaza and G. Cambon. Lfsr-based deterministic and pseudo-random test pattern generator structures. In *Proceedings of the European Test Conference*, pages 27–34, April 1991.

[66] D. G. (Ed.). *Advances in Electronic Testing: Challenges and Methodologies*. Springer-Verlag US 2006, Springer, Boston, MA, 2006. ISBN 978-0-387-29409-4. doi: https://doi.org/10.1007/0-387-29409-0.

[67] S. Edirisooriya and G. Edirisooriya. Diagnosis of scan path failures. In *Proceedings of the 13th IEEE VLSI Test Symposium (VTS)*, pages 250–255, Apr 1995. doi: 10.1109/VTEST.1995.512645.

[68] E. B. Eichelberger. *Structured logic testing*. Englewood Cliffs, N.J. : Prentice Hall, 1991. ISBN 0138536805. URL https://trove.nla.gov.au/work/180110519.

[69] K. L. Einspahr and S. C. Seth. A switch-level test generation system for synchronous and asynchronous circuits. *Journal of Electronic Testing*, 6(1):59–73, 1995. doi: 10.1007/BF00993130. URL https://doi.org/10.1007/BF00993130.

[70] R. D. Eldred. Test routines based on symbolic logical statements. In *13th National Meeting of the Association for Computing Machinery*, ACM '58, pages 1–2, New York, NY, USA, 1958. ACM. doi: 10.1145/610937.610995. URL http://doi.acm.org/10.1145/610937.610995.

[71] M. ElShoukry, M. Tehranipoor, and C. Ravikumar. Partial gating optimization for power reduction during test application. In *Proceedings of 14th Asian Test Symposium (ATS) 2005*, pages 242–247, Dec 2005. doi: 10.1109/ATS.2005.87.

[72] M. Filipek, G. Mrugalski, N. Mukherjee, B. Nadeau-Dostie, J. Rajski, J. Solecki, and J. Tyszer. Low-power programmable PRPG with test compression capabilities. *IEEE Transactions on VLSI Systems*, 23(6):1063–1076, 2015. doi: 10.1109/TVLSI.2014.2332465.

[73] H. Fujiwara. *Logic Testing and Design for Testability.* Massachusetts Institute of Technology, Cambridge, MA, USA, 1985. ISBN 0-262-06096-5.

[74] H. Fujiwara and K. Fujiwara. Strongly secure scan design using generalized feed forward shift registers. *IEICE Transactions*, 98-D(10):1852–1855, 2015.

[75] J. M. Galey, R. E. Norby, and J. P. Roth. Techniques for the diagnosis of switching circuit failures. In *2nd Annual Symposium on Switching Circuit Theory and Logical Design (SWCT 1961)*, pages 152–160, Oct 1961. doi: 10.1109/FOCS.1961.33.

[76] S. Gerstendorfer and H. Wunderlich. Minimized power consumption for scan-based bist. In *Proceedings of International Test Conference (ITC) 1999*, pages 77–84, 1999. doi: 10.1109/TEST.1999.805616.

[77] J. Geuzebroek, E. J. Marinissen, A. Majhi, A. Glowatz, and F. Hapke. Embedded multi-detect atpg and its effect on the detection of unmodeled defects. In *2007 IEEE International Test Conference*, pages 1–10, Oct 2007. doi: 10.1109/TEST.2007.4437649.

[78] V. Gherman, H. J. Wunderlich, H. Vranken, F. Hapke, M. Wittke, and M. Garbers. Efficient pattern mapping for deterministic logic bist. In *Proceedings of the International Test Conference*, pages 48–56, Oct 2004. doi: 10.1109/TEST.2004.1386936.

[79] P. Girard. Survey of low-power testing of VLSI circuits. *IEEE Transactions on Design & Test of Computers*, pages 82 – 92, May 2002.

[80] P. Girard, N. Nicolici, and X. Wen, editors. *Power-Aware Testing and Test Strategies for Low Power Devices*. Springer US, first edition, 2010. ISBN 978-1-4419-0928-2. doi: 10.1007/978-1-4419-0928-2.

[81] P. Goel. An implicit enumeration algorithm to generate tests for combinational logic circuits. *IEEE Transactions on Computers*, C-30(3):215–222, March 1981. ISSN 0018-9340. doi: 10.1109/TC.1981.1675757.

[82] S. K. Goel, N. Devta-Prasanna, and M. Ward. Comparing the effectiveness of deterministic bridge fault and multiple-detect stuck fault patterns for physical bridge defects: A simulation and silicon study. In *2009 International Test Conference*, pages 1–10, Nov 2009. doi: 10.1109/TEST.2009.5355762.

[83] S. W. Golomb. *Shift Register Sequences*. Aegean Park Press, Laguna Hills, California, USA, 1982.

[84] R. Guo and S. Venkataraman. An algorithmic technique for diagnosis of faulty scan chains. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 25(9):1861–1868, Sept 2006. ISSN 0278-0070. doi: 10.1109/TCAD.2005.858267.

[85] R. Gupta and M. A. Breuer. The ballast methodology for structured partial scan design. *IEEE Transaction on Computers*, 39:538–544, April 1990. ISSN 0018-9340. doi: http://dx.doi.org/10.1109/12.54846.

[86] K. Hafner, H. C. Ritter, T. M. Schwair, S. Wallstab, M. Deppermann, J. Gessner, S. Koesters, W. Moeller, and G. Sandweg. Design and test of an integrated cryptochip. *IEEE Design & Test of Computers*, 8(4):6–17, 1991. doi: 10.1109/54.107201.

[87] A. W. Hakmi, S. Holst, H. J. Wunderlich, J. Schlffel, F. Hapke, and A. Glowatz. Restrict encoding for mixed-mode bist. In *Proceedings of the 27th IEEE VLSI Test Symposium*, pages 179–184, May 2009. doi: 10.1109/VTS.2009.43.

[88] I. Hamzaoglu and J. H. Patel. Reducing test application time for full scan embedded cores. In *Digest of Papers. Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing (Cat. No.99CB36352)*, pages 260–267, June 1999. doi: 10.1109/FTCS.1999.781060.

[89] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois. Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers. *IEEE Transactions on Computers*, 44(2):223–233, Feb 1995. ISSN 0018-9340. doi: 10.1109/12.364534.

[90] D. Hély, M. Flottes, F. Bancel, B. Rouzeyre, N. Bérard, and M. Renovell. Scan design and secure chip. In *10th IEEE International On-Line Testing Symposium (IOLTS 2004), 12-14 July 2004, Funchal, Madeira Island, Portugal*, pages 219–226, 2004. doi: 10.1109/IOLTS.2004.40.

[91] D. Hély, F. Bancel, M. Flottes, and B. Rouzeyre. Secure scan techniques: A comparison. In *12th IEEE International On-Line Testing Symposium (IOLTS 2006), 10-12 July 2006, Como, Italy*, pages 119–124, 2006. doi: 10.1109/IOLTS.2006.55.

[92] K. Heragu, J. H. Patel, and V. Agrawal. Sigma: A simulator for segment delay faults. In *Proceedings of International Conference on Computer Aided Design*, pages 502–508, Nov 1996. doi: 10.1109/ICCAD.1996.569902.

[93] K. Heragu, J. H. Patel, and V. D. Agrawal. Segment delay faults: a new fault model. In *Proceedings of 14th VLSI Test Symposium*, pages 32–39, Apr 1996. doi: 10.1109/VTEST.1996.510832.

[94] M. S. Hsiao, G. S. Saund, E. M. Rudnick, and J. H. Patel. Partial scan selection based on dynamic reachability and observability information. In *Proceedings of the Eleventh International Conference on VLSI Design*, VLSID '98, pages 174–, Washington, DC, USA, 1998. ISBN 0-8186-8224-8.

[95] F. Hsu, K. Butler, and J. Patel. A case study on the implementation of the illinois

scan architecture. In *Proceedings of International Test Conference (ITC) 2001*, pages 538–547, 2001. doi: 10.1109/TEST.2001.966672.

[96] Y. Hu, X. Fu, X. Fan, and H. Fujiwara. Localized random access scan: Towards low area and routing overhead. In *Proceedings of Asia and South Pacific Design Automation Conference (ASPDAC) 2008*, ASP-DAC '08, pages 565–570. IEEE Computer Society Press, 2008.

[97] L. Huang and Q. Xu. Economic analysis of testing homogeneous manycore chips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(8):1257–1270, Aug 2010. ISSN 0278-0070. doi: 10.1109/TCAD.2010.2049052.

[98] T.-C. Huang and K.-J. Lee. An input control technique for power reduction in scan circuits during test application. In *Test Symposium, 1999. (ATS '99) Proceedings. Eighth Asian*, pages 315–320, 1999. doi: 10.1109/ATS.1999.810769.

[99] Y. Huang, R. Guo, W. T. Cheng, and J. C. M. Li. Survey of scan chain diagnosis. *IEEE Transactions on Design & Test of Computers*, 25(3):240–248, May 2008. ISSN 0740-7475. doi: 10.1109/MDT.2008.83.

[100] INTEL Fact Sheets. Intels 10nm technology: Delivering the highest logic transistor density in the industry through the use of hyper scaling. *Intel Newsroom*, 2017. URL https://newsroom.intel.com/press-kits/leading-edge-intel-technology-manufacturing/.

[101] N. Ito. Automatic incorporation of on-chip testability circuits. In *Proceedings of Design Automation Conference*, pages 529–534, 1991.

[102] ITRS-2007. *International Technology Roadmap for Semiconductors Industry*. International Technology Roadmap for Semiconductors, 2007 edition edition, 2007.

[103] ITRS-2013. *Test and Test Equipment*. International Technology Roadmap for Semiconductor, 2013 edition edition, 2013.

[104] V. Iyengar and K. Chakrabarty. Precedence-based, preemptive, and power-constrained test scheduling for system-on-a-chip. In *Proceedings 19th IEEE VLSI Test Symposium. VTS 2001*, pages 368–374, 2001. doi: 10.1109/VTS.2001.923464.

[105] V. Iyengar, K. Chakrabarty, and E. J. Marinissen. Recent advances in test planning for modular testing of core-based socs. In *Test Symposium, 2002. (ATS '02). Proceedings of the 11th Asian*, pages 320–325, Nov 2002. doi: 10.1109/ATS.2002.1181731.

[106] S. K. Jain and V. D. Agrawal. Modeling and test generation algorithms for mos circuits. *IEEE Transactions on Computers*, C-34(5):426–433, May 1985. ISSN 0018-9340. doi: 10.1109/TC.1985.1676582.

[107] P. Kalla and M. J. Ciesielski. A comprehensive approach to the partial scan problem using implicit state enumeration. In *Proceedings of the IEEE International Test Conference*, ITC '98, pages 651–657, Washington, DC, USA, 1998. ISBN 0-7803-5093-6.

[108] R. Kapur and R. Ruiz. Maximum test cost reduction. *DFTMAX Compression Backgrounder, Synopsys*, 2009.

[109] R. Kapur and T. Williams. Tough challenges as design and test go nanometer. *Computer*, 32(11):42 – 45, Nov 1999.

[110] X. Kavousianos, D. Bakalis, and D. Nikolos. Efficient partial scan cell gating for low-power scan-based testing. *ACM Transactions on Design and Automation of Electronic Systems*, 14(2):28:1–28:15, Apr. 2009. ISSN 1084-4309. doi: 10.1145/1497561.1497571.

[111] B. Koenemann. Lfsr-coded test patterns for scan designs. In *Proceedings of the European Test Conference (ETC)*, pages 237–242, 1991.

[112] C. V. Krishna and N. A. Touba. Reducing test data volume using lfsr reseeding

with seed compression. In *Proceedings. International Test Conference*, pages 321–330, 2002. doi: 10.1109/TEST.2002.1041775.

[113] A. Krsti and K.-T. Cheng. *Delay Fault Testing for VLSI Circuits.* Springer US, first edition, 1998. ISBN 978-1-4615-5597-1, 978-0-7923-8295-9. doi: 10.1007/978-1-4615-5597-1.

[114] H. Kubo. A procedure for generating test sequences to detect sequential circuit failures. *NEC Research and Development*, 12(4):69–78, October 1968.

[115] A. Kumar, M. Kassab, E. Moghaddam, N. Mukherjee, J. Rajski, S. M. Reddy, J. Tyszer, and C. Wang. Isometric test data compression. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(11):1847–1859, Nov 2015. ISSN 0278-0070. doi: 10.1109/TCAD.2015.2432133.

[116] B. Kumar, A. Jindal, J. Tudu, B. Pandey, and V. Singh. Revisiting random access scan for effective enhancement of post-silicon observability. In *2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pages 132–137, July 2017. doi: 10.1109/IOLTS.2017.8046208.

[117] S. Kundu. On diagnosis of faults in a scan-chain. In *Digest of Papers, the 11th Annual IEEE VLSI Test Symposium (VTS)*, pages 303–308, April 1993. doi: 10.1109/VTEST.1993.313363.

[118] S. Kundu and A. Sanyal. *Power-Aware Testing and Test Strategies for Low Power Devices*. Springer US, Boston, MA, 2010. ISBN 978-1-4419-0928-2. doi: 10.1007/978-1-4419-0928-2_2.

[119] A. Kunzmann and J. H. Wunderlich. An analytical approach to the partial scan problem. *Journal of Electronic Testing: Theory and Applications*, 1:163–174, 1990.

[120] K. Le, D. Baik, and K. Saluja. Test time reduction to test for path-delay faults using enhanced random-access scan. In *Proceedings International Conference on VLSI Design (VLSID) 2007*, number - in VLSID '07, pages 769–774, 2007.

[121] H. S. Lee and K. Chakrabarty. Test challenges for 3d integrated circuits. *IEEE Transactions on IC Design and Test*, Sept/Oct 2009.

[122] J. Lee, M. Tehranipoor, C. Patel, and J. Plusquellic. Securing scan design using lock and key technique. In *20th IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT 2005), 3-5 October 2005, Monterey, CA, USA*, pages 51–62, 2005. doi: 10.1109/DFTVS.2005.58.

[123] J. Lee, M. Tehranipoor, and J. Plusquellic. A low-cost solution for protecting ips against scan-based side-channel attacks. In *24th IEEE VLSI Test Symposium (VTS 2006), 30 April - 4 May 2006, Berkeley, California, USA*, pages 94–99, 2006. doi: 10.1109/VTS.2006.7.

[124] K.-J. Lee, C. A. Njinda, and M. A. Breuer. Switest: a switch level test generation system for cmos combinational circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(5):625–637, May 1994. ISSN 0278-0070. doi: 10.1109/43.277636.

[125] K.-J. Lee, J.-J. Chen, and C.-H. Huang. Using a single input to support multiple scan chains. In *1998 IEEE/ACM International Conference on Computer-Aided Design. Digest of Technical Papers*, pages 74–78, Nov 1998. doi: 10.1109/ICCAD.1998.144247.

[126] S. Y. Lee and K. Saluja. Test application time reduction for sequential circuits with scan. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(9):1128–1140, Sep 1995.

[127] D. Leet, P. Shearon, and R. France. A cmos lssd test generation system. *IBM Journal of Research and Development*, 28(5):625–635, Sept 1984. ISSN 0018-8646. doi: 10.1147/rd.285.0625.

[128] Y. Levendel and P. Menon. Transistion faults in combinational circuits: input transition test generation and fault simulation. In *Fault Tolerant Computing Symposium*, pages 278–283, July 1986.

[129] W. Li, S. M. Reddy, and I. Pomeranz. On reducing peak current and power during test. In *IEEE Computer Society Annual Symposium on VLSI: New Frontiers in VLSI Design (ISVLSI'05)*, pages 156–161, May 2005. doi: 10.1109/ISVLSI.2005.53.

[130] W.-N. Li, S. M. Reddy, and S. K. Sahni. On path selection in combinational logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 8(1):56–63, Jan 1989. ISSN 0278-0070. doi: 10.1109/43.21819.

[131] C. H. Liang and C. L. Lee. An effective methodology for mixed scan and reset design based on test generation and structure of sequential circuits. In *Proceedings of the 10th Anniversary Compendium of Papers from Asian Test Symposium 1992-2001*, ATS '01, pages 290–, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1233-x.

[132] X. Lin, I. Pomeranz, and S. M. Reddy. Full scan fault coverage with partial scan. In *Proceedings of the conference on Design, Automation, and Test in Europe*, DATE '99, New York, NY, USA, 1999. ACM. ISBN 1-58113-121-6. doi: http://doi.acm.org/10.1145/307418.307545.

[133] X. Lin, J. Rajski, I. Pomeranz, and S. Reddy. On static test compaction and test pattern ordering for scan designs. In *Proceedings of International Test Conference (ITC) 2001.*, pages 1088–1097, 2001.

[134] X. Lin, R. Press, J. Rajski, P. Reuter, T. Rinderknecht, B. Swanson, and N. Tamarapalli. High-frequency, at-speed scan testing. *IEEE Design and Test of Computer*, 20(5):17 – 25, September-October 2003.

[135] P. C. Maxwell and R. C. Aitken. Biased voting: A method for simulating cmos bridging faults in the presence of variable gate logic thresholds. In *Proceedings of IEEE International Test Conference - (ITC)*, pages 63–72, Oct 1993. doi: 10.1109/TEST.1993.470717.

[136] P. Mazumder and E. M. Rudnick, editors. *Genetic Algorithms for VLSI Design, Layout &Amp; Test Automation.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999. ISBN 0-13-011566-5.

[137] K. C. Y. Mei. Bridging and stuck-at faults. *IEEE Transactions on Computers*, C-23(7):720–727, July 1974. ISSN 0018-9340. doi: 10.1109/T-C.1974.224020.

[138] A. Mishra, N. Sinha, Satdev, V. Singh, S. Chakravarty, and A. D. Singh. Modified scan flip-flop for low power testing. In *Proceedings of the 19th IEEE Asian Test Symposium, ATS 2010, 1-4 December 2010, Shanghai, China*, pages 367–370, 2010. doi: 10.1109/ATS.2010.69.

[139] S. Mitra and K. S. Kim. X-compact: an efficient response compaction technique. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(3):421–432, March 2004.

[140] S. Mitra and K. S. Kim. Xpand: an efficient test stimulus compression technique. *IEEE Transactions on Computers*, 55(2):163–173, Feb 2006. ISSN 0018-9340. doi: 10.1109/TC.2006.31.

[141] K. Miyase and S. Kajihara. Optimal scan tree construction with test vector modification for test compression. In *Proceedings of 12th IEEE Asian Test Symposium*, pages 136–141, Nov 2003.

[142] E. K. Moghaddam, J. Rajski, S. M. Reddy, X. Li, N. Mukherjee, and M. Kassab. Low capture power at-speed test in edt environment. In *Proceedings of IEEE International Test Conference*, pages 24 – 34, November 2010.

[143] A. Mudlapur, V. Agrawal, and A. Singh. A random scan architecture to reduce hardware overhead. In *Proceedings of the International Test Conference*, number 15.1 in ITC '05, pages 1–9, 2005.

[144] P. Muth. A nine-valued circuit model for test generation. *IEEE Transactions on*

Computers, C-25(6):630–636, June 1976. ISSN 0018-9340. doi: 10.1109/TC.1976.1674663.

[145] S. Narayanan and A. Das. An efficient scheme to diagnose scan chains. In *Proceedings of International Test Conference (ITC)*, pages 704–713, Nov 1997. doi: 10.1109/TEST.1997.639683.

[146] S. Narayanan, R. Gupta, and M. Breuer. Configuring multiple scan chains for minimum test time. In *Digest of Technical Papers., IEEE/ACM International Conference on Computer-Aided Design (ICCAD) 1992*, pages 4–8, Nov 1992.

[147] G. D. Natale, M. Doulcier, M. Flottes, and B. Rouzeyre. Self-test techniques for crypto-devices. *IEEE Transactions on VLSI Systems*, 18(2):329–333, 2010. doi: 10.1109/TVLSI.2008.2010045.

[148] N. Nicolic and X. Wen. Embedded tutorial on low power test. In *Proceedings of 12th European Test Symposium*, pages 202–210, 2007.

[149] S. K. Niraj K. Jha. *Testing and Reliable Design of CMOS Circuits*, volume 88. Springer US, first edition, 1990. ISBN 978-1-4613-1525-4. doi: 10.1007/978-1-4613-1525-4.

[150] F. Novak and A. Biasizzo. Security extension for IEEE std 1149.1. *J. Electronic Testing*, 22(3):301–303, 2006. doi: 10.1007/s10836-006-7720-x.

[151] C. Paar and J. Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer Publishing Company, Incorporated, 1st edition, 2009. ISBN 3642041000, 9783642041006.

[152] S. Paul, R. S. Chakraborty, and S. Bhunia. Vim-scan: A low overhead scan design approach for protection of secret key in scan-based secure chips. In *25th IEEE VLSI Test Symposium (VTS 2007), 6-10 May 2007, Berkeley, California, USA*, pages 455–460, 2007. doi: 10.1109/VTS.2007.89.

[153] I. Pomeranz. Reducing the input test data volume under transparent scan. *IET Computers Digital Techniques*, 8(1):1–10, January 2014.

[154] I. Pomeranz and S. Reddy. Reducing test application time for full scan circuits by the addition of transfer sequences. In *Proceedings of the Ninth Asian Test Symposium*, pages 317–322, 2000.

[155] I. Pomeranz and S. M. Reddy. On n-detection test sets and variable n-detection test sets for transition faults. In *Proceedings of 17th IEEE VLSI Test Symposium*, pages 173–180, 1999. doi: 10.1109/VTEST.1999.766662.

[156] B. Pouya and A. L. Crouch. Optimization trade-offs for vector volume and test power. In *Proceedings of International Test Conference (ITC) 2000*, pages 873–881, 2000. doi: 10.1109/TEST.2000.894298.

[157] G. R. Putzolu and J. P. Roth. A heuristic algorithm for the testing of asynchronous circuits. *IEEE Transactions on Computers*, C-20(6):639–647, June 1971. ISSN 0018-9340. doi: 10.1109/T-C.1971.223315.

[158] J. Rajski, M. Kassab, N. Mukherjee, N. Tamarapalli, J. Tyszer, and J. Qian. Embedded deterministic test for low-cost manufacturing. *IEEE Transactions on Design & Test of Computers*, 20(5):58–66, Sept 2003. ISSN 0740-7475. doi: 10.1109/MDT.2003.1232257.

[159] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee. Embedded deterministic test. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(5):776–792, May 2004.

[160] J. Rajski, J. Tyszer, N. Mukherjee, and T. Rinderknecht. Embedded test for low cost manufacturing. In *Proceedings of the 17th International Conference on VLSI Design*, pages 21–23, 2004. doi: 10.1109/ICVD.2004.1260896.

[161] S. Ravi. Power-aware test: Challenges and solutions. In *Proceedings of IEEE*

International Test Conference (ITC) 2007, pages 1–10, Oct 2007. doi: 10.1109/TEST.2007.4437660.

[162] M. A. Razzaq, V. Singh, and A. D. Singh. SSTKR: secure and testable scan design through test key randomization. In *Proceedings of the 20th IEEE Asian Test Symposium, ATS 2011, New Delhi, India, November 20-23, 2011*, pages 60–65, 2011. doi: 10.1109/ATS.2011.85.

[163] S. M. Reddy, M. K. Reddy, and V. D. Agrawal. Robust tests for stuck-open faults in cmos combinational logic circuits. In *Proceedings of International Symposium on Fault-Tolerant Computing*, pages 44–49, 1984.

[164] S. M. Reddy, I. Pomeranz, and S. Kajihara. On the effects of test compaction on defect coverage. In *Proceedings of 14th VLSI Test Symposium*, pages 430–435, Apr 1996. doi: 10.1109/VTEST.1996.510889.

[165] S. Remersaro, X. Lin, Z. Zhang, S. M. Reddy, I. Pomeranz, and J. Rajski. Preferred fill: A scalable method to reduce capture power for scan based designs. In *2006 IEEE International Test Conference*, pages 1–10, Oct 2006. doi: 10.1109/TEST.2006.297694.

[166] A. W. Righter, C. F. Hawkins, J. M. Soden, and P. Maxwell. Cmos ic reliability indicators and burn-in economics. In *Proceedings International Test Conference 1998 (IEEE Cat. No.98CH36270)*, pages 194–203, Oct 1998. doi: 10.1109/TEST.1998.743152.

[167] P. M. Rosinger, B. M. Al-Hashimi, and N. Nicolici. Power constrained test scheduling using power profile manipulation. In *The 2001 IEEE International Symposium on Circuits and Systems (ISCAS 2001) (Cat. No.01CH37196)*, volume 5, pages 251–254 vol. 5, 2001. doi: 10.1109/ISCAS.2001.922032.

[168] E. Rudnick and J. Patel. Efficient techniques for dynamic test sequence compaction. *IEEE Transactions on Computers*, 48(3):323–330, Mar 1999.

[169] M. Sachdev. *Defect Oriented Testing for CMOS Analog and Digital Circuits.* Kluwer Academic Publishers, Norwell, MA, USA, first edition, 1998. ISBN 0-7923-8083-5.

[170] K. K. Saluja. Outstanding challenges in testing nano technology based integrated circuits. In *Proceedings of Asian Test Symposium*, 2003.

[171] A. Sanghani, B. Yang, K. Natarajan, and C. Liu. Design and implementation of a time-division multiplexing scan architecture using serializer and deserializer in gpu chips. In *29th VLSI Test Symposium*, pages 219–224, May 2011. doi: 10.1109/VTS.2011.5783724.

[172] R. Sankaralingam and N. A. Touba. Controlling peak power during scan testing. In *Proceedings 20th IEEE VLSI Test Symposium (VTS 2002)*, pages 153–159, April 2002. doi: 10.1109/VTS.2002.1011127.

[173] R. Sankaralingam, R. R. Oruganti, and N. A. Touba. Static compaction techniques to control scan vector power dissipation. In *Proceedings 18th IEEE VLSI Test Symposium*, pages 35–40, April 2000. doi: 10.1109/VTEST.2000.843824.

[174] R. Sankaralingam, B. Pouya, and N. A. Touba. Reducing power dissipation during test using scan chain disable. In *Proceedings of 19th IEEE VLSI Test Symposium (VTS) 2001*, pages 319–324, 2001. doi: 10.1109/VTS.2001.923456.

[175] V. R. Sar-Dessai and D. M. H. Walker. Resistive bridge fault modeling, simulation and test generation. In *International Test Conference 1999. Proceedings (IEEE Cat. No.99CH37034)*, pages 596–605, 1999. doi: 10.1109/TEST.1999.805784.

[176] Y. Sato, S. Hamada, T. Maeda, A. Takatori, Y. Nozuyama, and S. Kajihara. Invisible delay quality SDQM model lights up what could not be seen. In *IEEE International Conference on Test, 2005.*, pages 9 pp.–1210, Nov 2005. doi: 10.1109/TEST.2005.1584088.

[177] G. S. Saund, M. S. Hsiao, and J. H. Patel. Partial scan beyond cycle cutting. In *Proceedings of the 27th International Symposium on Fault-Tolerant Computing (FTCS '97)*, FTCS '97, pages 320–, Washington, DC, USA, 1997. ISBN 0-8186-7831-3.

[178] J. Savir. Skewed-load transition test: Part i, calculus. In *Proceedings of International Test Conference*, pages 705–713, 1992.

[179] J. Savir and S. Patil. Broad-side delay test. *IEEE Transanction on Computer-Aided Design of Integrated Circuit and System*, 13(8):1057 – 1065, August 1994.

[180] J. Saxena, K. M. Butler, V. B. Jayaram, S. Kundu, N. V. Arvind, P. Sreeprakash, and M. Hachinger. A case study of ir-drop in structured in at-speed testing. In *Proceedings of IEEE International Test Conference (ITC) 2003*, pages 1098 – 1104, September 2003.

[181] J. L. Schafer, F. A. Policastri, and R. J. McNulty. Partner srls for improved shift register diagnostics. In *Digest of Papers, 10th Anniversary of IEEE VLSI Test Symposium (VTS)*, pages 198–201, April 1992. doi: 10.1109/VTEST.1992.232749.

[182] D. M. Schuler, E. G. Ulrich, T. E. Baker, and S. P. Bryant. Random test generation using concurrent logic simulation. In *Proceedings of the 12th Design Automation Conference*, DAC '75, pages 261–267, Piscataway, NJ, USA, 1975. IEEE Press. URL http://dl.acm.org/citation.cfm?id=800261.809076.

[183] M. D. Schuster and R. E. Bryant. Concurrent fault simulation of mos digital circuits. *CaltechAUTHORS*, 1983. URL http://resolver.caltech.edu/CaltechAUTHORS:20120420-114600956.

[184] M. D. Schuster and R. E. Bryant. Concurrent fault simulation of mos digital circuits. In *Proceedings of the Conference on Advanced Research in VLSI*, pages 129–138, January 1984.

[185] S. Seo, Y. Lee, J. Lee, and S. Kang. A scan shifting method based on clock gating of multiple groups for low power scan testing. In *16th International Symposium on Quality Electronic Design (ISQED) 2015*, pages 162–166, March 2015. doi: 10.1109/ISQED.2015.7085417.

[186] M. Sharma, J. Patel, and J. Rearick. Test data compression and test time reduction of longest-path-per-gate tests based on illinois scan architecture. In *Proceedings of 21st VLSI Test Symposium (VTS) 2003*, pages 15–21, April 2003.

[187] S. Sharma and M. S. Hsiao. Combination of structural and state analysis for partial scan. In *Proceedings of the The 14th International Conference on VLSI Design (VLSID '01)*, VLSID '01, pages 134–, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-0831-6.

[188] M. D. Silva, M. Flottes, G. D. Natale, B. Rouzeyre, P. Prinetto, and M. Restifo. Scan chain encryption for the test, diagnosis and debug of secure circuits. In *22nd IEEE European Test Symposium, ETS 2017, Limassol, Cyprus, May 22-26, 2017*, pages 1–6, 2017. doi: 10.1109/ETS.2017.7968248.

[189] M. D. Silva, M. l. Flottes, G. D. Natale, and B. Rouzeyre. Experimentations on scan chain encryption with present. In *2017 IEEE 2nd International Verification and Security Workshop (IVSW)*, pages 45–50, July 2017. doi: 10.1109/IVSW.2017.8031543.

[190] M. D. Silva, M. L. Flottes, G. D. Natale, and B. Rouzeyre. Preventing scan attacks on secure circuits through scan chain encryption. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 1–1, 2018. ISSN 0278-0070. doi: 10.1109/TCAD.2018.2818722.

[191] O. Sinanoglu. Eliminating performance penalty of scan. In *25th International Conference on VLSI Design, VLSID 2012, Hyderabad, India, January 7-11, 2012*, pages 346–351, 2012. doi: 10.1109/VLSID.2012.95.

[192] G. L. Smith. Model for delay faults based upon paths. In *International Test Conference (ITC 1985)*, pages 342–349, November 1985.

[193] J. Song, T. Jung, J. Jung, and S. Park. An Efficient Technique to Protect AES Secret Key from Scan Test Channel Attacks. *Journal of Semiconductor Technology and Science*, 12:286–292, 2012.

[194] P. Song, F. Stellari, T. Xia, and A. J. Weger. A novel scan chain diagnostics technique based on light emission from leakage current. In *Proceedings of International Test Conference (ITC)*, pages 140–147, Oct 2004. doi: 10.1109/TEST.2004.1386946.

[195] K. Stanley. High-accuracy flush-and-scan software diagnostic. *IEEE Transactions on Design & Test of Computers*, 18(6):56–62, Nov 2001. ISSN 0740-7475. doi: 10.1109/54.970425.

[196] F. Stellari, P. Song, T. Xia, and A. J. Weger. Broken scan chain diagnostics based on time-integrated and time-dependent emission measurements. In *30th International Symposium for Testing and Failure Analysis (ISTFA)*, pages 52–57, 2004.

[197] V. Stojanovic and V. G. Oklobdzija. Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems. *IEEE Journal of Solid-State Circuits*, 34:536–548, 1999.

[198] C. E. Stroud, editor. *A Designers Guide to Built-In Self-Test*. Springer US, Springer Science+Business Media New York, first edition, 2002. ISBN 978-0-306-47504-7. doi: 10.1007/b117480.

[199] T. Takasaki, T. Inoue, and H. Fujiwara. Partial scan design methods based on internally balanced structure. In *Proc. of the Asia and South Pacific Design Automation Conference (ASPDAC) 1998*, pages 211–216, Feb 1998.

[200] T. Takasaki, T. Inoue, and H. Fujiwara. Partial scan design methods based on internally balanced structure. In *Design Automation Conference, Proc. of the Asia and South Pacific*, pages 211–216, Feb 1998.

[201] M. Tehranipoor and K. M. Butler. Power supply noise: A survey on effects and research. *IEEE Transactions on Design & Test of Computers*, 27(2):51–67, 2010. ISSN 0740-7475. doi: http://doi.ieeecomputersociety.org/10.1109/MDT.2010.52.

[202] M. Tehranipoor, K. Peng, and K. Chakrabarty. *Test and Diagnosis for Small-Delay Defects*. Springer Publishing Company, Incorporated, 1st edition, 2011. ISBN 1441982965, 9781441982964.

[203] R. C. Tekumalla, P. Kumar, P. Krishnamoorthy, and P. Madhani. Low-power and area-efficient scan cell for integrated circuit testing, Oct. 22 2013. US Patent 8,566,658.

[204] The Semiconductors Newsletter. Intel now packs 100 million transistors in each square millimeter. *IEEE SPECTRUM*, March 2017. URL https://spectrum.ieee.org/nanoclast/semiconductors/processors/intel-now-packs-100-million-transistors-in-each-square-millimeter.

[205] The Semiconductors Newsletter. Euv lithography finally ready for chip manufacturing. *IEEE SPECTRUM*, January 2018. URL https://spectrum.ieee.org/semiconductors/nanotechnology/euv-lithography-finally-ready-for-chip-manufacturing.

[206] N. A. Touba. Survey of test vector compression techniques. *IEEE Transactions on Design and Test of Computers*, 23(4):294–303, April 2006. ISSN 0740-7475. doi: 10.1109/MDT.2006.105.

[207] J. Tudu. Jscan: A joint-scan dft architecture to minimize test time, pattern volume, and power. In *2016 20th International Symposium on VLSI Design and Test (VDAT)*, pages 1–6, May 2016. doi: 10.1109/ISVDAT.2016.8064866.

[208] J. T. Tudu. Low power test methodology for SoC : Solutions for peak power minimization. Master's thesis, Dept of Computer Science and Automation, Indian Institute of Science Bangalore, India, July 2010. URL http://etd.ncsi.iisc.ernet.in/handle/2005/2242.

[209] J. T. Tudu. *Power Issues in SoCs : Power Aware DFT Architecture and Power Estimation*. PhD thesis, Dept of Computer Science and Automation, Indian Institute of Science Bangalore, India, July 2016. URL http://etd.iisc.ernet.in/2005/3003.

[210] J. T. Tudu and S. Ahlawat. Guided shifting of test pattern to minimize test time in serial scan. In *2016 20th International Symposium on VLSI Design and Test (VDAT)*, pages 1–6, May 2016. doi: 10.1109/ISVDAT.2016.8064851.

[211] J. T. Tudu, E. Larsson, V. Singh, and V. D. Agrawal. On minimization of peak power for scan circuit during test. In *14th IEEE European Test Symposium, ETS 2009, Sevilla, Spain, May 25-29, 2009*, pages 25–30, 2009. doi: 10.1109/ETS.2009.36.

[212] J. T. Tudu, E. Larsson, V. Singh, and H. Fujiwara. Scan cell reordering to minimize peak power during test cycle: A graph theoretic approach. In *15th European Test Symposium, ETS 2010, Prague, Czech Republic, May 24-28, 2010*, page 259, Nov 2010. doi: 10.1109/TEST.2005.1583994.

[213] J. T. Tudu, E. Larsson, V. Singh, and H. Fujiwara. Graph theoretic approach for scan cell reordering to minimize peak shift power. In *Proceedings of the 20th Symposium on Great Lakes Symposium on VLSI*, GLSVLSI '10, pages 73–78. ACM, 2010.

[214] D. Vaghani, S. Ahlawat, J. Tudu, M. Fujita, and V. Singh. On securing scan design through test vector encryption. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, May 2018. doi: 10.1109/ISCAS.2018.8351212.

[215] I. Verbauwhede, F. Hoornaert, J. Vandewalle, and H. D. Man. Security considerations in the design and implementation of a new DES chip. In *Advances in Cryptology - EUROCRYPT '87, Workshop on the Theory and Application of of Cryptographic Techniques, Amsterdam, The Netherlands, April 13-15, 1987, Proceedings*, pages 287–300, 1987. doi: 10.1007/3-540-39118-5_26.

[216] R. L. Wadsack. Fault modeling and logic simulation of cmos and mos integrated circuits. *The Bell System Technical Journal*, 57(5):1449–1474, May 1978. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1978.tb02106.x.

[217] K. Wagner. Design for testability in the amdahl 580. In *Digest of Computer Society International Conference*, pages 384–288, 1983.

[218] J. A. Waicukauski, E. Lindbloom, B. K. Rosen, and V. S. Iyengar. Transition fault simulation. *IEEE Design and Test of Computers*, 4(2):32–38, April 1987. ISSN 0740-7475. doi: 10.1109/MDT.1987.295104.

[219] L.-T. Wang, C.-W. Wu, and X. Wen, editors. *{VLSI} Test Principles and Architectures*. Morgan Kaufmann, San Francisco, first edition, 2006. ISBN 978-0-12-370597-6. doi: https://doi.org/10.1016/B978-012370597-6/50000-7. URL https://www.sciencedirect.com/science/article/pii/B9780123705976500007.

[220] L. T. Wang, X. Wen, S. Wu, Z. Wang, Z. Jiang, B. Sheu, and X. Gu. Virtualscan: Test compression technology using combinational logic and one-pass atpg. *IEEE Transactions on Design & Test of Computers*, 25(2):122–130, March 2008. ISSN 0740-7475. doi: 10.1109/MDT.2008.56.

[221] S. Wang and S. K. Gupta. Atpg for heat dissipation minimization during test application. *IEEE Transactions on Computers*, 47(2):256–262, Feb 1998. ISSN 0018-9340. doi: 10.1109/12.663775.

[222] S. Wang and S. K. Gupta. An automatic test pattern generator for minimizing

switching activity during scan testing activity. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(8):954–968, Aug 2002. ISSN 0278-0070. doi: 10.1109/TCAD.2002.800460.

[223] S. Wang, X. Liu, and S. T. Chakradhar. Hybrid delay scan: a low hardware overhead scan-based delay test technique for high fault coverage and compact test sets. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE) 2004*, page 21296. IEEE Computer Society, 2004.

[224] S. J. Wang, K. S. M. Li, S. C. Chen, H. Y. Shiu, and Y. L. Chu. Scan-chain partition for high test-data compressibility and low shift power under routing constraint. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28(5):716–727, May 2009. ISSN 0278-0070. doi: 10.1109/TCAD.2009.2015741.

[225] X. Wen, Y. Yamashita, S. Kajihara, L.-T. Wang, K. K. Saluja, and K. Kinoshita. On low-capture-power test generation for scan testing. In *23rd IEEE VLSI Test Symposium (VTS'05)*, pages 265–270, May 2005. doi: 10.1109/VTS.2005.60.

[226] X. Wen, Y. Yamashita, S. Morishima, S. Kajihara, L.-T. Wang, K. K. Saluja, and K. Kinoshita. Low-capture-power test generation for scan-based at-speed testing. In *IEEE International Conference on Test, 2005.*, pages 10 pp.–1028, Nov 2005. doi: 10.1109/TEST.2005.1584068.

[227] X. Wen, K. Miyase, S. Kajihara, T. Suzuki, Y. Yamato, P. Girard, Y. Ohsumi, and L.-T. Wang. A novel scheme to reduce power supply noise for high-quality at-speed scan testing. In *2007 IEEE International Test Conference*, pages 1–10, Oct 2007. doi: 10.1109/TEST.2007.4437632.

[228] M. J. Y. Williams and J. B. Angell. Enhancing testability of large-scale integrated circuits via test points and additional logic. *IEEE Transactions on Computers*, C-22(1):46–60, Jan 1973. ISSN 0018-9340. doi: 10.1109/T-C.1973.223600.

[229] M. J. Y. Williams and J. B. Angell. Signature analysis: A new digital field service method. *Hewlett- Packard Journal*, 28(9):2–8, May 1977.

[230] T. W. Williams, W. Daehn, M. Gruetzner, and C. W. Starke. Comparison of aliasing errors for primitive and non-primitive polynomials. In *Proceedings of the International Test Conference*, pages 282–288, September 1986.

[231] T. W. Williams, W. Daehn, M. Gruetzner, and C. W. Starke. Bounds and analysis of aliasing errors in linear feedback shift registers. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 7(1):75–83, Jan 1988. ISSN 0278-0070. doi: 10.1109/43.3132.

[232] P. Wohl, J. A. Waicukauski, S. Patel, F. DaSilva, T. W. Williams, and R. Kapur. Efficient compression of deterministic patterns into multiple prpg seeds. In *IEEE International Conference on Test, 2005.*, pages 10 pp.–925, Nov 2005. doi: 10.1109/TEST.2005.1584057.

[233] P. Wohl, J. A. Waicukauski, R. Kapur, S. Ramnath, E. Gizdarski, T. W. Williams, and P. Jaini. Minimizing the impact of scan compression. In *25th IEEE VLSI Test Symposium (VTS'07)*, pages 67–74, May 2007. doi: 10.1109/VTS.2007.38.

[234] P. Wohl, J. A. Waicukauski, F. Neuveux, and E. Gizdarski. Fully x-tolerant, very high scan compression. In *Design Automation Conference*, pages 362–367, June 2010. doi: 10.1145/1837274.1837366.

[235] F. Wu, L. Dilillo, A. Bosio, P. Girard, S. Pravossoudovitch, A. Virazel, M. Tehranipoor, K. Miyase, X. Wen, and N. Ahmed. Is test power reduction through x-filling good enough? In *Proceedings of IEEE International Test Conference, ITC 2010, Austin, TX, USA, November 2-4, 2010*, page 805, 2010. doi: 10.1109/TEST.2010.5699297.

[236] Y. Wu. Diagnosis of scan chain failures. In *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 217–222, Nov 1998. doi: 10.1109/DFTVS.1998.732169.

[237] D. Xiang and J. H. Patel. Partial scan design based on circuit state information

and functional analysis. *IEEE Transactions on Computers*, 53:276–287, March 2004. ISSN 0018-9340. doi: http://dx.doi.org/10.1109/TC.2004.1261835.

[238] D. Xiang, S. Venkataraman, W. K. Fuchs, and J. H. Patel. Partial scan design based on circuit state information. In *In Proceedings of the Design Automation Conference*, pages 807–812, Las Vegas, NV, 1996.

[239] D. Xiang, K. Li, J. Sun, and H. Fujiwara. Reconfigured scan forest for test application cost, test data volume, and test power reduction. *IEEE Transactions on Computers*, 56(4):557–562, April 2007. ISSN 0018-9340. doi: 10.1109/TC.2007.1002.

[240] G. Xu and A. D. Singh. Achieving high transition delay fault coverage with partial DTSFF scan chains. In *IEEE International Test Conference, ITC 2007, Santa Clara, California, USA, October 21-26*, pages 1–9, 2007. doi: 10.1109/TEST.2007.4437608.

[241] Yang, S. Chakravarty, N. Devta-Prasanna, S. M. Reddy, and I. Pomeranz. Improving the detectability of resistive open faults in scan cells. In *Proceedings of the 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, DFT '09, pages 383–391, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-0-7695-3839-6. doi: http://dx.doi.org/10.1109/DFT.2009.30.

[242] B. Yang, K. Wu, and R. Karri. Scan based side channel attack on dedicated hardware implementations of data encryption standard. In *Proceedings 2004 International Test Conference (ITC 2004), October 26-28, 2004, Charlotte, NC, USA*, pages 339–344, 2004. doi: 10.1109/ITC.2004.157.

[243] B. Yang, K. Wu, and R. Karri. Secure scan: a design-for-test architecture for crypto chips. In *Proceedings 42nd Design Automation Conference, 2005.*, pages 135–140, June 2005.

[244] F. Yang, S. Chakravarty, N. Devta-Prasanna, S. M. Reddy, and I. Pomeranz. On the detectability of scan chain internal faults : An industrial case study. In

Proceedings of the 26th IEEE VLSI Test Symposium, pages 79–84, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 0-7695-3123-7. doi: 10.1109/VTS.2008.13.

[245] F. Yang, S. Chakravarty, N. Devta-Prasanna, S. M. Reddy, and I. Pomeranz. Improving the detectability of resistive open faults in scan cells. In *Proceedings of the 2009 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, DFT '09, pages 383–391, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-0-7695-3839-6. doi: http://dx.doi.org/10.1109/DFT.2009.30.

[246] K. Yang, K.-T. Cheng, and L.-C. Wang. Trangen: A sat-based atpg for path-oriented transition faults. In *Proceedings of the 2004 Asia and South Pacific Design Automation Conference*, ASP-DAC '04, pages 92–97, Piscataway, NJ, USA, 2004. IEEE Press. ISBN 0-7803-8175-0. URL http://dl.acm.org/citation.cfm?id=1015090.1015114.

[247] G. Zeng and H. Ito. Concurrent core test for soc using shared test set and scan chain disable. In *Proceedings of the Design Automation Test in Europe Conference*, volume 1, pages 1–6, March 2006. doi: 10.1109/DATE.2006.243928.

[248] W. Zhang, S. Lu, and S. Zhang. Low power scan flip-flop cell, Nov. 4 2014. US Patent 8,880,965.

[249] Y. Zorian. A distributed bist control scheme for complex vlsi devices. In *11th Annual IEEE VLSI Test Symposium (VTS) 1993*, pages 4–9, April 1993. doi: 10.1109/VTEST.1993.313316.