



HAL
open science

Analog Hardware Fault Diagnosis

Antonios Pavlidis

► **To cite this version:**

Antonios Pavlidis. Analog Hardware Fault Diagnosis. Micro and nanotechnologies/Microelectronics. Sorbonne Université, 2021. English. NNT : 2021SORUS452 . tel-03526323v2

HAL Id: tel-03526323

<https://hal.science/tel-03526323v2>

Submitted on 22 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT
DE SORBONNE UNIVERSITÉ

ANALOG HARDWARE FAULT DIAGNOSIS
(DIAGNOSTIC DE FAUTES MATÉRIELLES DANS LES
CIRCUITS ANALOGIQUES)

présentée par
ANTONIOS PAVLIDIS

École Doctorale Informatique, Télécommunications et Électronique

réalisée au
Laboratoire d'Informatique Paris 6



soutenue le July 2021

devant le jury composé de :

M.	Patrick Girard, DR CNRS, LIRMM, Montpellier, France	Rapporteur
M.	Manuel Barragan, CR CNRS, TIMA, Grenoble, France	Rapporteur
M.	Yann Deval, Prof., Université Bordeaux, IMS, Bordeaux, France	Examineur
M.	Wim Dobbelaere, On Semiconductor, Oudenaarde, Belgium	Examineur
M.	Andreas Kaiser, DR CNRS, IEMN, Lille, France	Examineur
M.	Stephen Sunter, Siemens Digital Industries Software, Canada	Examineur
Mme.	Marie-Minerve Louërat, CR CNRS, LIP6, Paris, France	Co-directrice de Thèse
M.	Haralampos-G. Stratigopoulos, DR CNRS, LIP6, Paris, France	Directeur de Thèse
M.	Eric Faehn, STMicroelectronics, Crolles, France	Invité

To my family that supported me during all these demanding years.

ABSTRACT

Nowadays, the number of Integrated Circuits (ICs) used in safety- and mission-critical applications, i.e., automotive, smart health-care, defense, critical infrastructure, etc., is ever increasing. These applications demand that ICs carry functional safety properties. In this thesis, we develop a Built-In Self Test (BIST) approach for Analog and Mixed-Signal (A/M-S) ICs, called *Symmetry-Based Built-In Self Test (SymBIST)*, which achieves several objectives towards the functional safety goal.

SymBIST is a generic BIST paradigm based on identifying inherent invariances and constructing them by processing internal signals. By construction, invariances should hold true only in error-free operation, while their violation points to abnormal operation. The invariances are being checked using dedicated on-die checkers. *SymBIST* is a one-stop solution for three different functional safety goals. First, it can be used for defect-oriented test with high defect coverage towards post-manufacturing test quality improvement. Second, it can be applied for on-line test concurrently with the operation towards detecting aging, latent defects, and single event upsets. Third, it can be used for fault diagnosis with high diagnosis resolution and small diagnosis cycle towards yield learning and implementing corrective actions to avoid failure re-occurrence.

We demonstrate *SymBIST* on an industrial Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC) Intellectual Property (IP). We also demonstrate the design of on-chip digital test stimulus generators and minimal re-configurations required in the defect-oriented and diagnosis operation modes. For the defect-oriented test use case, *SymBIST* offers a fast time in the order of sub- μ s and Likelihood-Weighted (L-W) defect coverage of over 86%. For the on-line test use case, *SymBIST* checks the invariances in real-time without interrupting the operation of the circuit and irrespective of the input and detects aging, latent defects, and transient errors. For the diagnosis use case, the *SymBIST* response is used as a digital diagnostic measurement. To reduce defect ambiguity, we use the same test infrastructure, but with different *SymBIST* setups. *SymBIST* shows high diagnosis resolution, i.e., 73% correct diagnosis of defects while over 96% of defects are in ambiguity groups of maximum size 5, and offers a fast diagnosis cycle in the order of a few μ s. Last but not least, *SymBIST* features no performance penalty, an area overhead of around 5%, and has a fully digital interface making it compatible with modern digital test access mechanisms based on two external pins.

RÉSUMÉ

Aujourd'hui, le nombre de circuits intégrés (CIs) utilisés dans les applications liées à des missions critiques et à la sûreté, comme l'automobile, la e-santé, la défense, les infrastructures critiques, ne cesse d'augmenter. Pour être utilisés dans ces applications, les CIs doivent présenter des propriétés de sûreté fonctionnelle. Cette thèse introduit un auto-test intégré (BIST) pour les CIs analogiques et à signaux mixtes (AMS), appelé autotest à symétrie (SymBIST), qui offre plusieurs voies pour obtenir la sûreté fonctionnelle.

SymBIST repose sur le principe du BIST et sur l'existence de signaux invariants, inhérents au CI. Les invariants retenus (tension ou courant) possèdent une valeur constante pour un fonctionnement nominal du CI, et une valeur bien distincte en cas de fonctionnement erroné. Les invariants sont vérifiés à l'aide de dispositifs intégrés spécifiques. SymBIST est une solution qui répond à trois objectifs de sûreté fonctionnelle. Il est utilisé tout d'abord pour tester les défauts du CI avec une couverture de test élevée. Il est également utilisé pour le test en ligne, lors du fonctionnement du CI, afin de détecter le vieillissement, les défauts latents et les perturbations dues à un événement aléatoire. Enfin, il est utilisé pour diagnostiquer les défauts avec une grande précision, avant de la mise en œuvre d'actions de correction.

SymBIST est démontré sur un convertisseur analogique-numérique à approximations successives. Nous présentons également des générateurs de stimuli numériques pour le test sur puce qui nécessitent une reconfiguration mineure pour passer de la détection de défauts au diagnostic. SymBIST, utilisé pour tester les défauts, a un temps de réponse inférieur à la μs et offre une couverture de test, pondérée de la vraisemblance, supérieure à 86%. Pour le test en ligne, SymBIST effectue une mesure des invariants en temps réel, sans interrompre le fonctionnement normal du circuit et indépendamment de l'entrée, pour détecter le vieillissement, les défauts latents, et les erreurs transitoires. En ce qui concerne le diagnostic, la réponse de SymBIST offre une représentation numérique du diagnostic. Pour réduire l'ambiguïté de la détection des défauts, nous tirons parti de la même infrastructure de test, en modifiant sa configuration. SymBIST offre une précision de diagnostic élevée, avec un diagnostic de défaut valide à 73%, tandis que plus de 97% des défauts appartiennent à des ensembles inférieurs ou égaux à 5 éléments, le tout en quelques μs . Enfin, SymBIST n'entraîne aucune pénalité de performance du CI, requiert une augmentation de surface d'environ 5%, et possède une interface entièrement numérique qui le rend compatible avec les mécanismes modernes d'accès aux tests numériques basés sur deux connecteurs externes.

PUBLICATIONS

The following body of work has been published in the course of the thesis at hand:

- [1] A. Pavlidis, M.-M. Louërat, E. Faehn, A. Kumar, and H.-G. Stratigopoulos, "Symmetry-based A/M-S BIST (SymBIST): demonstration on a SAR ADC IP," in *Design, Automation & Test in Europe Conference Exhibition*, 2020, pp. 282–285.
- [2] A. Pavlidis, M.-M. Louërat, E. Faehn, A. Kumar, and H.-G. Stratigopoulos, "SymBIST: symmetry-based analog and mixed-signal built-in self-test for functional safety," *IEEE Transactions on Circuits and Systems - I: Regular Papers*, 2021.
- [3] A. Pavlidis, E. Faehn, M.-M. Louërat, and H.-G. Stratigopoulos, "BIST-assisted analog fault diagnosis," in *IEEE European Test Symposium (ETS)*, 2021.

The following body of work has been published in collaboration with other projects:

- [1] S. El-Sayed, T. Spyrou, A. Pavlidis, E. Afacan, L. A. Camuñas-Mesa, B. Linares-Barranco, and H.-G. Stratigopoulos, "Spiking neuron hardware-level fault modeling," in *2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2020, pp. 1–4.
- [2] M. Elshamy, G. Di Natale, A. Pavlidis, M. Louërat, and H.-G. Stratigopoulos, "Hardware Trojan attacks in analog/mixed-signal ICs via the test access mechanism," in *2020 IEEE European Test Symposium (ETS)*, 2020, pp. 1–6.
- [3] M. Elshamy, G. Di Natale, A. Sayed, A. Pavlidis, M.-M. Louërat, H. Aboushady, and H.-G. Stratigopoulos, "Digital-to-analog hardware Trojan attacks," *IEEE Transactions on Circuits and Systems - I: Regular Papers*, 2021, (under review).

*Science may set limits to knowledge
but should not set limits to imagination.*

— *Bertrand Russell, History of Western Thoughts, p.26 1945*

ACKNOWLEDGMENTS

This thesis and a little bit more than 3-year of research finally come to an end with the support and the help of many people.

First of all I would like to thank Haralampos. Thanks for letting me find my way and for directing me whenever it was required. I was really lucky to have such a supervisor, and I am pretty sure all your PhD students will have the same opinion. The second person I would like to thank is my co-supervisor, Marie-Minerve. Marie-Minerve was always there to find a solution to any problem. Without her help our every day life in the lab would be much more difficult.

My every day life wouldn't be that amazing had I not shared my office with these amazing people. Thanks Julian, Sarah , Gabriel, Mohamed, Alan, Rodrigo, Ning, Ilias and Theofilos. We made so many wonderful things and I really miss most of them.

I would also like to acknowledge the financial support in the framework of the French National Research Agency ([ANR](#)) Electrical Diagnosis for IoT SoCs in Automotive ([EDITSoC](#)) project with N° ANR-17-CE24-0014-02. I would like to thank everyone who participated in the [EDITSoC](#) project from LIRMM and STMicroelectronics, and shared with me his knowledge. Especially, I would like to thank Eric Faehn and Anand Kumar from STMicroelectronics who worked with me in this project. I would also like to thank Stephen Sunter from Mentor, A Siemens Business, for the technical advice on the Tessent®DefectSim mixed-signal defect simulator. Finally, I would like to thank everyone I have worked with from Sorbonne University, LIP6 and EDITE. It was a pleasure working with you.

CONTENTS

I MAIN PART

1	INTRODUCTION	3
1.1	Problem Outline	3
1.1.1	The Need for Testing	3
1.1.2	The Need for Diagnosis	4
1.2	Functional Safety as a Requirement in Modern Applications	5
1.3	BIST towards Functional Safety	6
1.4	Thesis Contribution	7
1.5	Thesis Structure	8
2	PRIOR ART ON ANALOG HARDWARE TESTING AND DIAGNOSIS	9
2.1	Prior Art on Analog BIST	9
2.1.1	Generic Defect-Oriented BIST	9
2.1.1.1	Topology Modification BIST	9
2.1.1.2	Oscillation BIST	10
2.1.2	ADC BIST	11
2.1.2.1	ADC BIST Measuring Dynamic Specifications	11
2.1.2.2	ADC BIST Measuring Static Specifications	12
2.1.2.3	Other ADC BIST Approaches	13
2.1.3	RF BIST	13
2.1.4	PLL BIST	14
2.1.4.1	PLL BIST Measuring Performances	14
2.1.4.2	Defect-Oriented PLL BIST	15
2.1.5	On-line Test	15
2.1.6	BIST Evaluation	16
2.1.7	BIST for Calibration	16
2.2	Prior Art in Analog Fault Diagnosis	17
2.2.1	Rule-Based Diagnosis	17
2.2.2	Model-Based Diagnosis	17
2.2.3	Fault Dictionary-Based Diagnosis	19
2.2.4	Fault Diagnosis Using BIST	20
2.3	Conclusions	20
3	SYMBIST PRINCIPLE	23
3.1	SymbBIST Principle of Operation	23
3.2	Invariances	25
3.3	Strategy	26
3.4	Modes of Operation	27
3.5	Diagnosis Using SymbBIST	28
3.6	Conclusions	30

4	CASE STUDY AND BIST INFRASTRUCTURE	31
4.1	Case Study: SAR ADC	31
4.1.1	SAR ADC Principle	31
4.1.2	SAR ADC IP by STM	32
4.2	Invariances in SAR ADC IP	36
4.2.1	SymBIST ₁	36
4.2.2	SymBIST ₂	37
4.2.3	SymBIST ₃	38
4.2.4	SymBIST ₄	38
4.3	SymBIST Infrastructure	38
4.3.1	Test Stimulus and Re-configuration for Applying SymBIST	39
4.3.2	Checker Design	40
4.3.3	Checker Self-Test	43
4.3.4	Test Access and Control Mechanism	43
4.4	SymBIST Efficiency	46
4.4.1	SymBIST Test Time	46
4.4.2	Overheads	46
4.5	Conclusions	47
5	DEFECT COVERAGE ESTIMATION FRAMEWORK	49
5.1	Defect Simulation Workflow	49
5.2	Defect Model	49
5.3	Defect Simulator	50
5.3.1	Sampling Techniques in previous Fault Simulators before Tessent®DefectSim	51
5.3.2	Likelihood-Weighted Random Sampling (LWRS)	51
5.3.3	Additional Settings that Reduce the Defect Simulation Time	54
5.3.4	Defect Injection in Tessent®DefectSim	55
5.4	Conclusions	57
6	RESULTS OF SYMBIST APPLICATION TO THE SAR ADC IP	59
6.1	Checker Self-Test	59
6.2	Off-line Testing Using SymBIST	59
6.2.1	Test Stimulus	59
6.2.2	Setting the Comparison Window for Desired Test Coverage vs. Yield Loss Trade-off	61
6.2.3	SymBIST Transient Simulations in Off-line Test Mode	62
6.2.4	Defect Coverage Analysis	64
6.3	Online Testing Using SymBIST	67
6.3.1	Test Stimulus	67
6.3.2	Transient Error	67
6.3.3	Latent Defect	68
6.4	BIST-Assisted Analog Fault Diagnosis	68
6.4.1	SymBIST and other Diagnosis Setups	69

6.4.2	Diagnosis Results	71
6.5	Conclusions	76
7	CONCLUSIONS AND FUTURE WORK	79
7.1	Conclusions	79
7.2	Contributions of the Thesis	79
7.3	Future Work	80

II APPENDIX

LIST OF FIGURES

Figure 1.1	IC lifetime	4
Figure 2.1	Example of a bandgap circuit (from [17]) with topology modifications.	10
Figure 3.1	<i>SymBIST</i> principle.	24
Figure 3.2	Invariance in FD circuit.	25
Figure 3.3	Invariance in identical blocks.	26
Figure 3.4	Invariance with pseudo-duplication.	26
Figure 3.5	<i>SymBIST</i> strategy.	27
Figure 4.1	High-level architecture of a SAR ADC.	32
Figure 4.2	Top-level architecture of the SAR ADC IP.	33
Figure 4.3	SARCELL block architecture.	34
Figure 4.4	10-bit DAC block architecture.	35
Figure 4.5	Comparator block architecture.	36
Figure 4.6	On-chip generation of dynamic test stimulus.	40
Figure 4.7	Checker design for the invariances in Equations (4.3)-(4.4) and (4.6)-(4.8). The switches disconnect the checker when <i>SymBIST</i> is disabled or set the checker into self-test mode.	41
Figure 4.8	Checker design for the invariance in Equation (4.9).	42
Figure 4.9	Test access and control mechanism of <i>SymBIST</i> (adapted from [152]).	45
Figure 4.10	DNL of original design and original design with embedded <i>SymBIST</i>	46
Figure 4.11	INL of original design and original design with embedded <i>SymBIST</i>	47
Figure 5.1	Defect Simulation Workflow.	50
Figure 5.2	Defect model from [7].	51
Figure 5.3	Injection of defects 5.1 and 5.2.	57
Figure 6.1	Re-configuration for applying the test stimulus in off-line test mode.	60
Figure 6.2	L-W defect coverage as a function of tolerance window placement.	61
Figure 6.3	Transient simulation of <i>SymBIST</i> invariance in Equation (4.3) for different defect scenarios and blocks.	62
Figure 6.4	Transient simulation of <i>SymBIST</i> invariance in Equation (4.3) for different defect scenarios in SUBDAC ₁	63
Figure 6.5	Checker output for the short defect simulation in SUBDAC ₁ shown in Figure 6.4.	64
Figure 6.6	Schematic of V_{cm} generator.	66

Figure 6.7	<i>SymBIST</i> response to transient error injected in the DAC.	68
Figure 6.8	<i>SymBIST</i> response to transient error injected in the bandgap.	69
Figure 6.9	<i>SymBIST</i> response to a latent defect.	70
Figure 6.10	Transient simulation of <i>SymBIST</i> invariance in Equation 4.4 for $\pm 50\%$ variation defect in a resistor in Bandgap, considering two different <i>SymBIST</i> setups.	72
Figure 6.11	Transient simulation of <i>SymBIST</i> invariance in Equation (4.4) for two short defects within the SUBDAC, considering two different <i>SymBIST</i> setups.	73
Figure 6.12	Percentage of defects uniquely diagnosed and in ambiguity groups of sizes ≤ 5 and ≤ 10	74
Figure 6.13	Gradual diagnosis resolution improvement.	75
Figure 6.14	Absolute defect coverage per test setup and cumulative absolute defect coverage.	76

LIST OF TABLES

Table 4.1	Matrix showing correspondence between BIST approaches and SAR ADC IP blocks.	39
Table 6.1	L-W defect coverage results with the comparison window set at $k = 5$	65

LISTINGS

Listing 5.1	Short circuit injection in a MOS transistor	56
Listing 5.2	Injection of a +50% variation in a resistor	56

LIST OF ALGORITHMS

Algorithm 1	Random Sampling of Defects from [1]	53
-------------	---	----

ACRONYMS

AC	Alternating Current
ADC	Analog-to-Digital Converter
ANN	Artificial Neural Network
ANR	French National Research Agency
ATE	Automated Test Equipment
ATPG	Automatic Test Pattern Generation
A/D	Analog-Digital
A/M-S	Analog and Mixed-Signal
A	Analog
BER	Bit Error Rate
BIST	Built-In Self Test
CPU	Central Processing Unit
CUT	Circuit Under Test
DAC	Digital-to-Analog Converter
DC	Direct Current
DfT	Design for Testing
DNL	Differential Non-Linearity
DPPM	Defective Parts per Million
DSP	Digital Signal Processor
DTMF	Dual-Tone Multifrequency
D	Digital
EDA	Electronic Design Automation
EDITS _{oC}	Electrical Diagnosis for IoT SoCs in Automotive
ENOB	Effective Number of Bits
EVM	Error Vector Magnitude
FD	Fully-Differential
FFT	Fast Fourier Transform
HCI	Hot Carrier Injection
HD	Hamming Distance
IC	Integrated Circuit

IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IFA	Inductive Fault Analysis
INL	Integral Non-Linearity
IoT	Internet of Things
IP	Intellectual Property
ISO	International Organization for Standardization
LNA	Low Noise Amplifier
LSB	Least Significant Bit
LTL	Lower Test Limit
LWRS	Likelihood-Weighted Random Sampling
L-W	Likelihood-Weighted
MC	Monte Carlo
MEMS	Micro-Electro-Mechanical System
MLP	Multilayer Perceptron
MOS	Metal–Oxide–Semiconductor
MSB	Most Significant Bit
NBTI	Negative Bias Temperature Instability
NMOS	N-Channel Metal-Oxide Semiconductor
OBIST	Oscillation-Based BIST
op-amp	Operational Amplifier
PA	Power Amplifier
PCA	Principal Component Analysis
PDF	probability density function
PD	Pull-Down
PFA	Physical Failure Analysis
PLL	Phase-Locked Loop
PMOS	P-Channel Metal-Oxide Semiconductor
ppm	parts per million
PRBS	Pseudo-Random Bit Sequence
PSO	Phase Shift Oscillator
PU	Pull-Up
PVT	process, voltage and temperature
RF	Radio Frequency

RL	Relative Likelihood
RMS	Root Mean Square
RSN	Re-configurable Scan Network
SAR	Successive Approximation Register
SC	Switched-Capacitor
SFDR	Spurious-Free Dynamic Range
SH	Sample-and-Hold
SIB	Segment Insertion Bit
SiP	System-in-Package
SI	Scan In
SNDR	Signal-to-Noise-and-Distortion Ratio
SNR	Signal-to-Noise Ratio
SoC	System-on-Chip
SO	Scan Out
SPICE	Simulation Program with Integrated Circuit Emphasis
Std.	Standard
STM	ST Microelectronics
SymBIST	Symmetry-Based Built-In Self Test
TDC	Time-to-Digital Converter
Tddb	Time-Dependent Dielectric Breakdown
TDR	Test Data Register
THD	Total Harmonic Distortion
USER-SMILE	Ultrafast Stimulus Error Removal and Segmented Model Identification of Linearity Errors
UTL	Upper Test Limit
VCO	Voltage Controlled Oscillator
VLSI	Very Large-Scale Integration

Part I

MAIN PART

INTRODUCTION

1.1 PROBLEM OUTLINE

Nowadays, electronic devices are widely used in almost every aspect of everyday life. These are products ranging from daily used devices, i.e., mobile phones and laptops, to more sophisticated and complicated automated equipment in health care, automotive, aeronautics and defense industry. The extensive use of these products has only been made possible with the advances in microelectronics.

Principal component of every electronic device is the **IC**, which is a device packing numerous electronic components like transistors, resistors and capacitors on one single chip. Modern systems include increasing numbers of **ICs**, i.e., the number of **ICs** in a typical automobile today exceeds 400 and continues to increase. Additionally, the trend nowadays is towards heterogeneous System-on-Chip (**SoC**) devices where different **ICs**, i.e., analog, digital processor, memory, etc., are integrated onto the same substrate. This reduces the system size and power consumption and also increases the speed of operation. The **IC** design complexity and number of **ICs** on a **SoC** are increasing rapidly and the technology limits are pushed to increase performance and market competitiveness.

Although the expected lifetime of the majority of the **ICs** is usually several years, **ICs** might fail due to various failure mechanisms. The main categories are the following: (a) defects induced during the several steps of the manufacturing process, i.e., dust particles, over/under etching etc.; (b) defects induced during packaging; (c) global process variations affecting complete regions of a wafer; (d) local process variations causing component mismatch; (e) defects caused by component ageing, i.e., electromigration, Negative Bias Temperature Instability (**NBTI**), Hot Carrier Injection (**HCI**), Time-Dependent Dielectric Breakdown (**TDDDB**). A detailed survey regarding the failure mechanisms of electronic components is given in [2] and in [3].

The numerous failure mechanisms generate the need for testing in order to have a robust **IC**. Moreover, it is also very important after a failure is detected to identify the mechanism that caused the **IC** to fail.

1.1.1 *The Need for Testing*

The process from the initial conception of an **IC** design until the actual use of it in an application is lengthy and consists of the high-level stages, as

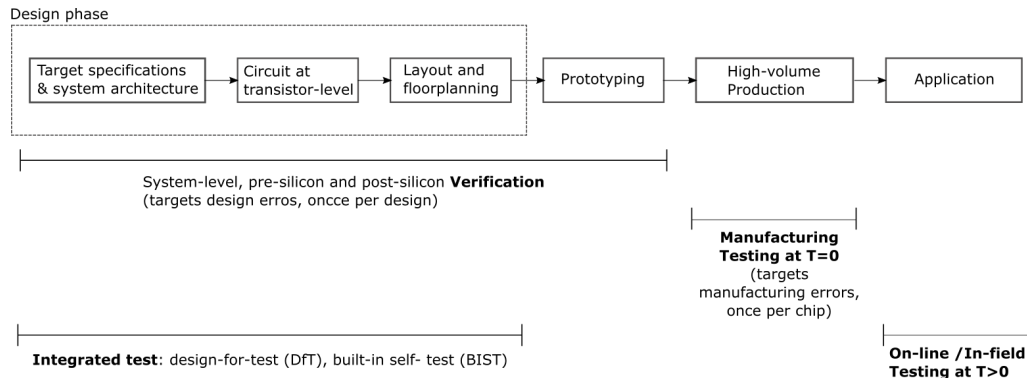


Figure 1.1: IC lifetime

illustrated in Figure 1.1. Testing is undertaken on an IC at different points in time from design concept through to full fabrication until the actual use in the application. Design phase starts with system architecture, circuit design at transistor level and layout design. Once this design phase is over and the chip is prototyped, it needs to pass the first type of test which is the system-level, pre- and post- silicon verification. The verification targets at design errors and runs once per design. If the design passes the verification, then a high-volume production of chips follows. At this stage, manufacturing tests are designed to ensure that every part of an IC is functional and that each chip meets the datasheet requirements and can be used adequately in real applications. Unlike verification which is performed once per design, testing is performed for every fabricated chip. Thus, it is crucial to minimize test application time and cost while guaranteeing the quality of passing chips. If an IC fails to pass any of the post-manufacturing tests, then it will be discarded. But even if a chip passes successfully the post-manufacturing test phase and meets all specifications at time zero, it can fail during the application, for example due to aging and environmental stress. Thus, especially for critical applications, there is a need for on-line testing in order to guarantee safety. On-line test means that the chip tests itself without depending on external test equipment. It can take place in idle times or concurrently with the operation as long as it is non-intrusive. Although the on-line testing only occurs in the latest stage, an integrated test needs to be considered already from the design phase. The integrated test consists of Design for Testing (DFT) and BIST approaches which need to be co-designed with the initial design.

1.1.2 The Need for Diagnosis

The next step after detecting a failure in an IC is actually to diagnose it. Fault diagnosis refers to the analysis performed to identify the root cause of failure that occurred in an IC either during manufacturing or in the

field during normal operation following a customer return. The output is the isolation of the defect responsible for the failure and its localization at transistor-level.

Diagnosis is a crucial step in a product life-cycle. It can reveal important statistics, such as defect distribution and yield detractors. In this way, it provides valuable feedback for improving the design to prevent failure re-occurrence and expand the safety features, and also for evaluating and improving the quality of post-manufacturing tests.

In a first step, diagnosis generates a set of candidate defects based on diagnostic measurements or observed syndromes. In a second step, the IC is physically examined, for example using a thermal camera, to highlight anomalies in the operation and narrow down further the set of candidate defects. In the third and last step, the IC is submitted to Physical Failure Analysis (PFA) where de-layering and cross-sectioning of the die is performed to confirm the defect using imaging. Since PFA is destructive and irreversible, ideally the first step should pinpoint the actual defect. However, very often the first step results in an ambiguity group of candidate defects. According to industrial experience, its size should be less than 5-10 candidate defects to increase the PFA success rate.

Diagnosis metrics include: (a) resolution, i.e., the size of the ambiguity group; (b) accuracy, i.e., whether a reported candidate corresponds to the actual defect; and (c) diagnosis cycle time, i.e., the time required to complete the diagnosis since the number of diagnoses performed per week per design can be in the order of thousands [4].

1.2 FUNCTIONAL SAFETY AS A REQUIREMENT IN MODERN APPLICATIONS

The number of ICs used in safety- and mission-critical applications, i.e., automotive, smart health-care, defense, critical infrastructure, etc., is ever increasing. This requires that ICs should be robust and defect-tolerant to meet the functional safety criteria. This implies that more ICs should be equipped with functional safety mechanisms. Moreover, modern systems include increasing numbers of ICs, i.e., the number of ICs in a typical automobile today exceeds 400 and continues to increase. This implies that the quality of individual ICs must increase to prevent decrease in the system's functional safety. The quality of the IC is measured by Defective Parts per Million (DPPM) and more specifically, it is desired that failure rate is in the order of sub-parts per million (ppm) [5].

Functional safety refers to the requirements to: (a) avoid IC malfunctions in the field by following robust design guidelines; (b) perform comprehensive post-manufacturing testing with proven high defect coverage and effective outlier screening; (c) detect reliability hazards in the

field before failures occur; (d) prevent failures in the field that could be detrimental; (e) detect failures in the field when they occur; (f) adding automatic protection to control failures when they occur in the field and recovering from them at an acceptable time span using fault-tolerance, self-repair, or self-healing principles; (g) ensure correct and uninterrupted operation in response to all inputs even under non-intended use or sometimes even misuse. Functional safety is regulated by standards depending on the application domain, e.g. International Organization for Standardization (ISO) 26262 for automotive and International Electrotechnical Commission (IEC) 61508 for industrial systems.

1.3 BIST TOWARDS FUNCTIONAL SAFETY

The purpose of BIST is to migrate part of the functionality of the Automated Test Equipment (ATE) onto the chip with the aim to facilitate test and reduce test cost. At an abstract level, BIST consists of (a) embedded test instruments, whose role is to generate test stimuli, perform measurement acquisition, and process measurements for building a comprehensive test response, and (b) a mechanism for accessing and controlling these test instruments from external pins. BIST can be defect-oriented, targeting detection of structural defects, or functional, in which case it targets measuring performances that are promised in the datasheet of the IC.

BIST is traditionally used for post-manufacturing testing, but it has also found application for silicon debugging, for guiding on-chip calibration, for facilitating fault diagnosis, and for on-line test where the status of the IC is checked in the field concurrently with the application or in idle times.

Functional safety has emerged as a new major application domain for BIST. In this context, BIST can help gaining better insight into the IC and improving defect coverage. In fact, many case studies have shown that the standard specification tests performed on an ATE offer no guarantee to meet the quality requirement [6]. A defect is always considered a potential threat and reliability hazard. An IC with a detected defect should be preferably discarded for safety reasons [5], [7], [8]. Even if from a functional viewpoint the performance complies with the specifications promised in the datasheet during post-manufacturing test time, i.e., time zero of the application, a defect may manifest itself later in the field of application referred to as a latent defect [9]. To this end, performing defect-oriented BIST on top of the standard specification tests and proving high defect coverage can address safety concerns. Thus, defect-oriented BIST is no longer expected to replace standard specification tests, which was the use case the community was hoping for in the early days, but it aims at enhancing confidence in ICs passing the test. The premise of BIST in this context is that it can detect defects that are reliability hazards, will

manifest themselves later in the field, or will be triggered in the context of system operation in the field, provoked by environmental stress, for example heat, humidity or vibration. In the same context, a BIST that can be performed on-line in the field concurrently with the application or in idle times can help detecting reliability hazards and failures at the time of occurrence [10]. It is also a key block in feedback loops that enable fault-tolerance, self-repair, or self-healing. Finally, it can facilitate fault diagnosis to understand the root-causes of errors towards improving the design and manufacturing processes [11].

It should be mentioned that there are additional approaches towards meeting quality requirements before deployment in the field, i.e., burn-in stress [12] and outlier screening [13]–[15].

Embedding BIST into A/M-S ICs is a complex task presenting several challenges. In particular: (a) the BIST circuitry should be transparent to the IC without degrading its performance and without requiring significant re-configuration or re-design; (b) the BIST circuitry should incur low and justifiable area overhead; (c) for defect-oriented BIST the simulation should be fast for enabling large-scale defect simulation in reasonable time and for allowing to perform defect simulation multiple times for several refined BIST versions and for comparing with other BIST approaches; (d) for on-line BIST real-time response should be fast for enabling low-latency error detection; (e) the BIST ideally should be flexible and reusable for different IC classes and different architectures within each IC class; (f) the BIST principle ideally should have proven quality before moving to high-volume production; (g) the BIST circuitry should be more robust than the IC having low failure probability, which typically implies that ideally the BIST wrapper should be fully digital; (h) the BIST ideally should be portable from one technology node to another without requiring significant re-design; and (i) the BIST instruments ideally should be interfaced to standard digital test access mechanisms.

1.4 THESIS CONTRIBUTION

In this thesis, a novel BIST paradigm for A/M-S ICs, called *SymBIST* is proposed. *SymBIST* exploits inherent symmetries in an A/M-S IC to construct signals that are invariant by default, and subsequently checks those signals against a tolerance window. Violation of invariant properties points to the occurrence of a defect or abnormal operation. *SymBIST* is designed to serve as a functional safety mechanism. It is reusable ranging from post-manufacturing test, where it targets defect detection, to on-line test in the field of operation, where it targets low-latency detection of transient failures and degradation due to aging. We demonstrate *SymBIST* on real SAR ADC IP by ST Microelectronics (STM). *SymBIST* has the following appealing attributes:

1. It offers high defect coverage.
2. It offers a fast test cycle, which is a key for making the defect simulation tractable.
3. It incurs low area overhead.
4. It achieves zero performance penalty.
5. It has a fully digital interface making it compatible with modern 2-pin test access and control mechanisms, e.g. Institute of Electrical and Electronics Engineers (IEEE) 1687.

SymBIST, originally proposed for defect-oriented post-manufacturing test and on-line test, can also be seamlessly reused for the purpose of diagnosis. Fault diagnosis methodologies for analog circuits lag far behind those for their digital counterparts. BIST can offer better insights into the circuit and, thereby, can assist diagnosis towards resolving ambiguity groups. Using *SymBIST* for diagnosis we demonstrate:

1. High diagnosis resolution, i.e., a high percentage of uniquely diagnosed defects and ambiguity groups of small size.
2. Fast diagnosis cycle in the order of μs .

1.5 THESIS STRUCTURE

The thesis is structured as follows. In Chapter 2, we provide a review of previous work (a) on A/M-S BIST with a focus on ADCs and (b) on diagnosis approaches for A/M-S ICs. In Chapter 3, we present the *SymBIST* concept. In Chapter 4, we present the SAR ADC case study, that is used for demonstrating *SymBIST*. In Chapter 5, we discuss the defect simulation framework, including the defect modeling approach and the mixed-signal defect simulator used to accelerate and automate defect simulations. In Chapter 6, we present the results of applying *SymBIST* for off-line defect-oriented post-manufacturing test, on-line test and fault diagnosis. Chapter 7 concludes this thesis and discusses future work and perspectives.

PRIOR ART ON ANALOG HARDWARE TESTING AND DIAGNOSIS

2.1 PRIOR ART ON ANALOG BIST

There is a large body of literature on [A/M-S BIST](#). In general, [A/M-S BIST](#) is proposed for mainly three uses, namely: (a) defect-oriented test; (b) direct on-chip measurement of performances, e.g. functional test; and (c) on-line test, either concurrently with the operation or in idle times. Additional uses include calibration for yield boosting and fault diagnosis. For any of these uses, [BIST](#) is in general specific to the circuit class and very often specific to different architectures within a given circuit class. Furthermore, [BIST](#) is in general designed and/or demonstrated for one use only.

2.1.1 *Generic Defect-Oriented BIST*

For defect-oriented test, generic [BIST](#) proposals include topology modification by inserting Pull-Up ([PU](#)) or Pull-Down ([PD](#)) transistors and oscillation-based test.

2.1.1.1 *Topology Modification BIST*

Topology modification [BIST](#) originally proposed in [16] is enabled by the addition of [PD](#) and [PU](#) transistors. A [PD](#) transistor connects a circuit node to ground, while a [PU](#) transistor connects a circuit node to the power supply. [PD](#) and [PU](#) transistors are activated by applying a logic 1 and 0 at their gates, respectively. In general, given a Circuit Under Test ([CUT](#)) with M nodes, in each node a [PD](#) and/or a [PU](#) transistor can be added. Therefore, the total number of possible [PU](#) and [PD](#) transistors in a [CUT](#) is 2^M . If N [PU](#) and [PD](#) transistors are added in total, where $N \leq 2^M$, then the circuit can be configured into 2^N topologies, including the original one when all [PD](#) and [PU](#) transistors are deactivated. The underlying principle is that by these re-configurations we are able to expose the presence of additional defects that are undetectable in the original topology.

Figure 2.1 illustrates an example of topology modifications in a bandgap circuit. In this case, a [PU](#) transistor controlled by signal B1 brings the node V_A to VDD and a [PD](#) transistor controlled by signal B2 brings the node V_B to ground.

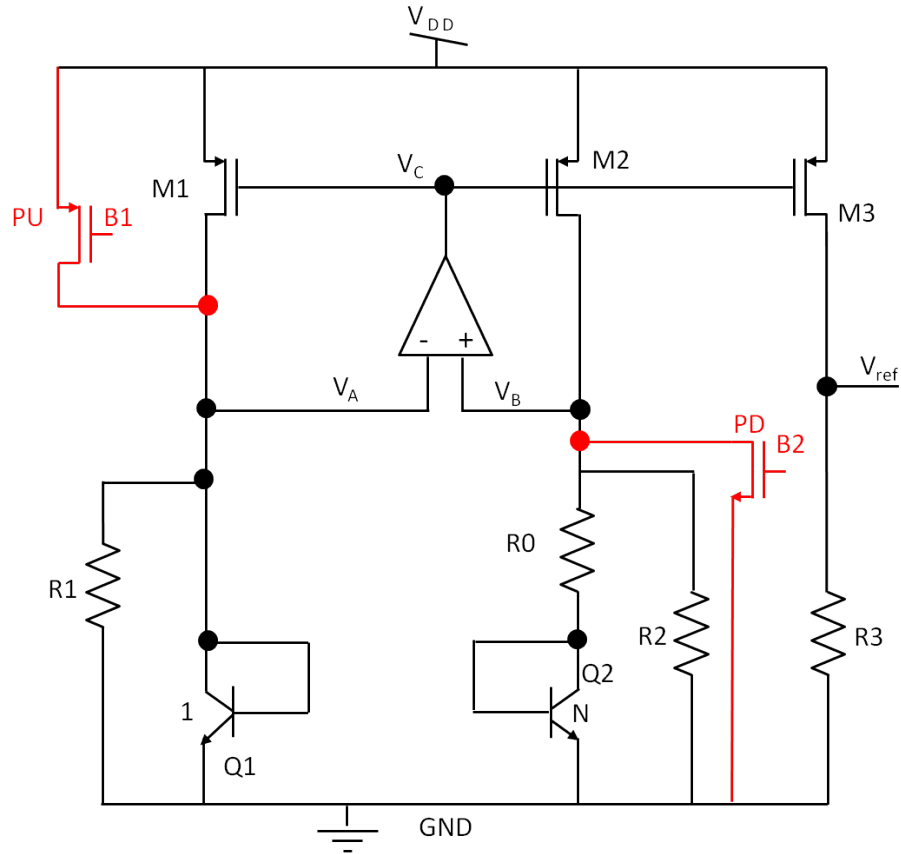


Figure 2.1: Example of a bandgap circuit (from [17]) with topology modifications.

2.1.1.2 Oscillation BIST

Oscillation-Based BIST (OBIST) has been widely used for testing different classes of ICs, i.e., filters [18]–[26], ADCs [27]–[29], Radio Frequency (RF) circuits [30] and Micro-Electro-Mechanical Systems (MEMSs) [31]. The underlying idea of OBIST is to apply feedback loops inside the CUT and convert it to multiple oscillators, where an oscillator is an electronic circuit that produces a sinusoidal or pulse wave signal without any input stimulus other than the Direct Current (DC) supply. The oscillation frequency and magnitude of a block configured into an oscillator depends on its components' parameters. Any deviation of the oscillation frequency or magnitude away from the nominal expected values points to defect detection. In [32], an algorithm to estimate the limits of the test metrics is described in order to achieve the desired test coverage versus yield loss trade-off.

As a few examples, in [22], a Switched-Capacitor (SC) biquadratic filter is considered as a case study and a non-linear feedback is applied to convert the filter into an oscillator. In [28], OBIST is shown for a $\Sigma\Delta$ ADC, where two oscillation frequencies are extracted and used together

with the output of a functional test to distinguish the defect-free circuits from the defective ones. In [33] the required re-configurations applied to the analog subsystem of a Dual-Tone Multifrequency (DTMF) receiver to implement OBIST are described.

Overall, oscillator-based test has been a very popular test mechanism showing numerous advantages: (a) it is applicable to a large number of IC classes, (b) there is no requirement for an external test stimulus since the CUT is used in a feedback loop, (c) digital test signatures can be extracted off-chip from processing the oscillation frequency and magnitude, (d) test parameters can be extracted as an average over several periods to minimize the noise impact, (e) it is very efficient for catastrophic and parametric faults [34].

2.1.2 ADC BIST

The majority of the BIST techniques for ADCs that have been published up to date are functional. The main reason is that a single simulation of the ADC already requires a very long time in the order of hours, thus defect simulation time is prohibitive to be able to demonstrate defect-oriented BIST.

ADCs are characterized by dynamic specifications, such as (a) Signal-to-Noise Ratio (SNR), (b) Signal-to-Noise-and-Distortion Ratio (SINDR), (c) Total Harmonic Distortion (THD), (d) Effective Number of Bits (ENOB), (e) Spurious-Free Dynamic Range (SFDR), etc. and static specifications such as (a) Differential Non-Linearity (DNL), (b) Integral Non-Linearity (INL), (c) offset (d) gain, etc.

2.1.2.1 ADC BIST Measuring Dynamic Specifications

The standard approach to measure the dynamic specifications of an ADC consists of applying a high-resolution sinusoidal at the input and computing the Fast Fourier Transform (FFT) at the output.

However, there is a requirement for the resolution of the sinusoidal input, which typically needs to be at least two bits higher than the effective resolution of the ADC. This poses a great design challenge for BIST implementation and several methods have been proposed to overcome this challenge.

A classical approach for generating on-chip an analog sinusoidal is to employ a closed-loop oscillator that involves a highly selective band-pass filter and a comparator [35]. To improve the resolution of the sinusoidal generators they can be combined with harmonic cancellation techniques [36]–[38]. In [36], a technique is proposed to suppress the low-frequency harmonics with a digital harmonic cancellation block and the high-frequency ones with a passive filter. In [37], a method is de-

scribed to compute the weights for summing the outputs of a Phase Shift Oscillator (PSO) to cancel the harmonics. In [38], numerous harmonic cancellation approaches are presented with the aim to simplify the on-chip implementation of the scaling weights.

Another classical approach is to employ an open-loop oscillator. The starting point is to use an ideal $\Sigma\Delta$ modulator in software that converts a high-resolution sinusoidal to a bit stream which, thereafter, is loaded and periodically reproduced in an on-chip circular shift register [39]. The bit stream can be converted on-chip to a high-resolution sinusoidal by passing it through a 1-bit Digital-to-Analog Converter (DAC) followed by a low-pass filter to remove the quantization noise. Interestingly, in the case of SC $\Sigma\Delta$ ADCs, the bit stream can be fed directly into the modulator by adding simple circuitry at its input [40]–[43].

Another challenge of an ADC BIST approach is the analysis of the test response since performing FFT on-chip incurs a high area overhead [44]. If the FFT cannot be performed in the Digital Signal Processor (DSP), then in the case of a full BIST implementation, the FFT algorithm needs to be replaced with an alternative less computationally-intensive algorithm, such as the sine-wave fitting algorithm [45].

2.1.2.2 ADC BIST Measuring Static Specifications

The standard approach to measure the static specifications of an ADC is to apply a ramp at the input and obtain the histogram of the number of occurrences of each code at the output.

Again, the first challenge concerns the on-chip test stimulus generation. Adaptive ramp generators can be employed, as proposed in [46], [47], or, alternatively, an exponential waveform can be used [48]. Another approach is based upon first identifying and computationally removing the source non-linearity, and then, accurately estimating the ADC static performances [49].

Regarding the test response analysis, it is required to store both the experimental and the reference histograms and use the DSP to perform the comparison. An efficient BIST implementation of the histogram analysis is proposed in [50]. For ADCs having a repetitive structure, such as pipeline, cyclic, and successive approximation ADCs, what is broadly known as reduced code testing can be applied [51]–[55]. Only a few codes need to be judiciously targeted and from this information the complete histogram can be extrapolated. For such ADCs that present an INL curve that is somewhat periodic and can be segmented, the Ultrafast Stimulus Error Removal and Segmented Model Identification of Linearity Errors (USER-SMILE) algorithm can be applied [56]. This algorithm relaxes the test stimulus linearity requirement by employing two non-linear input signals and, thereafter, it uses some post-processing to remove the

test stimulus error and recover accurate static test accuracy by orders of magnitude faster than the standard histogram method.

2.1.2.3 *Other ADC BIST Approaches*

Instead of targeting a BIST approach aiming at measuring directly the dynamic and static specifications of the ADC, it is also possible to consider a BIST approach where the aim is to obtain measurements that reveal important design parameters, such as the poles and settling errors of the integrators in the case of $\Sigma\Delta$ ADCs [57]. An approach proposed also specifically for pipeline ADCs is to reconfigure consecutive pipeline stages to form $\Sigma\Delta$ modulators and then test instead the $\Sigma\Delta$ modulators through digital means [58].

2.1.3 *RF BIST*

For RF transceivers, a common BIST technique consists in creating a loop-back connection between the transmitter and the receiver, in order to test the whole RF transceiver. The Power Amplifier (PA) output of the transmitter is connected to the Low Noise Amplifier (LNA) input of the receiver using an attenuator and a switch. Next, digitally modulated baseband signals are transmitted and the baseband response signals are used to evaluate the response, i.e., measure the Error Vector Magnitude (EVM), extract parameters from the response, etc. [59]–[66].

Sensor-based testing is another common BIST technique, which can be employed inside the blocks of RF transceivers.

A type of sensor that is commonly used to implement BIST is the current sensor [67]–[69]. The parasitic resistor of the line provides a voltage drop which unbalances the CUT. The output current is proportional to the RF power supply current of the CUT and the output current is switched to the input of an envelope detector to obtain a DC signature, which nevertheless carries information about the RF amplitude of the power supply current.

A second way to extract DC signatures is through envelope (a.k.a. amplitude or power detectors) detectors that transfer directly information about the RF amplitude by monitoring current or voltage of internal nodes [70]–[75].

There exist also non-intrusive sensors, i.e., process variation-aware sensors [68], [76], [77] and temperature sensors [78]–[81], that can extract information without being electrically connected to the CUT. The process variation-aware sensors monitor the process variations which are correlated to the performances of the CUT. Thereafter, the performances of the CUT can be predicted from the sensor measurements using the alternate test paradigm [82]–[92]. However, defect detection is not feasible with

variation-aware sensors since they are not electrically connected to the CUT. To overcome this problem, temperature sensors can be also used to provide defect detection by monitoring the temperature of the CUT [93]. When the CUT is powered-on, it is self-heated due to the power dissipation and the heat is diffused in the substrate creating a temperature profile in the vicinity of the CUT. In the presence of a defect, the power dissipation will alter, resulting in a shift of the temperature profile. A temperature sensor captures the temperature shift and indicates the presence of the defect.

2.1.4 PLL BIST

A Phase-Locked Loop (PLL) is a fundamental building block used to synthesize clocks for data synchronization and to provide the frequency sources for up-conversion and down-conversion in RF transceivers. For PLLs, BIST techniques exist for measuring the performances, i.e., jitter, loop gain, lock time, etc., and for facilitating defect-oriented tests.

2.1.4.1 PLL BIST Measuring Performances

A key performance parameter of a PLL is the high-frequency jitter. Jitter must be low to guarantee data synchronization and low Bit Error Rate (BER) in communication systems. There are a few ways to define the PLL's jitter [94]: (a) timing jitter, defined as difference between the edge timing and the ideal edge timing, (b) period jitter, defined as the difference between each period and the average period, and (c) cycle-to-cycle jitter, defined as the difference between the period and the preceding period. Measuring jitter off-chip is difficult and costly since high speed sampling oscilloscopes are very costly and the measurement requires long test times. Moreover, the signal gets further polluted when it is extracted off-chip due to pad, probing, and coupling parasitics. Therefore, there is a need for designing a BIST to measure jitter on-chip.

There is a large body of literature describing on-chip jitter measurement [94]–[100]. For example in [100], a jitter measurement technique is proposed, where the PLL's response is undersampled and the count of unstable bits at the clock rising edges is correlated to the high-frequency jitter. Instead of measuring the jitter it is also possible to estimate the transfer function of the jitter [101], [102].

Finally, there have been several approaches for measuring other PLL parameters such as the Voltage Controlled Oscillator (VCO) gain, lock time, etc. [95], [103]–[105]. In [95], logic gates are used to develop a BIST mechanism to measure the lock range, lock time, loop gain, as well as the Root Mean Square (RMS) jitter.

2.1.4.2 Defect-Oriented PLL BIST

Defect-oriented BIST approaches for PLLs have been proposed in [106]–[108]. In [108], a Pseudo-Random Bit Sequence (PRBS)-based test is proposed, where a digital PRBS is injected in the charge pump perturbing the PLL, and the cross-correlation of the PRBS pattern with the output of the phase/frequency detector is considered for defect detection. The major challenge of this technique is to select the most suitable injection and observation points in the loop, so as to detect defects in the whole bandwidth.

2.1.5 On-line Test

BIST can also be used to perform on-line test in the field either concurrently with the application or in idle times. In the concurrent test mode, no re-configurations are allowed in the CUT and its inputs are defined by the application that it is used. On the contrary, in on-line test mode during idle times, it is possible to perform non-intrusive re-configurations in the CUT and apply a desired test stimulus, however all these actions need to be performed by the BIST itself.

For on-line test, generic BIST includes duplication or triple modular redundancy, but these approaches are very costly. Cost-effective BIST for on-line test has been proposed only for linear time-invariant circuits, i.e., filters, using pseudo-duplication concepts [109]–[111] and for Fully-Differential (FD) circuit implementations [112]–[114]. Regarding linear time-invariant circuits, in [109], a strategy is proposed for switched-capacitor filters where a programmable biquad that can mimic every filter stage is configured to monitor successively the filter stages. In [110], the matrices of the state-variable equations are encoded into a continuous checksum which is implemented by small extra hardware. In [111], it is shown how to generate with small extra hardware an estimator that monitors some observable nodes of the circuit and, once fully connected to the circuit, produces an output that converges exponentially fast to the output of the circuit and follows the output for any input change.

For FD circuits, node pairs carrying FD signals should satisfy the following property:

$$V_1 + V_2 = 2V_{cm} \quad (2.1)$$

where V_1 , V_2 are the node voltages and V_{cm} is the common-mode voltage. An on-line test approach can be built based on monitoring that this property always holds. Any deviation of the sum from the common-mode voltage points to erroneous operation. In practice the sum is expected to stay within a tolerance window due to noise and voltage and temperature variations.

Analog comparators or checkers that compare duplicate or **FD** signals with adaptive tolerance windows are proposed in [113], [114].

2.1.6 *BIST Evaluation*

Any functional **BIST** approach should be evaluated in terms of parametric test metrics, i.e., resultant test escapes and yield loss, before moving to high-volume production. Ideally, the evaluation should take place at simulation level so as to provide early feedback towards refining the **BIST** and making a decision whether to pursue the **BIST** or summarily reject it. The challenge is that analog simulations are time-consuming. In fact, test escape and yield loss events have low-probability and a Monte Carlo (**MC**) analysis with a reasonable number of runs will produce very few such events, if any at all, so as to be able to compute test metrics that have statistical significance. Analog test metric estimation algorithms for assessing functional **BIST** techniques have been proposed based on parametric density estimation [115], Copulas theory [116], and statistical blockade and extreme value theory [117]–[119]. In [120], a practical simulation flow for evaluating test metrics for **ADCs** and, in general, for circuits with long simulation times, is proposed based on statistical behavioral modeling and fast behavioral-level **MC** using the statistical blockade technique.

2.1.7 *BIST for Calibration*

Calibration schemes are oftentimes utilized in **A/M-S** and **RF ICs** with the aim to boost yield, i.e., by compensating against process variations and non-idealities.

At a minimum, a calibration scheme utilizes digitally controlled tuning knobs that act on the circuit performances. Tuning knobs may include bias voltages, current sources, or single tunable components, such as resistors, capacitors, and varactors.

The standard calibration algorithm consists in multiple testing/tuning iterations where in each step the performances are measured and the next best tuning knob setting is decided based on some optimization algorithm.

The calibration scheme may also utilize **BIST** for performance measurement which can speed-up the test cycle and alleviate the dependence on complex **ATE**. For example, one-shot calibration schemes based on process-variation-aware sensors and machine learning are proposed in [121], [122].

The most advance calibration schemes are fully implemented on-chip rendering the circuit self-healing. These schemes can be also used during the lifetime of the circuit to compensate against aging. They comprise

tuning knobs or actuators, sensors for extracting information-rich measurements or directly the performances, and a digital processor engine that maps the outputs of the sensors to tuning knob values and aims at driving the optimization so as to identify a good balance among multiple competing performance goals [123]–[125].

2.2 PRIOR ART IN ANALOG FAULT DIAGNOSIS

While for digital ICs there exist several in-house frameworks and commercial diagnosis Electronic Design Automation (EDA) tools [4], [126]–[130], for analog ICs there is neither a commercial tool nor a standardized diagnosis approach. Analog fault diagnosis is still a manual, tedious, and time-consuming process very often resulting in large ambiguity groups and no actionable diagnosis information. In fact, there is a vast literature on analog fault diagnosis, yet none of the proposed solutions has matured enough to meet industry standards.

2.2.1 Rule-Based Diagnosis

The traditional approach is a rule-based system which takes the form “IF symptom(s) THEN fault(s)”. In this approach, rules are developed to match the symptoms of an IC to an actual fault.

In [131], assembly faults in RF circuits embedded in System-in-Package (SiP) are diagnosed. These SiP assembly faults are:

1. Power supply faults.
2. Bias network faults.
3. RF load component faults.
4. Signal path faults.

The symptoms used to diagnose the above mentioned faults are: (a) DC measurements, (b) current consumption, and (c) noise power. In general, it is difficult to acquire the knowledge to build a rule-based system for diagnosis.

2.2.2 Model-Based Diagnosis

Model-based diagnosis approaches have also been studied extensively. The idea is to first build a model linking diagnostic measurements to circuit parameters. Then, given the diagnostic measurements from the real failing device, the model is used to identify the faulty circuit parameter, as well as its deviation from the nominal value. The model can be

constructed using nonlinear circuit equations [132], sensitivity analysis [133], [134], regression [135], [136], or behavioral modeling [137], [138].

In [132], fault diagnosis using frequency measurements for linear systems is proposed. However, even for a linear system described by a set of linear equations

$$m = f(p) \quad (2.2)$$

the corresponding diagnosis equations

$$p = f^{-1}(m) \quad (2.3)$$

are non-linear.

In [133] and [134], the sensitivity matrix S is used to build the diagnosis equations. The sensitivity matrix is defined as the relation between the vector of the normalized output measurement deviations, denoted by $\frac{\Delta M}{M}$, and the vector of the normalized component deviations, denoted by $\frac{\Delta P}{P}$:

$$S \cdot \frac{\Delta P}{P} = \frac{\Delta M}{M} \quad (2.4)$$

Assuming that we have N output measurements and K components, the elements that construct the sensitivity matrix are given by:

$$s_{P_j}^{M_i} = \frac{\frac{\Delta M_i}{M_i}}{\frac{\Delta P_j}{P_j}}, \quad i = 1, \dots, N, \text{ and } j = 1, \dots, K \quad (2.5)$$

where $s_{P_j}^{M_i}$ is the differential sensitivity of the output measurements M_i with respect to component P_j for a given input signal.

Diagnosis aims at extracting the component parameters, therefore we are interested in solving:

$$\frac{\Delta P}{P} = S^{-1} \cdot \frac{\Delta M}{M} \quad (2.6)$$

In [135] and [136], non-linear regression models are constructed for the purpose of diagnosis. Firstly, in the pre-diagnosis phase, regression models mapping diagnostic measurements to component parameters are trained using simulations. After the device has failed, the same diagnostic measurements are obtained from the CUT and given as an input to the regression model that predicts the components' deviation.

Another model-based diagnosis approach found in literature aims at building a behavioral model of the IC [137], [138]. The underlying idea is that all possible faults are injected in the behavioral model until the test parameter of the model matches the one of the actual faulty chip. The fault injection stops when a match is found and the fault causing the failure is identified. More specifically, in [137] a mathematical model was constructed to estimate the behavior of a biquad filter, whereas in [138] a behavioral model of a second-order SC $\Sigma\Delta$ ADC was designed, including the critical non-idealities, i.e., clock jitter, KT/C noise, and Operational Amplifier (op-amp) non-idealities .

The model-based diagnosis approach can be used for diagnosis of soft faults, i.e., deviations of circuit parameters; however, it is recognized that hard faults, i.e., defects such as short- and open-circuits, are the main root cause of IC failures [139]. Besides, it is challenging to construct a model that faithfully reproduces the analog IC behavior.

2.2.3 Fault Dictionary-Based Diagnosis

Perhaps the most common diagnosis approach is based on the use of a fault dictionary. Given a list of defects generated by Inductive Fault Analysis (IFA) [139], one defect is injected at a time in the netlist, and the IC is simulated to obtain the diagnostic measurement pattern. The fault dictionary contains the pairs of defects and diagnostic measurement patterns. Then, the diagnostic measurement pattern from the failed IC is mapped to one of the logged diagnostic measurement patterns of simulated defects based on some similarity metric. The mapping can be established after training a multi-class classifier using the fault dictionary dataset [11].

A growing body of literature has evaluated different combinations of (a) test stimuli, (b) measurement post-processing, (c) classifier types, as a way to resolve ambiguity groups [140]–[144].

In [140], the authors used white noise input as test stimulus and an Artificial Neural Network (ANN) classifier to diagnose faults. The white noise targets to excite all the transistors and scale the frequency response of the circuit. The training data of the ANN will contain diagnostic measurements from both faulty and fault-free circuit instances and multiple networks can be trained in order to improve the diagnosis results. During diagnosis these networks are used to localize the fault that has occurred.

In [141], an ANN classifier was proposed that uses wavelet decomposition as feature pre-processing. The features used by the ANN for the classification are generated after wavelet transformation is performed in the output of the CUT and the wavelet coefficients are then processed by Principal Component Analysis (PCA) for feature selection and are

normalized. Wavelet decomposition transforms time-varying signals into wave-like signals that oscillate, known as wavelets.

In [142], the proposed test stimulus is power supply ramping and the signatures used to train the ANN classifier are constructed by the power supply current response at different time stamps. In [143], a sinusoidal input test stimulus is used and FFT is performed in the output of the CUT. Then, the harmonics of the output are used as features and are classified using a two-stage Multilayer Perceptron (MLP) ANN.

Finally, there exist methods for selecting the minimal number of multifrequency test sets [145] and the optimum test points based on the entropy [144].

The challenge with the fault dictionary approach is the long fault simulation time. However, the recently proposed analog fault simulators [7], [8], [146] can help in speeding up the fault dictionary development, as demonstrated in [147].

2.2.4 Fault Diagnosis Using BIST

Traditionally, BIST aims at adding on-chip test structures to facilitate and reduce the cost of post-manufacturing test. BIST can offer better insights into the circuit and, thereby, can assist diagnosis towards resolving ambiguity groups. Despite this, in the literature there are few examples of using BIST in this field. In [148], BIST is used for diagnosis for the class of fully differential analog filters and in [149] an Automatic Test Pattern Generation (ATPG) algorithm is described for generating the sinusoidal test stimulus.

2.3 CONCLUSIONS

The main contributions of this thesis with respect to the state-of-the-art can be summarized as follows:

1. A new generic defect-oriented BIST approach is proposed, called *SymBIST*, virtually applicable to all A/M-S ICs. *SymBIST* relies on constructing or identifying existing invariances and checking whether those invariances are satisfied.
2. *SymBIST* is a reusable BIST approach for defect-oriented test, on-line test and diagnosis. These test-related tasks can be accomplished using the same *SymBIST* test infrastructure. The operation mode is set externally by the user and *SymBIST* generates internally the appropriate test stimulus and makes the necessary re-configurations depending on the operation mode.

3. *SymBIST* is the first BIST demonstrated for post-manufacturing defect-oriented test of ADCs. This is thanks to the short test time of *SymBIST*, which allows performing large-scale defect simulation of the ADC in reasonable time.
4. *SymBIST* is in addition the first BIST demonstrated for on-line test of ADCs. *SymBIST* checks the health status of the ADCs concurrently with its operation and independently of the input. This use is demonstrated for latent defects and single event upsets.

In the following Chapters we will analyze in detail the aforementioned contributions.

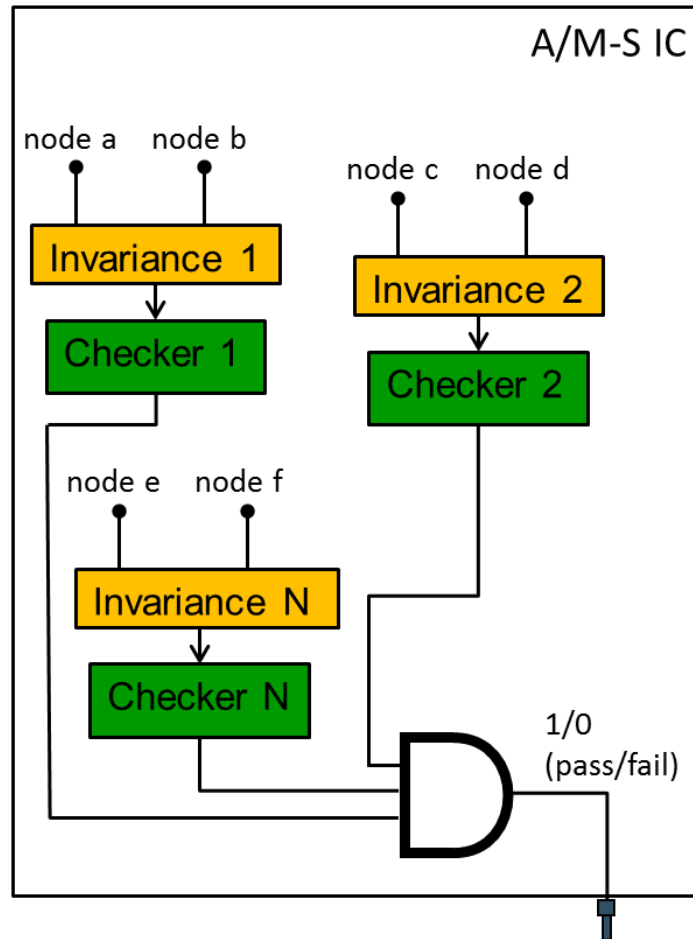
SYMBIST PRINCIPLE

This chapter describes the proposed generic BIST paradigm for A/M-S ICs, called *SymbBIST*, and shows its uses for post-manufacturing defect-oriented test, on-line test, and fault diagnosis. In Section 3.1, we will provide an overview of the *SymbBIST* principle of operation. In Section 3.2, we will discuss invariance generation which is the basis of *SymbBIST*. In Section 3.3, we will describe the strategy for implementing *SymbBIST*. In Section 3.4, we will describe the use of *SymbBIST* in the context of functional safety, including defect coverage and on-line testing. In Section 3.5, we will show how *SymbBIST* can be adapted for fault diagnosis too.

3.1 SYMBIST PRINCIPLE OF OPERATION

As illustrated in Figure 3.1, the underlying idea in *SymbBIST* is to build invariant signals by monitoring internal nodes, where invariance in this context means a signal that by design should be fixed to a default value regardless the input of the circuit. These invariances are monitored by checkers and, if one or more invariances deviate from their default value, then this points to an anomaly in the operation and the corresponding checkers will flag an error. The convention used is that 1/0 checker outputs correspond respectively to pass/fail decision. All checker outputs are connected to an AND gate to provide a single combined 1-bit pass/fail decision.

In practice, the invariant signal is not expected to match exactly its nominal default value due to noise and process, voltage and temperature (PVT) variations. For this reason, the checkers implement a window comparison and verify that the invariant signal lies within a tolerance window in error-free operation. This tolerance window is set to $[\alpha - \delta, \alpha + \delta]$, where α is the invariant signal nominal value and $\delta > 0$. $\alpha - \delta$ and $\alpha + \delta$ are the Lower Test Limit (LTL) and Upper Test Limit (UTL), respectively. In this case, a checker flags an error when the invariant signal slides outside this window. A first estimate of the parameters α and δ can be computed by performing a MC analysis. Specifically α and δ are set to μ and $k \cdot \sigma$, respectively, where μ and σ are the mean and standard deviation, respectively, of the invariant signal across the MC runs. The extracted k from MC analysis can be fine-tuned taking into account worst-case specifications and environmental conditions. Moreover, the comparison window may shift due to thermal noise. For this reason, as is typical in all test programs, *SymbBIST* is repeated several times and a

Figure 3.1: *SymbIST* principle.

voting scheme is used to decide on pass or fail, similar to averaging of a measurement. If one or more trials result in failure, this points to an outlier device which can be discarded for safety reasons.

Note that the invariance is typically violated by large in the presence of a defect, thus the comparison window can be approximate. This means that the checker as well as the internally generated reference voltages *UTL* and *LTL* can be of low-precision.

In general, there is a trade-off between false-positives, e.g., yield loss, and false negatives, e.g., test escapes, and the coefficient k should be set accordingly to meet the desired trade-off. A low k favors test escape reduction at the expense of some yield loss, while a high k guarantees high yield at the expense of some test escapes. For improving safety, a defect should be rejected in post-manufacturing even if it does not produce a fault. During on-line test, however, it is critical to reduce the false positive risk. Thus, for on-line test we can envision a larger tolerance window than in post-manufacturing testing. Overall, *SymbIST* can be tuned to result in high defect coverage, i.e., few test escapes, while warranting negligible yield loss.

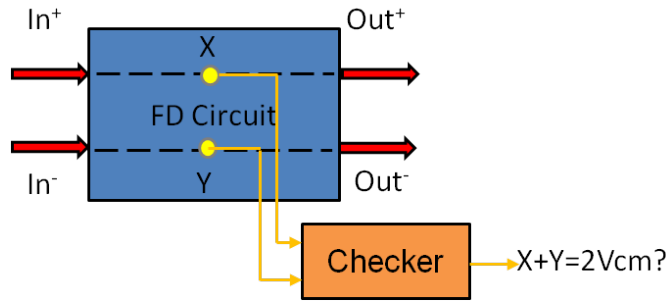


Figure 3.2: Invariance in FD circuit.

3.2 INVARIANCES

Invariances can be built by exploiting symmetries that are inherent to virtually all *A/M-S ICs*. Such symmetries exist thanks to *FD* signal processing [112], complementary signal processing, and replication of identical blocks. Symmetries can also be created artificially with re-configuration using switches, duplication of blocks, or pseudo-duplication of blocks. The goal of pseudo-duplication is to avoid fully replicating an entire block but instead creating a less complex block which produces the same output. In essence, pseudo-duplication constructs two nominally identical signals that are carried via distinct circuit paths. As we will see later, for our case study it was not necessary to perform any re-configuration, duplication, or pseudo-duplication, but we list these techniques as options for applying successfully *SymBIST* to other designs.

For node pairs carrying *FD* or complementary signals we can build an invariance in the form of $V_1 + V_2 = \alpha$, where V_1 and V_2 are the node voltages. For example, in the case of *FD* signals, $\alpha = 2V_{cm}$, where V_{cm} is the common-mode voltage. Figure 3.2 shows an example of a *FD* circuit. The two *FD* nodes X and Y are monitored by a checker that checks whether the sum of these *FD* nodes is equal to $2V_{cm}$. Notice that differential signaling has been conceived precisely to shield the performance of the circuit from many non-idealities affecting common-mode, i.e., poor power supply, temperature variations, noise, etc., thus deviation in common-mode may be innocuous for the *CUT*. This is taken into account by implementing a tolerance window in the checker operation. In contrast, a defect affecting the operation is expected to invalidate the *FD* encoding and bring the invariance outside this tolerance window.

For identical blocks as shown in Figure 3.3, duplicated blocks, or pseudo-duplicated blocks, we can drive them with the same input and build an invariance in the form of $V_1 - V_2 = \alpha$, where in this case V_1 and V_2 are outputs of the two blocks and α has a default value of 0. Figure 3.4 shows a high-level implementation of pseudo-duplication, where both the linear analog and the pseudo-duplicated circuit are driven with

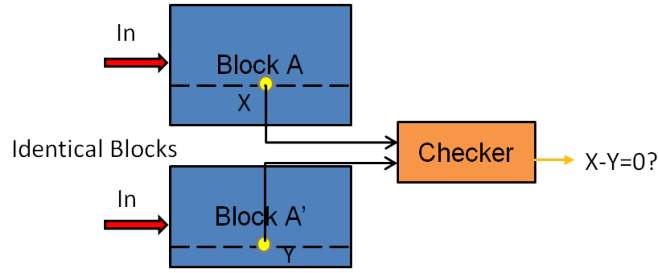


Figure 3.3: Invariance in identical blocks.

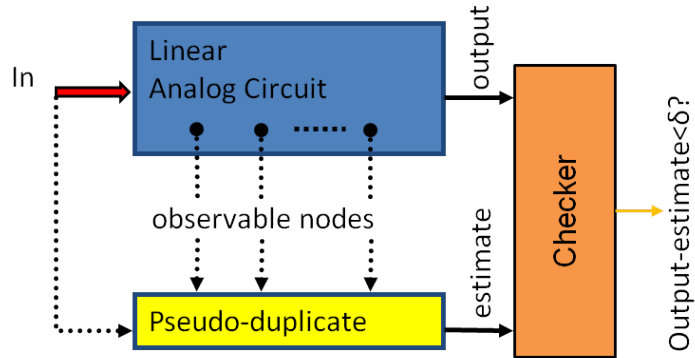


Figure 3.4: Invariance with pseudo-duplication.

the same input. Then the outputs of the two circuits are compared by a checker that checks whether their difference is smaller than δ .

While *SymbIST* is a generic *BIST* paradigm and similar invariances based on symmetry can be derived for any *A/M-S* circuit class, the invariances need to be handcrafted on a circuit by circuit basis and it is likely that distinct invariances can be defined for a given circuit.

3.3 STRATEGY

A high-level abstraction of the proposed *SymbIST* strategy is illustrated in Figure 3.5. The *A/M-S* IC is divided into purely Digital (*D*) blocks on one side and Analog (*A*) and mixed Analog-Digital (*A/D*) blocks on the other side. We assume that the purely digital blocks are tested with standard digital *BIST*, i.e., with scan insertion and a combination of stuck-at, bridging, I_{ddq} , and transitional *ATPG*. The *A* and *A/D* blocks are divided into three groups. The first two groups include blocks that are *FD*, they perform single-to-*FD* conversion, they provide complementary outputs, they appear multiple times, etc. For these blocks invariances exist naturally and the *SymbIST* strategy applies directly. The third group includes the rest of the blocks. For some of the blocks it may be possible to perform re-configuration or pseudo-duplication so as to build invariances and apply the *SymbIST* strategy. We also have the option to perform direct

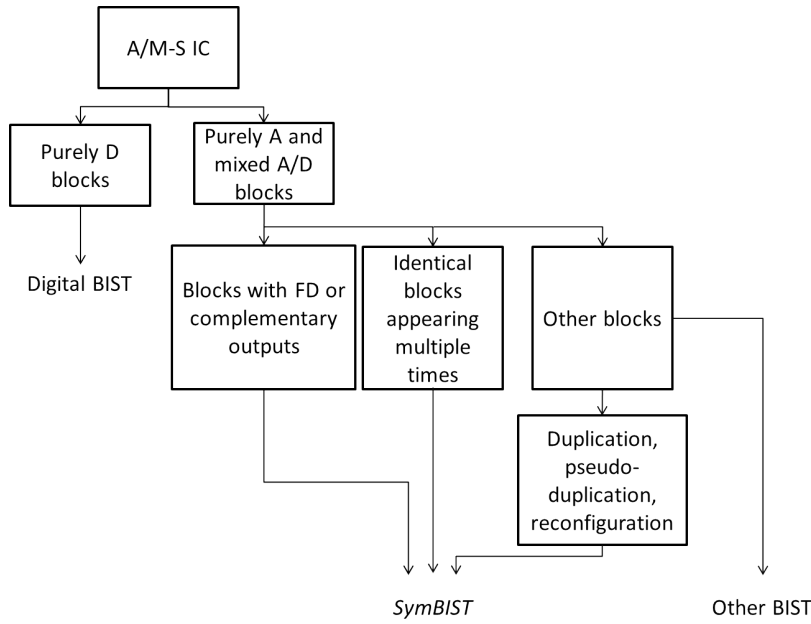


Figure 3.5: *SymBIST* strategy.

duplication of blocks. For the remaining blocks that are not handled with *SymBIST*, we need to develop other *BIST* approaches.

3.4 MODES OF OPERATION

SymBIST is designed to serve as a functional safety-mechanism that is reusable starting from post-manufacturing test, where it is defect-oriented targeting the detection of structural defects with high coverage and screening of outliers, to on-line test in the field of operation, where it targets low-latency detection of transient failures, reliability hazards, and degradation due to aging. While *SymBIST* does not achieve error correction, it can serve as the error-detection and error-diagnosis mechanism in the feedback loop that performs the error-tolerance or self-repair function.

More specifically, the *A/M-S IC* with embedded *SymBIST* has four possible modes of operation enabled by a signal EN:

$$\text{EN} \langle 0 : 1 \rangle = \begin{cases} 00 : \textit{SymBIST} \text{ self-test} \\ 11 : \text{off-line test} \\ 10 : \text{on-line test} \\ 01 : \textit{SymBIST} \text{ disabled} \end{cases} \quad (3.1)$$

The first mode consists in a self-test of the *SymBIST* infrastructure. The *SymBIST* infrastructure occupies considerably smaller area on the die compared to the area of the *A/M-S IC* itself and, thereby, the probability of a defect occurrence within the *SymBIST* infrastructure is considerably

smaller compared to a defect occurrence within the A/M-S IC. Nevertheless, a good strategy is to test the *SymBIST* infrastructure first before using it to test the A/M-S IC. For example, a defective checker may result in a misleading test decision for the A/M-S IC, i.e., it can mask a defect within the A/M-S IC resulting in a test escape or false negative.

The second mode is the off-line test mode that employs *SymBIST* for post-manufacturing defect-oriented test. This mode requires a built-in test stimulus generator and possibly a re-configuration of the A/M-S IC that should be non-intrusive, i.e., incur no performance penalty, when the A/M-S IC runs in normal mode.

The third mode is the on-line test mode that employs *SymBIST* for concurrent error detection during the normal operation of the A/M-S IC. In this case, the running input in normal operation is used and the checkers monitor the invariances on-the-fly flagging errors in real-time, possibly with some low latency since the error needs to propagate to a pair of nodes that are used for building an invariance.

The fourth and last mode allows switching-off *SymBIST* during normal operation so as to save power. In this case, periodic test can be performed either during normal operation by enabling periodically the on-line test mode or in idle times by enabling the off-line test mode.

3.5 DIAGNOSIS USING SYMBIST

SymBIST can be used in different phases of testing. The modes of operation of *SymBIST* described in the previous Section 3.4 include post-manufacturing defect-oriented testing and on-line testing. In addition, *SymBIST* can offer better insights into the circuit like any other BIST and, thereby, can assist fault diagnosis. Herein, we explain how *SymBIST* can be adapted for fault diagnosis too.

Let $C_k(t)$ denote the time-varying checker output for the k -th invariance, $k = 1, \dots, K$, where K is the number of invariances. For a checker clocked at a frequency f_{clk} , $C_k(t)$ becomes a bitstring with period $1/f_{clk}$. We consider the vector $C_k[t_i]$ of $n = T \cdot f_{clk}$ bits, where T is the duration of the test stimulus and t_i are sampling times at half-period, i.e., $t_i = i \cdot (1/2f_{clk})$, $i = 1, \dots, n$. If the invariance is permanently satisfied (violated), then $C_k[t_i]$ will be a vector of zeros (ones). It is likely, however, that a defect will cause the invariance to slide outside the tolerance window at specific clock cycles, in which case $C_k[t_i]$ will be a vector containing both zeros and ones. We use the vector $C_k[t_i]$ as an n -bit digital diagnostic measurement pattern.

This diagnostic measurement pattern is a function of the parameters of the *SymBIST* setup. In general, this set of parameters includes the primary test stimulus to the IC denoted by I_{pri} , internally generated test stimuli denoted by I_{int} , and the tolerance window defined by δ . We can consider

unequal tolerance window limits δ_- and δ_+ , i.e., the tolerance window becomes $[\delta_-, \delta_+]$. A setup may involve also some internal re-configuration denoted by R . Let

$$S^\ell = \{I_{\text{pri}}^\ell, I_{\text{int}}^\ell, R^\ell, \delta_-^\ell, \delta_+^\ell\} \quad (3.2)$$

denote the set of parameters of the ℓ -th *SymbIST* setup, $\ell = 1, \dots, L$, where L is the number of *SymbIST* setups.

In post-manufacturing test, the tolerance window sets a desired trade-off between test escapes and yield loss. In contrast, for diagnosis the placement can target uniquely exposing defects. For example, consider a defect that results in a time-varying invariant signal that is permanently above the upper limit of the tolerance window. By placing a higher upper limit at the average value of the invariant signal, we can obtain a vector $C_k[t_i]$ that toggles between 1 and 0, potentially offering a good diagnostic measurement pattern for this defect.

We follow a fault dictionary approach assuming a list of N_f faults denoted by F_j , $j = 1, \dots, N_f$. The diagnostic measurement pattern for fault F_j obtained by the k -th checker using the ℓ -th *SymbIST* setup is denoted by

$$C_k^{F_j, \ell}[t_i] = f(S^\ell). \quad (3.3)$$

Combining the diagnostic measurement patterns of K checkers monitoring the K invariances and considering L *SymbIST* setups, we obtain a diagnostic measurement pattern for fault F_j in the form of a $K \times L \times n$ bitstring

$$DM(F_j) = [C_1^{F_j, 1}, \dots, C_K^{F_j, 1}, \dots, C_1^{F_j, L}, \dots, C_K^{F_j, L}]. \quad (3.4)$$

The diagnosis objective is that the Hamming Distance (**HD**) of the diagnostic measurement patterns of any two faults differs at least by 1, which can be expressed as

$$HD(DM(F_a), DM(F_b)) > 1, a \neq b. \quad (3.5)$$

An ambiguity group is set of faults for which the **HD** of the diagnostic measurements of any pair of faults is 0.

In the diagnosis phase of faulty device X , the diagnostic measurement pattern DM_X is first obtained and then matched with a row in the fault dictionary matrix

$$\begin{bmatrix} DM(F_1) \\ \vdots \\ DM(F_{N_f}) \end{bmatrix}, \quad (3.6)$$

which has the diagnostic measurement pattern of fault j in row j . The row number points to the diagnosed fault.

SymBIST setups are sequentially added by searching in the space of test stimuli, tolerance window limits, and re-configurations, so as to enrich the digital diagnostic measurement pattern towards resolving the ambiguity groups.

3.6 CONCLUSIONS

A generic *BIST* paradigm for analog and mixed-signal circuits called *SymBIST* has been introduced in this chapter. We proposed *SymBIST* to test the analog and mixed analog-digital blocks of an *A/M-S IC*, whereas we assume that the purely digital blocks are tested with standard digital *BIST*. We showed how invariances can be constructed virtually for any class of *A/M-S ICs* thanks to existing fully-differential and complementary signal processing and block replication, as well as via selected duplication and pseudo-duplication. Furthermore, we described the different modes of operation of *SymBIST*, including post-manufacturing defect-oriented testing, on-line testing, and diagnosis.

CASE STUDY AND BIST INFRASTRUCTURE

This chapter deals with the *SymBIST* application to a real case study, in particular a SAR ADC IP designed by STM in a 65nm technology. In Section 4.1, we will provide an overview of our case study. In Section 4.2, we will show how to construct the invariances to cover the entire mixed-signal part of the SAR ADC IP. In Section 4.3, we will describe the *SymBIST* infrastructure, including the test stimulus generation, the checker design, and the test access and control mechanism. In Section 4.4, we will discuss the efficiency of *SymBIST* in terms of test time and overheads.

4.1 CASE STUDY: SAR ADC

The case study used in this work is a 65nm 10-bit SAR ADC IP by ST Microelectronics. In Section 4.1.1, we provide a brief overview of the operation principle of SAR ADCs. The reader is referred to a textbook for a more complete treatment of SAR ADCs [150]. In Section 4.1.2, we will give a concise top-down description of the architecture of the SAR ADC IP by STM.

4.1.1 SAR ADC Principle

The SAR ADC is used in applications that require low power consumption and medium conversion rate, such as data acquisition. The high-level architecture of a SAR ADC is shown in Figure 4.1. In order to process rapidly changing signals, SAR ADCs have an input Sample-and-Hold (SH) to keep the signal constant during the conversion cycle. The conversion cycle takes $n + 2$ clock periods, where n is the number of bits or resolution and the extra two clock periods are for sampling and capturing the n -bit digital output. In each clock period one bit is determined, starting from the Most Significant Bit (MSB) and continuing in each clock period to the next MSB. In each bit conversion, the input voltage is compared to a comparison level created from a DAC, and the outcome of this comparison determines whether the bit will be set to 1 or 0. The SAR Logic controls the DAC and also sets the bit resulting from the comparison. In the first clock period of a conversion cycle, the comparison level from the DAC is set to the midscale voltage $V_{FS}/2$, where V_{FS} is the full scale voltage, and the comparison with the input voltage determines the MSB $B[n]$. In the second clock period, the comparison level is set to $3V_{FS}/4$ if $B[n] = 1$ or to $V_{FS}/4$ if $B[n] = 0$ and the comparison

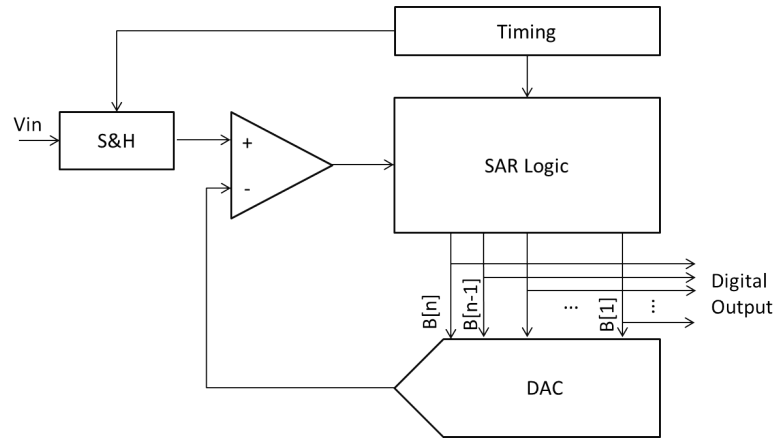


Figure 4.1: High-level architecture of a SAR ADC.

with the input voltage determines the **MSB-1** $B[n-1]$. In the third clock period, the comparison level is set to $7V_{FS}/8$ if $B[n]B[n-1] = 11$, to $5V_{FS}/8$ if $B[n]B[n-1] = 10$, to $3V_{FS}/8$ if $B[n]B[n-1] = 01$ or to $V_{FS}/8$ if $B[n]B[n-1] = 00$, and the comparison determines the **MSB-2** $B[n-2]$. For n bits, the **DAC** implements $2^n - 1$ comparison levels. The conversion continues until all n bits are determined, in which case a new input sample is held and a new conversion cycle begins.

4.1.2 SAR ADC IP by STM

The top-level architecture of the 10-bit **SAR ADC IP** by **STM** is illustrated in Figure 4.2. The circuit accepts a **FD** analog input $\Delta IN = (IN+) - (IN-)$ with a peak-to-peak voltage $2 \cdot V_{REFP} = 1V$, where V_{REFP} is nominally $0.5V$. The common mode of the two inputs $IN+$ and $IN-$ is $V_{DD}/2 = 0.6V$, where $V_{DD} = 1.2V$ is the power supply. CLK is the master clock of the **ADC** with frequency $f_{clk} = 156 \text{ MHz}$ and CLK_{12} is the conversion clock of the **ADC** with frequency $f_{clk_{12}} = f_{clk}/12$. The 10-bit digital output is denoted by $D \langle 9 : 0 \rangle$. The top-level blocks are as follows:

SARCELL: It is the main block of the **SAR ADC** which implements the architecture in Figure 4.1.

SAR Control: It creates 12 pulses $P \langle 0 : 11 \rangle$ used to control the sampling, conversion, and digital output capture phases in the **SARCELL**.

Bandgap: It creates the required biasing for all the blocks of the **SAR ADC**.

Reference Buffer: It creates the comparison levels $V_{REF} \langle 0 : 32 \rangle$ that are used by the **DAC** during the conversion. As we will see below, the **DAC** is a combination of two sub-**DACs** with a 5-bit digital input each. Thus, each sub-**DAC** can set $2^5 - 1 = 31$ comparison levels. $V_{REF}[1]$ to $V_{REF}[31]$ denote these comparison levels, $V_{REF}[0] = \text{GND}$, and $V_{REF}[32] = V_{REFP}$. The midscale voltage is $V_{REF}[16]$. These voltages are generated from a re-

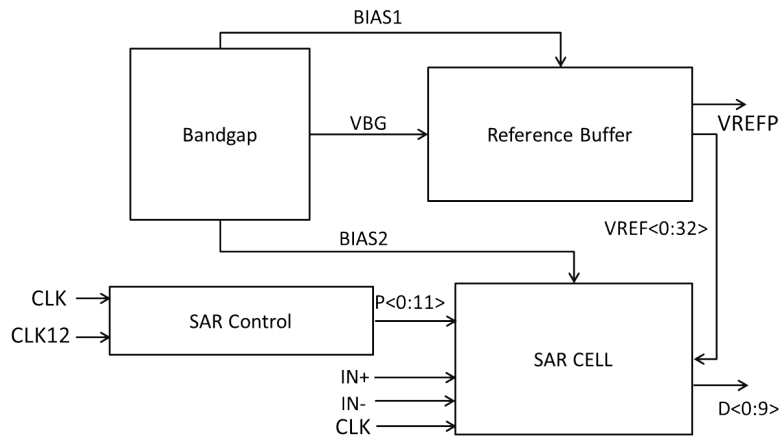


Figure 4.2: Top-level architecture of the SAR ADC IP.

sistive ladder with 32 equal resistors forming a voltage divider. Therefore, $VREF[k] = \frac{k}{16} VREF[16]$, $k = 1, \dots, 32$.

The SARCELL block comprises the following blocks as illustrated in Figure 4.3:

Phase Generator: It controls the timing of the ADC operation by generating the phases for sampling, comparison, conversion, etc.

V_{cm} Generator: It generates the common mode voltage V_{cm} used inside the DAC.

SAR Logic: It controls the conversion process by providing the digital input to the DAC, it stores the result of each comparison, and provides the digital output once the conversion is completed.

10-bit DAC: The DAC is a resistive plus charge redistribution DAC. As shown from its architecture in Figure 4.4, it is composed of two structurally identical sub-DACs, namely SUBDAC₁ and SUBDAC₂, with a 5-bit digital input each, and a SC array. The sampling operation is performed within the DAC. SUBDAC₁ converts the 5 MSBs to comparison levels M+ and M-, while SUBDAC₂ converts the 5 Least Significant Bits (LSBs) to

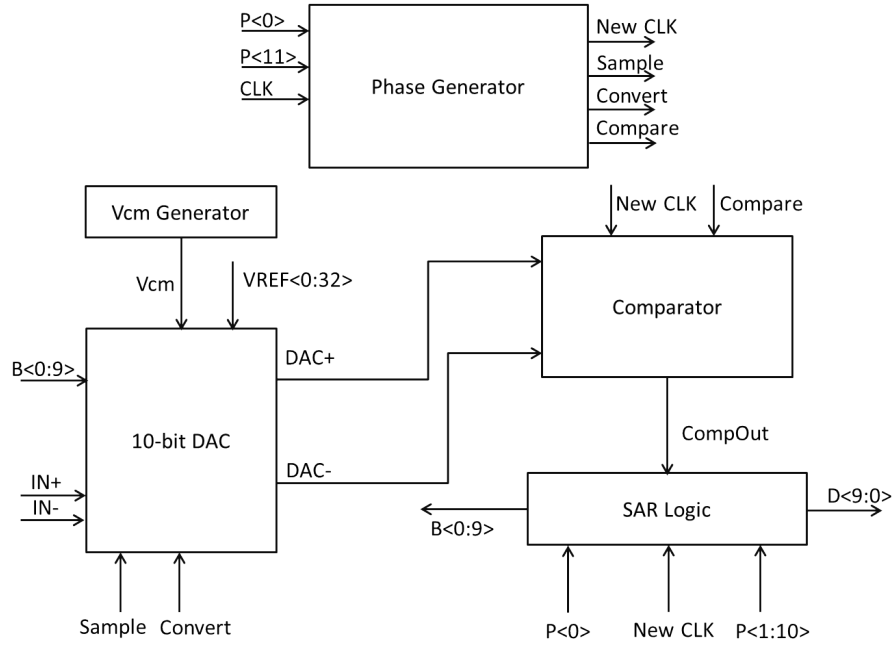


Figure 4.3: SARCELL block architecture.

comparison levels $L+$ and $L-$. The Boolean functions implemented by SUBDAC1 and SUBDAC2 are given by:

$$\begin{aligned}
 M+ &= VREF \left[\sum_{j=5}^9 B[j] \cdot 2^{j-5} \right] \\
 M- &= VREF \left[32 - \sum_{j=5}^9 B[j] \cdot 2^{j-5} \right] \\
 L+ &= VREF \left[\sum_{j=0}^4 B[j] \cdot 2^j \right] \\
 L- &= VREF \left[32 - \sum_{j=0}^4 B[j] \cdot 2^j \right]
 \end{aligned} \tag{4.1}$$

Let $DAC+(i)$ and $DAC-(i)$ denote the DAC output voltages $DAC+$ and $DAC-$ at the i -th conversion cycle, respectively. It can be shown that their difference, denoted by $\Delta DAC(i)$, is given by:

$$\begin{aligned}
 \Delta DAC(i) &= \frac{1}{\sum_{k=1}^3 C_k} \left[C_1 \cdot \Delta M(i) + C_2 \Delta L(i) \right. \\
 &\quad \left. - C_1 \Delta IN + C_2 \cdot (VREF[32] - VREF[0]) \right]
 \end{aligned} \tag{4.2}$$

where $\Delta M(i) = M+(i) - M-(i)$ is the difference between $M+$ and $M-$ at the i -th conversion cycle, $\Delta L(i) = L+(i) - L-(i)$ is the difference between

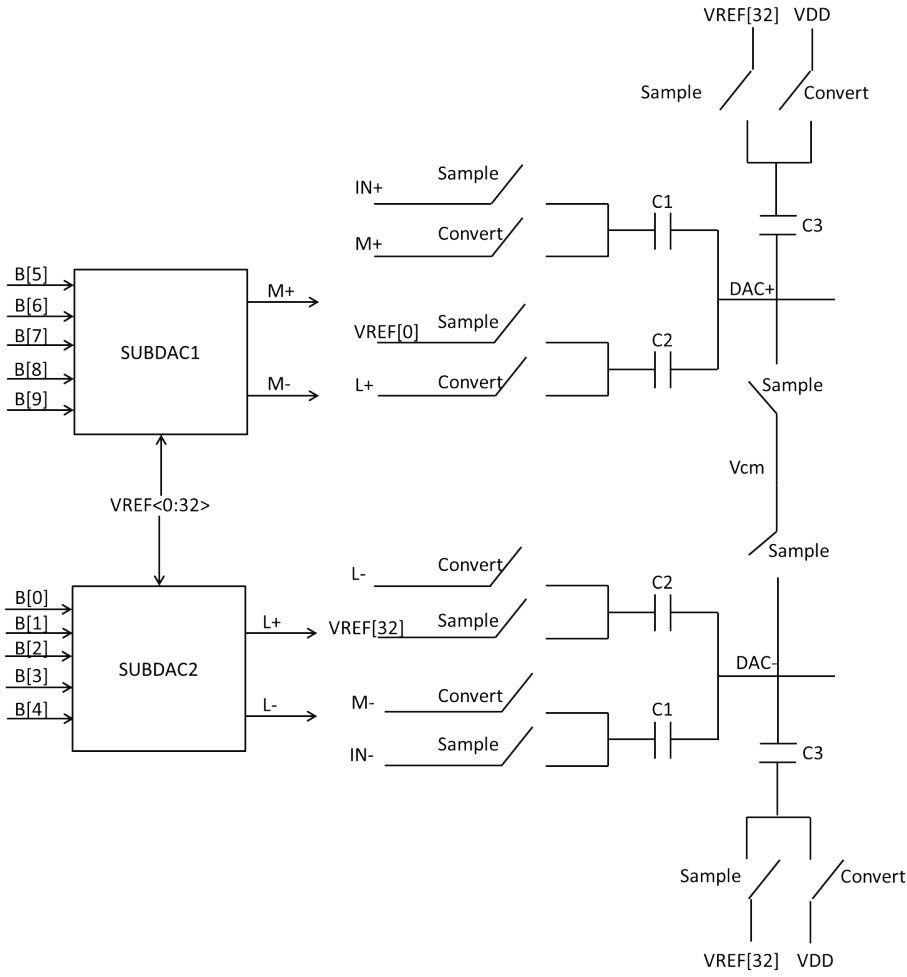


Figure 4.4: 10-bit DAC block architecture.

$L+$ and $L-$ at the i -th conversion cycle, and $\Delta IN = (IN+) - (IN-)$. The capacitors are $C_1 = 32C_U$, $C_2 = C_U$, and $C_3 = 16C_U$, where C_U is the unit capacitor. C_1 and C_3 are implemented with capacitor banks.

Comparator: It compares the two outputs of the DAC and the outcome of the comparison is driven to the SAR Logic block in order to set the corresponding digital bit. The block-level architecture of the comparator is shown Figure 4.5. It comprises a pre-amplifier, a comparator latch, an RS latch, and an offset compensation circuit for the pre-amplifier.

A conversion cycle starts with setting the sample signal high, which samples the FD input signal by charging the capacitors C_1 inside the DAC. Next, the sample signal becomes low and the convert signal becomes high. The SAR Logic assigns $B[9] = 1$ while all the other bits $B[j]$, $j < 9$, are kept at 0. In this case, $M+(1) = M-(1) = VREF[16]$, $L+(1) = VREF[0]$, and $L-(1) = VREF[32]$, which gives $\Delta DAC(1) = -\frac{C_1}{\sum_{k=1}^3 C_k} \Delta IN$ from Equation (4.2). Therefore, the comparator checks the sign of the FD input and if it is positive the SAR Logic block sets $B[9] = 1$, otherwise if it is negative the SAR Logic block sets $B[9] = 0$.

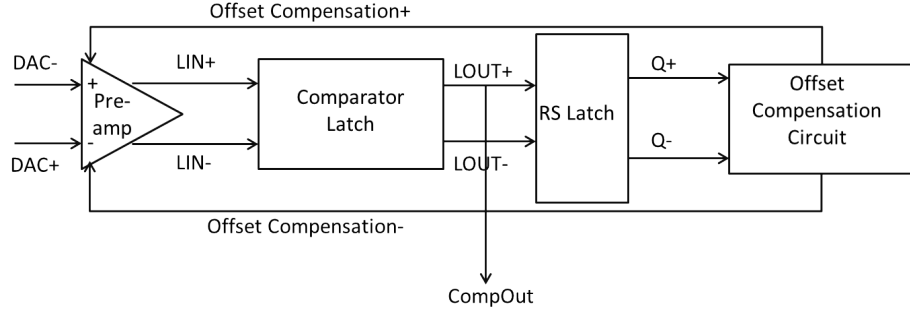


Figure 4.5: Comparator block architecture.

The bit $B[9]$ is kept fixed for the rest of the conversion. In the next conversion cycle, the SAR Logic block assigns $B[8] = 1$, while all the other bits $B[j]$, $j < 8$, are kept at 0. If $B[9] = 1$, then $M+(2) = VREF[24]$, $M-(2) = VREF[8]$, $L+(2) = VREF[0]$, and $L-(2) = VREF[32]$, which gives $\Delta DAC(2) = \frac{C_1}{\sum_{k=1}^3 C_k} [(VREF[24] - VREF[8]) - \Delta IN]$ from Equation (4.2). Since $VREF[24] - VREF[8] = VREF[16]$, the comparator compares $VREF[16] - \Delta IN$ to 0. If $B[9] = 0$, then it can be shown from Equation (4.2) that $\Delta DAC(2) = \frac{C_1}{\sum_{i=k}^3 C_k} [(VREF[8] - VREF[24]) - \Delta IN]$, that is, the comparator compares $-VREF[16] - \Delta IN$ to 0. In other words, in this second conversion cycle the comparator compares $|\Delta IN|$ to $VREF[16]$. In the third conversion cycle, it can be shown from Equation (4.2) that the comparator compares $|\Delta IN|$ to $VREF[24] = 3VREF[16]/2$ if $B[8] = 1$ or to $VREF[8] = VREF[16]/2$ if $B[8] = 0$, and so forth. When the 10 conversion cycles are completed, the digital output $D<0:9> = B<0:9>$ is driven at the output pins.

4.2 INVARIANCES IN SAR ADC IP

In this section, we show how to construct invariances inside the SAR ADC IP shown in Figure 4.2 with the aim to implement successfully the *SymBIST* paradigm. More specifically, looking into the architecture of the SAR ADC IP, we observe that the two sub-DACs within the DAC are structurally identical, each sub-DAC has complimentary outputs, the SC array has symmetrical paths, the pre-amplifier is FD, and the comparator and RS latches have complimentary outputs. Based on these observations, we can build the following invariances that hold true for any FD input ΔIN and at every conversion cycle.

4.2.1 *SymBIST*₁

By construction, as shown from Equation (4.1), the outputs of the two sub-DACs regardless of their inputs take complimentary values $VREF[j]$

and $VREF[32 - j]$, $j = 0, \dots, 32$. Therefore, the following two invariances should always hold true:

$$M+(i) + M-(i) = VREF[32] \quad (4.3)$$

$$L+(i) + L-(i) = VREF[32] \quad (4.4)$$

These invariances can flag failures within the circuitry of the two sub-DACs. In addition, since all comparison levels $VREF[j]$, $j = 0, \dots, 32$, are used by the DAC, as shown from Equation (4.1), these invariances can also flag failures within the bandgap and reference buffer. Note that in the IP a pin is used to measure only comparison level $VREF[32]$. Instead, with invariances in Equation (4.3)-(4.4) we implement a more thorough test by checking all comparison levels. As an auxiliary benefit, this pin may be viewed as obsolete.

4.2.2 *SymBIST2*

It can be shown that the sum of the DAC output voltages at the i -th conversion is given by:

$$\begin{aligned} DAC+(i) + DAC-(i) = & \\ & 2V_{cm} + \frac{1}{\sum_{k=1}^3 C_k} \left[C_1 \cdot (M+(i) + M-(i)) - C_1 \cdot (IN+ + IN-) \right. \\ & + C_2 \cdot (L+(i) + L-(i)) - C_2 \cdot (VREF[32] - VREF[0]) \\ & \left. + 2C_3 \cdot (VDD - VREF[32]) \right] \quad (4.5) \end{aligned}$$

Using Equations (4.3)-(4.4), substituting $C_3 = C_1/2$, and considering that $IN+ + IN- = VDD$, Equation (4.5) gets simplified as follows:

$$DAC+(i) + DAC-(i) = 2V_{cm} \quad (4.6)$$

This invariance can flag failures within the complete DAC, including the sub-DACs and the SC array, within the circuits that provide the voltage references to the DAC, i.e., V_{cm} generator and reference buffer, as well as within the bandgap that provides the biasing to the reference buffer.

4.2.3 *SymBIST*₃

Thanks to the fully-differential structure of the pre-amplifier within the comparator, the following invariance should be satisfied at the outputs of the pre-amplifier regardless of the difference $\Delta\text{DAC}(i)$ being amplified:

$$\text{LIN}+(i) + \text{LIN}-(i) = 2V_{\text{cm}2} \quad (4.7)$$

where $\text{LIN}+(i)$ and $\text{LIN}-(i)$ are the **FD** outputs of the pre-amplifier at the end of the i -th conversion and $V_{\text{cm}2}$ is the common mode at the outputs of the pre-amplifier. This invariance can flag failures within the pre-amplifier and the offset compensation circuit, as well as within the bandgap that provides the biasing to these circuits.

4.2.4 *SymBIST*₄

More tests can be constructed for the comparator latch and RS latch within the comparator block by checking the invariances:

$$Q+(i) + Q-(i) = VDD \quad (4.8)$$

$$\text{sgn}(Q+(i) - Q-(i)) - \text{sgn}(\text{LIN}+(i) - \text{LIN}-(i)) = 0 \quad (4.9)$$

where $\text{sgn}(\cdot)$ denotes the sign function and $Q+(i)$ and $Q-(i)$ are the complementary outputs of the RS latch at the end of the i -th conversion. This invariance can flag failures within the comparator latch, the RS latch and the offset compensation circuit that end up as a stuck-at fault at one of the outputs or at an inversion of the outputs.

Table 4.1 summarizes the **BIST** approaches corresponding to the different blocks of the **SAR ADC IP**. **BIST** approaches are divided into *SymBIST* for the **A/M-S** blocks and digital **BIST** for the purely digital blocks, namely the **SAR** control, phase generator, and **SAR** Logic. As it can be seen, the 6 *SymBIST* invariances in Equations (4.3)-(4.4) and (4.6)-(4.9) are capable of covering the complete **A/M-S** part of the **SAR ADC IP**.

4.3 SYMBIST INFRASTRUCTURE

This section describes the *SymBIST* infrastructure embedded into the **SAR ADC IP**. As we will see in Chapter 6, this same infrastructure will be used for post-manufacturing defect-oriented off-line testing, on-line testing, and fault diagnosis

Blocks	Sym-BIST1	Sym-BIST2	Sym-BIST3	Sym-BIST4	Digital BIST
SAR Control					X
Phase Generator					X
SAR Logic					X
Bandgap	X	X	X		
Reference Buffer	X	X			
SUBDAC ₁	X	X			
SUBDAC ₂	X	X			
SC Array	X	X			
V _{cm} Generator		X			
Pre-amplifier			X		
Comparator Latch				X	
RS Latch				X	
Offset Compensation Circuit			X	X	

Table 4.1: Matrix showing correspondence between BIST approaches and SAR ADC IP blocks.

4.3.1 Test Stimulus and Re-configuration for Applying SymBIST

The test stimulus has two parts, namely a static and a dynamic, and both parts can be robustly generated on-chip. The static part is simply a DC input to the ADC, denoted by $\Delta I_{N_{DC}}$. The dynamic part is a set of digital test patterns that are applied sequentially to the inputs of the two SUBDACs. More specifically, we cycle through all 2^5 bit combinations at the inputs of each SUBDAC. The rationale of this dynamic part of the test stimulus is that it activates all components within the DAC and also extensively exercises the comparator since various differences $\Delta DAC(i)$ are generated at its input. The components within the bandgap and reference buffer are also activated since during this test all comparison levels $VREF[j]$, $j = 0, \dots, 32$, are used within the DAC, as shown from Equation (4.1). The V_{cm} Generator is checked directly with the invariance in Equation (4.6).

The SUBDAC input sequence generator is shown in Figure 4.6. A 5-bit digital counter is used to generate the incremental counting $W \langle 0 : 4 \rangle$ and a shuffling block is used to shuffle its outputs so as to generate the digital test stimuli $Q \langle 0 : 4 \rangle$ and $Q \langle 5 : 9 \rangle$ that cycle non-incrementally through all 2^5 bit combinations at the inputs of each SUBDAC. The shuffling block can be programmed to implement different cycles. Then, 10 2:1 multiplexers are used to switch during test mode the inputs of the SUBDACs, denoted now by $B_{new} \langle 0 : 4 \rangle$ and $B_{new} \langle 5 : 9 \rangle$,

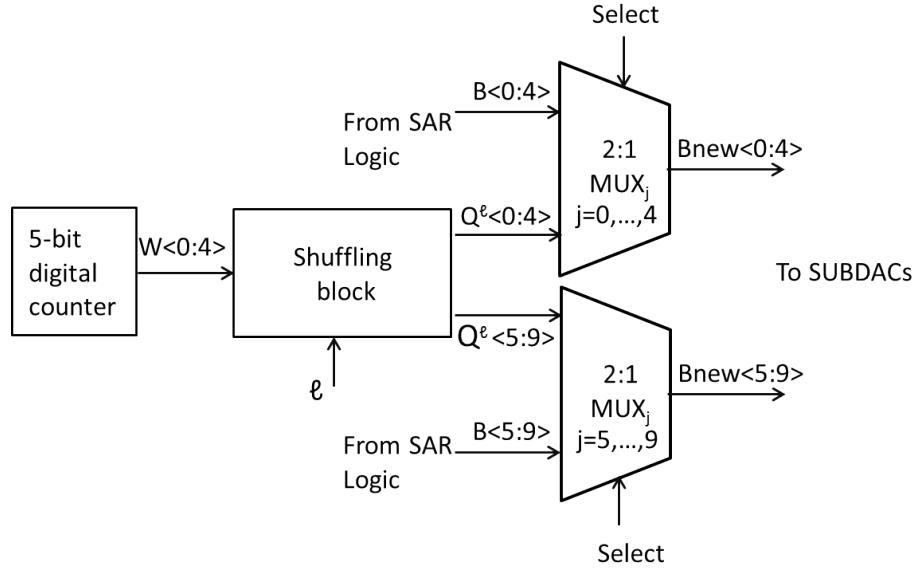


Figure 4.6: On-chip generation of dynamic test stimulus.

from the SAR Logic outputs $B\langle 0 : 4 \rangle$ and $B\langle 5 : 9 \rangle$ to $Q\langle 0 : 4 \rangle$ and $Q\langle 5 : 9 \rangle$, respectively. The shuffling block is controlled by a digital signal l , whereas the multiplexers are controlled by a Select signal. These digital signals are set according to the desired mode of operation of *SymBIST*, as explained in Sections 3.4 and 3.5, and can be accessed and controlled externally using the standard test access and control mechanism described in 4.3.4.

Figure 4.6 shows a generic block-level architecture of the test stimulus generation and required re-configurations in order to apply *SymBIST* for off-line testing, on-line testing and fault diagnosis. Depending on the specific mode of operation of *SymBIST* that runs, different combinations of signals l and Select apply and the architecture of the on-chip generation of the dynamic test stimulus is simplified. In Chapter 6, before presenting the results, the specific test stimulus and re-configurations for each operation mode are presented.

4.3.2 Checker Design

The invariances described in Section 4.2 are of two types, i.e., one type $V_1 + V_2 = \alpha$ which concerns invariances in Equations (4.3)-(4.4) and (4.6)-(4.8), and one type $V_1 - V_2 = 0$ which concerns Equation (4.9). A dedicated checker is designed for each type.

Figure 4.7 shows the checker design for the invariances of type $V_1 + V_2 = \alpha$. A straightforward design would be based on a summing amplifier. Herein, we propose a simplified design to reduce the area of the checker. In particular, let $V_j = V_j^{DC} + v_j$, where V_j^{DC} denotes the large-signal DC

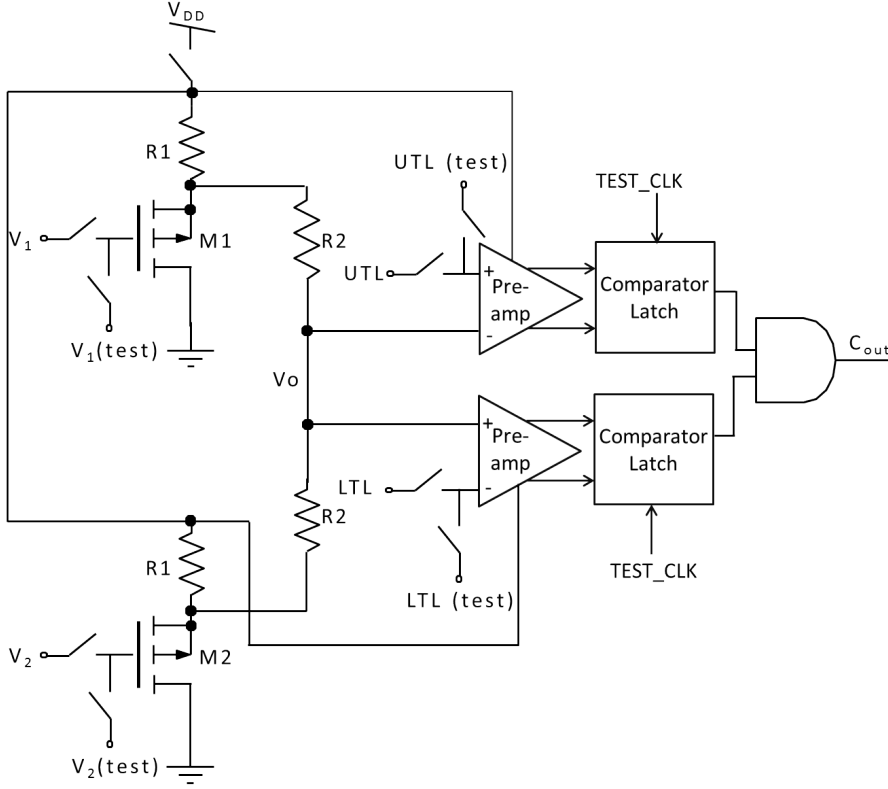


Figure 4.7: Checker design for the invariances in Equations (4.3)-(4.4) and (4.6)-(4.8). The switches disconnect the checker when *SymbIST* is disabled or set the checker into self-test mode.

quantity and v_j denotes the small-signal Alternating Current (AC) quantity of V_j , $j = 1, 2$. The proposed checker is composed of four stages. The first stage includes two buffers implemented with two source follower amplifiers using identical P-Channel Metal-Oxide Semiconductor (PMOS) transistors M_1 and M_2 . The buffers are used so as to avoid loading the nodes that are being monitored. Let g_m denote the transconductance of transistors M_1 and M_2 , and let I_1^{DC} and I_2^{DC} denote the DC biasing currents of transistors M_1 and M_2 , respectively. The second stage is a voltage divider, where $R_2 \gg R_1$, that generates the signal V_o :

$$V_o = \left(V_{DD} - \frac{R_1}{2} (I_1^{DC} + I_2^{DC}) \right) + G \cdot (v_1 + v_2), \quad (4.10)$$

where

$$G = \frac{g_m R_1}{2(g_m R_1 + 1)}. \quad (4.11)$$

The expression of V_o can be re-written as:

$$V_o = f(V_1^{DC}, V_2^{DC}) + G \cdot (V_1 + V_2), \quad (4.12)$$

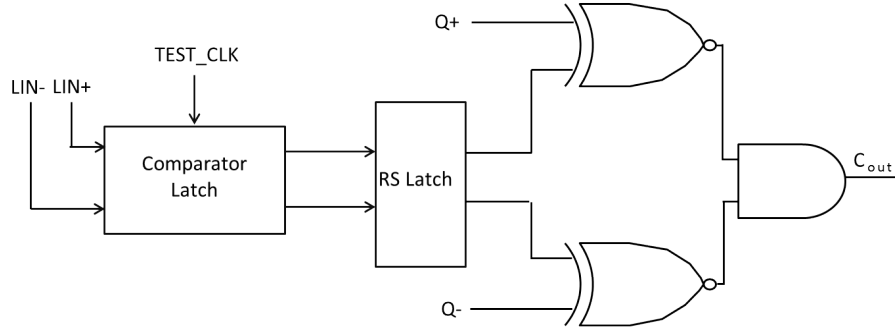


Figure 4.8: Checker design for the invariance in Equation (4.9).

where

$$f(V_1^{\text{DC}}, V_2^{\text{DC}}) = \left(V_{\text{DD}} - \frac{R_1}{2} (I_1^{\text{DC}} + I_2^{\text{DC}}) \right) - G \cdot (V_1^{\text{DC}} + V_2^{\text{DC}}) \quad (4.13)$$

is the DC component of V_o depending on the DC biasing of the monitored nodes and the gain quantity G multiplies the invariant signal. Ideally, in error-free operation, V_o is a DC signal with value $V_o = f(V_1^{\text{DC}}, V_2^{\text{DC}}) + G \cdot \alpha$. A defect will shift the DC component of V_o and/or will add an AC component to it. The third stage includes two comparators that compare V_o to the test limits $\text{UTL} = \mu + k\sigma$ and $\text{LTL} = \mu - k\sigma$, where μ and σ are the mean and standard deviation of V_o computed over several MC runs and over the 2^5 values observed during the duration of the test stimulus for each MC run. The UTL and LTL are generated internally using resistor voltage dividers. Note that the comparison window defined by the lower and upper test limits eliminates false positives due to noise and PVT variations within both the SAR ADC and the first two stages of the checker. The fourth stage passes the outputs of the two comparators through an AND gate to obtain a checker output C_{out} in the form of a 1-bit pass/fail response. When the invariance is satisfied, i.e., V_o lies within the range defined by the test limits, the output of the checker is high, i.e., $C_{\text{out}} = 1$, whereas when the invariance is violated the output of the checker is low, i.e., $C_{\text{out}} = 0$. Figure 4.7 also shows the insertion of switches to disconnect the checker when *SymBIST* is disabled or to set the checker into self-test mode.

Figure 4.8 shows the checker design for the invariance in Equation (4.9). It monitors the LIN+ and LIN- outputs of the pre-amplifier and the outputs Q+ and Q- of the RS latch within the comparator in the SARCELL block. It is easy to verify that $C_{\text{out}} = 1$ when the invariance in Equation (4.9) holds true and $C_{\text{out}} = 0$ when it is violated. Note that this checker does not implement a tolerance window since the invariance is constructed from digital signals.

Given that the checker design for the invariances in Equations (4.3)-(4.4) and (4.6)-(4.8) is identical, we have two options. The first option is to use a single checker and use it to check these invariances sequentially by using corresponding test limits. The second option is to use one checker per invariance, thus checking the invariances in parallel. In off-line test mode, the first option offers a trade-off between area overhead and test time. However, in on-line test mode, the first option has the disadvantage that a transient error may be detectable by a unique invariance that is momentarily not being checked when the transient error occurs. In our implementation, we adopted the second option.

4.3.3 Checker Self-Test

As discussed in Section 3.4, it is advised to test the *SymBIST* infrastructure prior to its usage for testing the ADC itself. This is desired especially for the checker in Figure 4.7 whose first three stages are analog and, thereby, less robust. To this end, we propose to implement a simple sequence of DC tests to exercise this checker, in order to first decide on its health status prior to its usage. More specifically, referring to Figure 4.7, we can assume different DC test stimuli for the four checker inputs, i.e., V_1 , V_2 , UTL, and LTL. A set of possible convenient DC values are $V_1 = \{\text{GND}, \text{VREF}[32], \text{VDD}\}$, $V_2 = \{\text{GND}, \text{VREF}[32], \text{VDD}\}$, and $k = \{3, 5\}$, where VREF[32] can be drawn directly from the reference buffer. A test uses a combination of such DC values. For each test, the checker is expected to give a high or low output. A flipped output points to a faulty checker. The goal is to create a minimum sequence of N tests that achieves sufficiently high defect coverage for the checker. Figure 4.7 includes the addition of switches for the self-test mode of the checker. All checkers can be tested in parallel, thus the checker self-test time is $N \cdot (1/f_{\text{clk}}) = N \cdot 6.41\text{ns}$.

4.3.4 Test Access and Control Mechanism

SoCs nowadays integrate numerous IP blocks and DfT structures. Since the number of these DfT structures might surpass a few hundreds, it is restrictive to access and control them from primary pins. For this reason it is essential to use an on-chip test infrastructure to connect all the DfT structures to a common test bus in order to be able to perform all the necessary actions required for testing the ICs, using a limited number of primary pins [151].

SymBIST can be interfaced to a digital test access and control mechanism based on two external pins which is the minimum. We can employ the test access and control mechanism proposed in [152], which is compatible with the latest IEEE 1687 Standard (Std.) for test infrastructure controlla-

bility and observability [153]. IEEE Std. 1687 deals with the great number of DFT structures and connects them serially via programmable Segment Insertion Bits (SIBs) to a Re-configurable Scan Network (RSN) between the Scan In (SI) and Scan Out (SO) ports. When the SIB of a DFT structure is opened, its Test Data Registers (TDRs) become part of the RSN such that it is accessed from the SI port and its output is streamed to the SO port.

The principle for connecting an analog IP to the common test infrastructure is depicted in Figure 4.9 [152]. This example contains one analog IP with its embedded DFT structures and several digital IPs, all connected to the same common scan path. On-chip ADCs and DACs are used to digitize the analog test responses and generate analog test stimuli from digital words. Note that in large SoCs these ADCs and DACs can be shared among multiple IPs, if they are not tested simultaneously. Three types of connections to the scan path are shown in Figure 4.9 for the analog IPs: (a) a DAC creates an analog input test stimulus from digital words and forces the test stimulus inside the analog IP or the DFT ; (b) an ADC is used to read out in digital format the analog output test signals from inside the IP or the DFT ; (c) a direct connection to the DFT providing a digital input control.

Figure 4.9 shows for simplicity 3-bit data converters and 3-bit words controlling the DFT structure, but in fact any TDR size can be used in the scan path. It also shows a number of TDRs that connect digital IPs to the scan path, a case where an analog signal from the analog IP is digitized from the ADC and then transferred inside a digital IP, and the case where a digital signal from a digital IP is converted from the DAC and then used inside the analog IP. Finally, Figure 4.9 shows the three main control signals, namely shift, capture and update. With the shift operation the data are shifted serially one bit per clock cycle. The capture operation loads the digitized output of the ADC to the scan path to be scanned out or used by some other IP inside the SoC. The update operation latches the data into the input of the DAC for the test signal to be forced. For each ADC and DAC, a counter and a packet size register are used that set the periodicity of the TDR update and capture operations. More specifically, before the serial access takes place, the counter connected to the ADC data registers is initialized with an offset equal to the number of cycles that must occur before the capture of the ADC starts, whereas the counter connected to the DAC data registers is initialized with an offset equal to the number of cycles that must occur before the serial data reaches the DAC register. The packet size registers are initialized with the same packet size, which is responsible for the frequency of the ADC capture and the DAC update.

When applying *SymBIST* to the SAR ADC we have the following cases with regard to the three types of connections mentioned above. Connection (a) is used for forcing the static analog test stimulus to the FD input

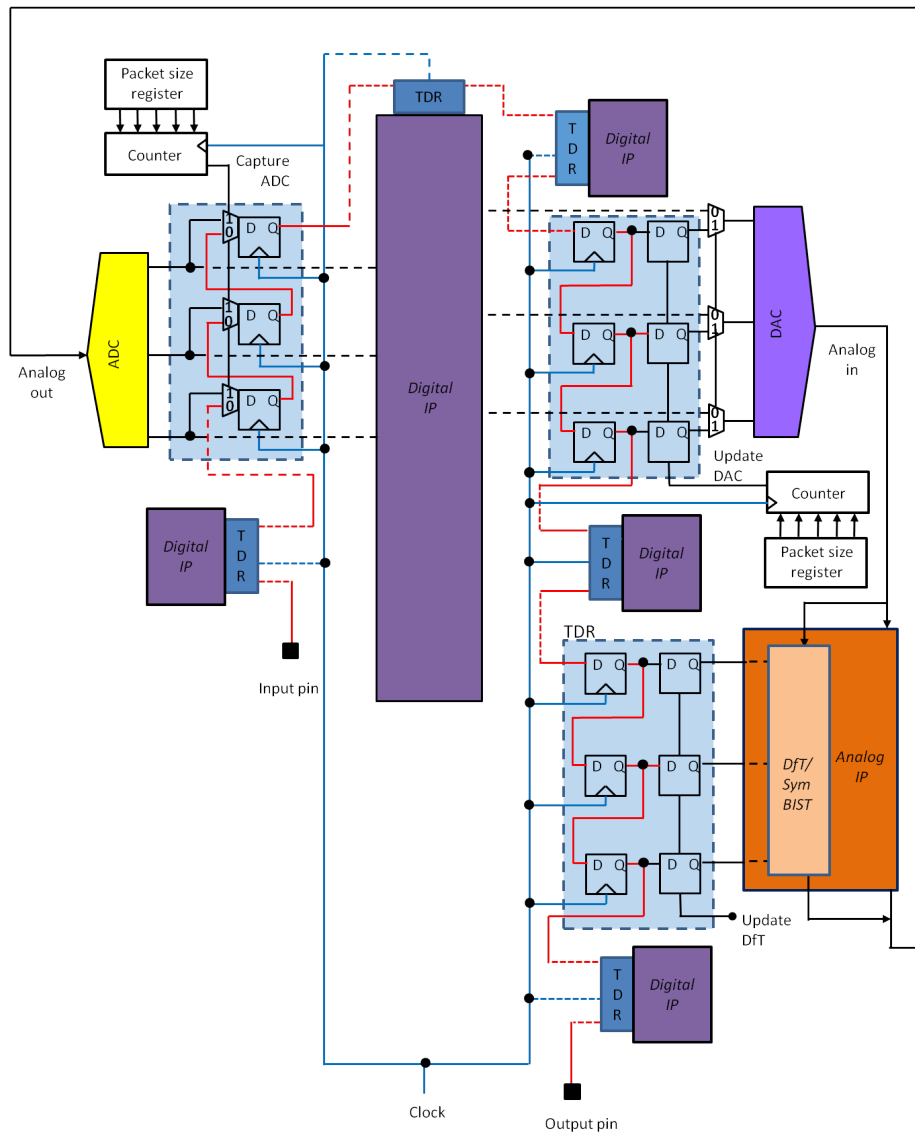


Figure 4.9: Test access and control mechanism of *SymBIST* (adapted from [152]).

of the *SAR ADC*, as explained in Section 4.3.1. Connection (b) is used for reading out the *SymBIST* 1-bit digital response indicating pass/fail. Since the test output is digital, the *ADC* is not needed and the *SymBIST* output can be directly connected to the scan path. More a 1-bit register is needed instead of a 3-bit register shown in Figure 4.9. Connection (c) is used to set the EN signal in order to choose the test mode, as described in Section 3.4, and to program the shuffling block that generates the dynamic part of the digital test stimulus.

For a more detailed description of the test infrastructure, the interested reader is referred to [154] and [152].

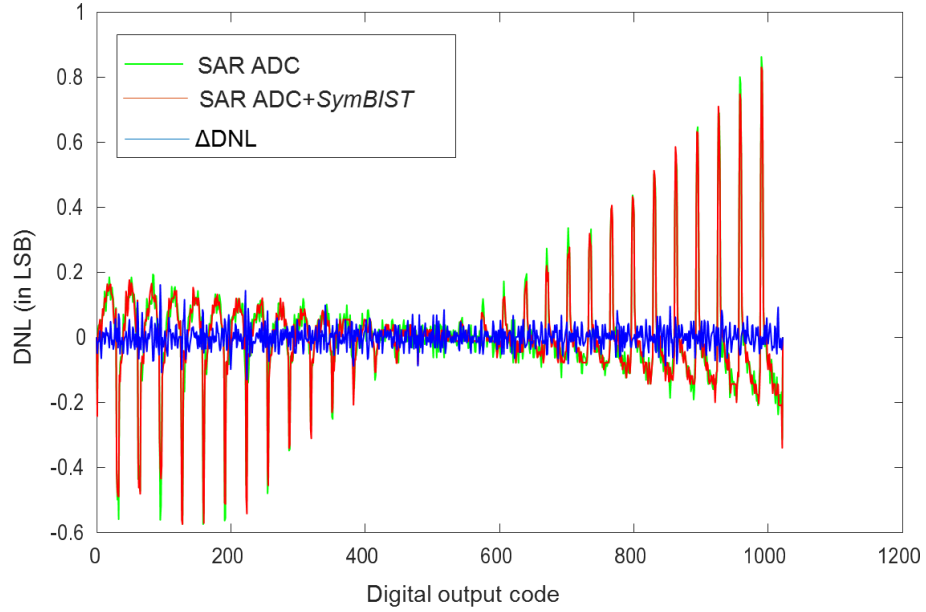


Figure 4.10: DNL of original design and original design with embedded *SymBIST*.

4.4 SYMBIST EFFICIENCY

4.4.1 *SymBIST* Test Time

In total, 6 checkers are used, and the same test stimulus is used to check all invariances in parallel. During on-line testing the checkers uninterruptedly check if the invariances are being violated, thus *SymBIST* runs concurrently with the operation and is transparent to it. During off-line testing, as explained in 4.3.1, we need to cycle through all 2^5 bit combinations at the inputs of each SUBDAC. Therefore, the duration of the test stimulus is 2^5 clock cycles, which corresponds to a test time $T = 2^5 \cdot (1/f_{\text{clk}}) = 0.205\mu\text{s}$ per *SymBIST* setup.

4.4.2 *Overheads*

The *SymBIST* infrastructure comprises the 5-bit digital counter used in the off-line test mode, multiplexers to re-configure the design in the off-line test mode, switches to enable the different modes of operation, and 6 checkers, i.e., one checker per invariance. The multiplexers are inserted in a digital signal path between the SAR Logic and 10-bit DAC, the checker in Figure 4.7 taps into nodes via switches and includes buffers as a first stage, and the checker in Figure 4.8 is digital. Therefore, the modifications are non-intrusive to the design and no design re-iterations are required. To confirm that *SymBIST* does not incur any performance penalty, we simulated at transistor-level the DNL and INL for the original design and

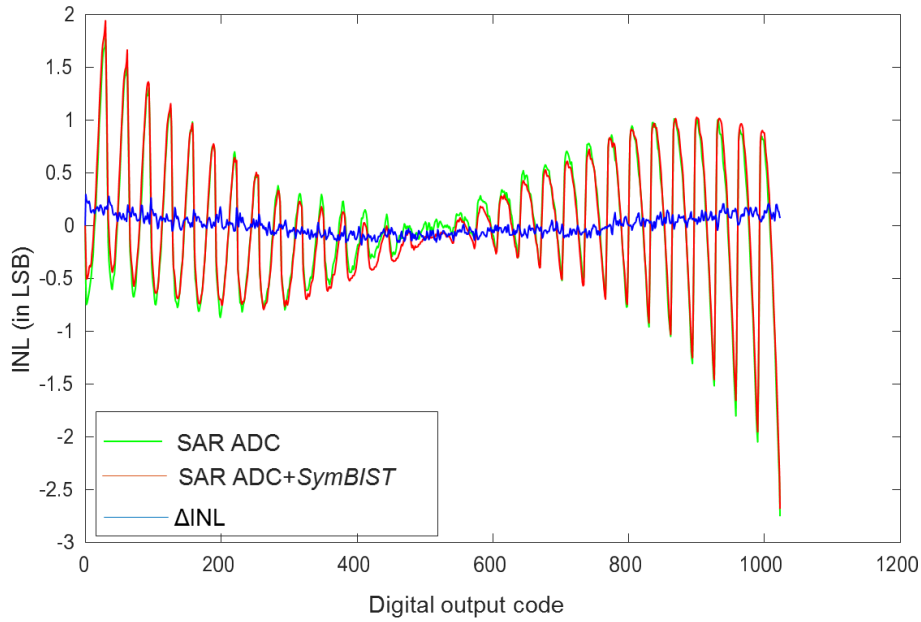


Figure 4.11: INL of original design and original design with embedded *SymBIST*.

the design with embedded *SymBIST* using a ramp histogram test [155]. The result is shown in Figs. 4.10 and 4.11, which also include the Δ of the *DNL* and *INL* curves. As it can be seen, the forms of the curves are similar and practically there is no change in the maximum observed *DNL* and *INL* values. Finally, from the layout of the *SAR ADC IP* that is available, and noticing that the checkers are built using existing blocks of the *SAR ADC IP*, i.e., pre-amplifier, comparator latch, and RS latch, we estimate the area overhead between 5% and 10%.

4.5 CONCLUSIONS

SymBIST is a generic *BIST* paradigm that can be applied to any *A/M-S* circuit class that has symmetries. In this chapter invariances were handcrafted on a *SAR ADC* and the appropriate checker for each invariance was designed. The application of *SymBIST* on the *SAR ADC* showed no performance penalty, while incurring low area overhead. Moreover *SymBIST* required minimum and non-intrusive design re-configuration and is compatible with the modern test access and control mechanisms.

Before moving to describing the results in Chapter 6 it is essential to set the defect simulation environment. In this chapter we will explain the defect simulation workflow that we adopt in this work. We will define the defect model that is used, where a defect is any observable unintended physical change in a circuit, and we will provide an overview of the Tessent®DefectSim mixed-signal defect simulator by Mentor®, A Siemens Business [7], that we used to perform defect simulation at transistor-level in an automated workflow.

5.1 DEFECT SIMULATION WORKFLOW

The defect simulation workflow followed in this work and suggested in [7] and [58], is presented in Figure 5.1. Given a defect model we construct a defect dictionary that contains all possible defects in the CUT. One defect is injected at a time and a test stimulus is applied in the CUT. Then the BIST signatures are generated and are used to distinguish the defective circuits from the functional ones. Once all defects are injected the defect coverage can be calculated.

5.2 DEFECT MODEL

As is common to all fault injection experiments for Very Large-Scale Integration (VLSI) circuits, we make a single fault assumption, that is only one defect will occur at a time [7]. All defect simulators, including the Tessent®DefectSim tool, use this single fault assumption for three reasons. First, when the yield is high enough to be economic, it is assumed that the probability of having multiple defects on a single die is negligible. Second, a test that can detect a specific defect will most likely detect any set of defects that include that defect. Third, the defect simulation time would explode and defect simulation would be intractable if multiple defect combinations were to be considered.

We adopt a standard defect model, which is also the default defect model used by the Tessent®DefectSim tool [7] and is shown in Figure 5.2. In particular, for Metal–Oxide–Semiconductor (MOS) transistors traditionally six defects are injected, i.e., shorts across gate-to-source, gate-to-drain, and drain-to-source, and opens in each terminal. However, all shorts have a similar effect on the transistor being stuck-on and all opens have a similar effect on the transistor being stuck-off. Thus, for MOS

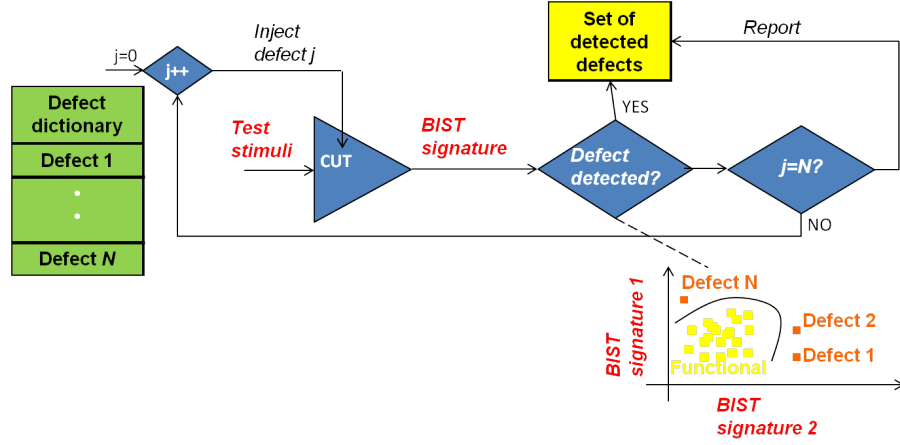


Figure 5.1: Defect Simulation Workflow.

transistors we use only gate open and drain-to-source short defects, as suggested in [7]. Similarly, for Bipolar transistors, we consider base open and collector-emitter short defects. For diodes, we consider open and short defects. Regarding shorts, the defect resistance varies in practice. To avoid simulating many defects, we consider the default resistance of 10Ω . Regarding opens, a weak PU or PD is assigned to each open defect to account for the facts that an ideal open does not exist and, besides, it cannot be handled by a Simulation Program with Integrated Circuit Emphasis (SPICE) simulator [7]. For example, for MOS transistors, we rely on the modeling approach in [156] where V_{GS} is a voltage controlled by V_{DS} with a gain proportional to $C_{gdo}/W \cdot L \cdot C_{ox}$. More specifically according to [156] the ratio between V_{GS} and V_{DS} can be expressed as:

$$\frac{C_{gdo}}{WLC_{ox}} \leq \frac{V_{gs}}{V_{ds}} \leq \frac{3C_{gdo}}{2WLC_{ox}} \quad (5.1)$$

For simplicity, the gain coefficient is set to the default value of 0.5 and therefore $V_{GS} = 0.5 * V_{DS}$. For passive elements, i.e., resistors and capacitors, we consider $\pm 50\%$ variations in their nominal value.

5.3 DEFECT SIMULATOR

Performing large-scale defect simulations, even for small circuits, is usually very time-consuming requires. The total defect simulation time mainly depends on three parameters: (a) the response time of the BIST; (b) the total number of defects; (c) the defects that are chosen to be simulated. The response time of *SymBIST* is very small, as explained in Section 4.4.1 and a simplified defect model is used, as explained in Section 5.2 in order to decrease the total number of defects in a given CUT. But even under these conditions we cannot simulate all the defects since the simulation




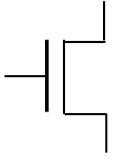
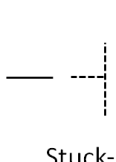
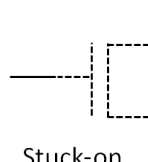
Type of component	Original	Defects	
Passive Components (Resistors, Capacitances, Inductors)	X 	$X(1-K)$ 	$X(1+K)$ 
Active Components (Transistors)		 Stuck-off	 Stuck-on R

Figure 5.2: Defect model from [7].

time is too long. In this section we will analyse the options used in Tessent®DefectSim tool, focusing on those that aim to decrease the total simulation time.

5.3.1 Sampling Techniques in previous Fault Simulators before Tessent®DefectSim

As explained earlier the number of potential defects in a mixed-signal circuit is very large and it is computationally too expensive, if not impossible, to simulate all possible defects. Therefore, the defect simulator samples a smaller set of defects in order to minimize the number of defects that will be simulated with the aim to make defect simulation tractable.

Previous papers analysing fault sampling techniques, published before the work described in [1], have mainly used simple random sampling or stratified sampling [157]–[160] and assumed that the defects have equal likelihood of occurrence. Assuming that the defects are equally possible significantly simplifies the statistical analysis, but actually, due to the wide range of defect densities and critical areas this assumption does not hold true for the defect oriented faults [161]. Stratified sampling has also been used to sample non-equally probable faults, where each stratum consists of faults that have similar likelihood of occurrence [162]. However, dividing the population into strata is usually a heuristic process and cannot be easily automated.

5.3.2 Likelihood-Weighted Random Sampling (LWRS)

Tessent®DefectSim uses Likelihood-Weighted Random Sampling (LWRS) in order to reduce the number of defects to be simulated. Defects are assigned a Relative Likelihood (RL) of occurrence that is estimated by combining global defect-type likelihoods, i.e., the likelihood of short-circuits

is typically higher than the likelihood of open-circuits, and component-specific likelihoods, i.e., the expected component area on the layout, as explained in [1] and [7]. Moreover, the likelihood of each defect varies from process to process.

The sampling of the defects should be random in order to minimize the possibility for selection bias. We should note that the user provides the defect model, the **RL** of each defect and the number of defects to be simulated, but it is the defect simulator that randomly selects the particular defects that will be simulated. In the following analysis N denotes the total number of defects in a **CUT** given a specific defect model, whereas n denotes the number of defects that the user requests to be simulated.

The defect simulator uses an algorithm described in algorithm 1 in order to guarantee that the defects with very high **RL** value will be certainly chosen and that the rest of the defects will be randomly selected. Firstly, the sum of all the **RL** values is calculated. Afterwards the **RL** value of each defect is compared to $\frac{RLSum}{n}$ and if it is greater, then this defect is certainly chosen, and **RLSum** is recomputed considering the value 1 instead of the **RL** of this defect that is certainly chosen. Finally, any defect whose **RL** is greater than $rand * \frac{RLSum}{n}$, where $rand$ is a random value between 0 and 1, is selected.

Once the **RL** of each defect is computed and the defects are randomly sampled, the next step is to estimate the defect coverage. Since each defect has a different **RL** value, instead of considering the absolute defect coverage we calculate the **L-W** defect coverage.

The absolute defect coverage is simply the percentage of detected defects over the total number of simulated defects, given by the following equation:

$$c_{absolute} = \frac{n_{detected}}{n}, \quad (5.2)$$

where $c_{absolute}$ is the absolute defect coverage, $n_{detected}$ is the number of detected defects and n is the total number of simulated defects. This coverage is a good metric when all potential defects are simulated and the defects are assumed to have equal probability of occurrence. Therefore there is a need for calculating a **L-W** defect coverage, so as to take into account also the likelihood of occurrence of each defect. One simple equation estimating the **L-W** defect coverage is:

$$c_{lweighted} = \frac{\sum_{i=1}^n \begin{cases} RL_i & , \text{if selected and detected} \\ 0 & , \text{if undetected} \end{cases}}{\sum_{i=1}^n RL_i, \text{if selected}} \quad (5.3)$$

Algorithm 1: Random Sampling of Defects from [1]

```

input :N,n,RLi
/* N: the total number of defects
   n: the number of defects that the user has requested to be
   simulated
   RLi: the relative-likelihood of the i-th defect, where i=1,..,N */
output:selectCertainList,selectRandomList
/* selectCertainList: a list with the defects that will be certainly
   selected by the simulator
   selectRandomList: a list with the defects that will be randomly
   selected by the simulator */

RLSum = 0 ;
for i = 1 to N do
  | RLSum = RLSum + RLi ;
end
selectCertainList=[] ;
/* Create an empty list in which we will add all the defects that
   will be certainly chosen by the defect simulator */
for i = 1 to N do
  | if RLi >  $\frac{RLSum}{n}$  then
    | add i-th defect to selectCertainList ;
    | RLSum = RLSum - RLi + 1 ;
    | /* Re-compute RLSum using for the RL that will be certainly
       | chosen the value 1 instead of RLi */
  | end
end
randvalue = rand ;
/* rand function returns a random number in the interval [0,1] with
   uniform distribution */
selectRandomList=[] ;
/* Create an empty list in which we will add all the defects that
   will be randomly chosen by the defect simulator */
for i = 1 to N do
  | if RLi > randvalue *  $\frac{RLsum}{n}$  then
    | add i-th defect to selectRandomList ;
  | end
end

```

which is actually the percentage of the RL of each detected defect over the sum of the RLs of all the defects. This defect coverage is efficient when all possible defects are simulated [163], but not when the defects are randomly selected as described in algorithm 1. Instead, taking into account that the RL of each defect can be scaled between 0 and 1, defects with very high RL value will be certainly chosen and defects that are randomly sampled represent a large number of defects, Equation 5.3 can be modified and we can obtain a more complete form:

$$c_{\text{lweighted}} = \frac{\sum_{i=1}^n \begin{cases} \text{RL}_i & , \text{if detected and selected certainly} \\ 1 & , \text{if detected and selected randomly} \\ 0 & , \text{if undetected} \end{cases}}{\sum_{i=1}^n \begin{cases} \text{RL}_i & , \text{if selected certainly} \\ 1 & , \text{if selected randomly} \end{cases}}, \quad (5.4)$$

In Equation 5.4 the RL value of the defects that will be certainly chosen is used as is, whereas for the defects that are randomly chosen over a large number of defects instead of using their RL value we use the value 1. This way it is guaranteed that all the randomly selected defects have a unit weight when calculating the L-W defect coverage.

The defect simulator also reports the 95% confidence interval of the L-W defect coverage, which is the area under the Gaussian probability density function (PDF) that contains 95% of the total area. Actually the 95% confidence interval sets a value range, which is 95% probable that it will contain the true L-W defect coverage. In [164] the 95% confidence interval is computed using uniform likelihood of selection and in [1] the same equations are extended for unequal likelihoods of selection, which is the case in the LWRS. Both works conclude that the 95% confidence interval is $c_{\text{lweighted}} \pm (1.96s + \frac{1}{2n})$, where $c_{\text{lweighted}}$ is the L-W defect coverage computed by the simulator and given by Equation 5.4, s is the standard deviation of the samples and $\frac{1}{2n}$ is an approximation term.

Additionally for mission-critical applications, 99% confidence interval of the L-W defect coverage can be reported which is equal to $c_{\text{lweighted}} \pm (2.58s + \frac{1}{2n})$ [164], [1].

5.3.3 Additional Settings that Reduce the Defect Simulation Time

To further reduce defect simulation time, we use the stop-on-detection option of the tool. With the stop-on-detection option, while defect simulation progresses, the simulation of a defect is stopped as soon as a test parameter is violated and the simulation of the next defect in the list begins.

More specifically, as explained in Section 3.1, the output of each checker is combined to provide a single 1-bit pass/fail output. As soon as the *SymBIST* output indicates defect detection, a defect is labelled as detected and the simulation for this particular defect stops. In contrast, functional BIST approaches aiming at measuring the performances, require the same simulation time for each defect, since all data should first be collected before calculating the performances. For example, the simulation time of a defect that has a catastrophic impact on the circuit would be the same with that of a defect that would have no impact on the circuit. Therefore the stop-on-detection option wouldn't accelerate the total simulation time.

Finally, since all defects are independent, they can be simulated in parallel using all Central Processing Unit (CPU) cores available in the computer.

5.3.4 Defect Injection in Tessent®DefectSim

In this section we will analyse the way the simulator handles the defect injection in the CUT. In Listings 5.1 and 5.2 two simplified defect models are presented in SPICE language.

The first Listing 5.1 represents a drain to source short circuit. This defect is simply modelled by an additional defective resistor equal to 10Ω between the drain and the source of a MOS transistor. The next step is to calculate the RL of this specific defect. First we need to check whether the drain and the source of the transistors are already shorted by design, i.e., to create a MOS capacitor [165]. In this case it doesn't make sense to add a defective resistance between the source and the drain of the transistor and thus we set the RL of this defect equal to 0 so as to avoid selecting this defect during random sampling. If the drain and the source of the MOS transistor are not already shorted, then the RL of this defect is set equal to $m * w * l * k$, where m is the number of parallel fingers, w is the width of the MOS transistor, l is the length of the MOS transistor and k is a factor that depends on the model of the transistor and on the specific component. This k factor allows the user to set different RL values for N-Channel Metal-Oxide Semiconductor (NMOS) and PMOS transistors or even different RL values for different types of NMOS or PMOS transistors in the same process.

```

<component M, defect_model pre_ds_short>

$defect_description= MOS drain-source short
$defect_type=short

Rdefect $node1 $node3 10
*Short circuit between drain and source modelled with a 10 ohm resistor
*A transistor is defined as follows: M_NMOS D G S B with D,G,S,B nodes
being the drain, the gate, the source and the bulk nodes of the n-mos
transistor respectively
$RL = [if {$node1 eq $node3} {
  puts "0"
} else {
  puts " $multiplier * $width * $length * $k "
}
*k is a variable that depends on the global defect-type likelihood and on
component-specific likelihood
}]
.ends

```

Listing 5.1: Short circuit injection in a MOS transistor

Similarly Listing 5.2 shows a +50% variation in the value of a resistance. Its *RL* value is set equal to value * *k*, where value is the initial value of the resistance and *k* is a factor that depends on the resistance model.

```

<component R, defect_model preHRL>

$defect_description = 50% R increase
$defect_type = variation

R_SHORT $node1 $node2 ${SpiceSpectreFormat(R)}[expr (1 + 50/100.0) *
  $value ]
$RL puts " $ $value * $k "

<endcomponent R>

```

Listing 5.2: Injection of a +50% variation in a resistor

Figure 5.3 shows an example of injecting the two above mentioned defect types in a simple voltage divider controlled by a MOS transistor operating as a switch. Since we have two resistors and one MOS transistor and taking into account only the defects described in Listings 5.1 and 5.2 we expect to have 3 defects in total, i.e., 1 per component. Let's assume that the test parameter is V_{cm} and that it should lie within a tolerance window, as described in Section 3.1. The defect simulator will first simulate the defect free case study shown in a). Once it confirms that the test parameter is within the tolerance window, it will start injecting the defects shown in Figure 5.3 b), c) and d), that are a short circuit between the drain and the source of the MOS transistor, a +50% variation in the resistance R_1 and a +50% variation in the resistance R_2 respectively. Before simulating the defects, the simulator will calculate the

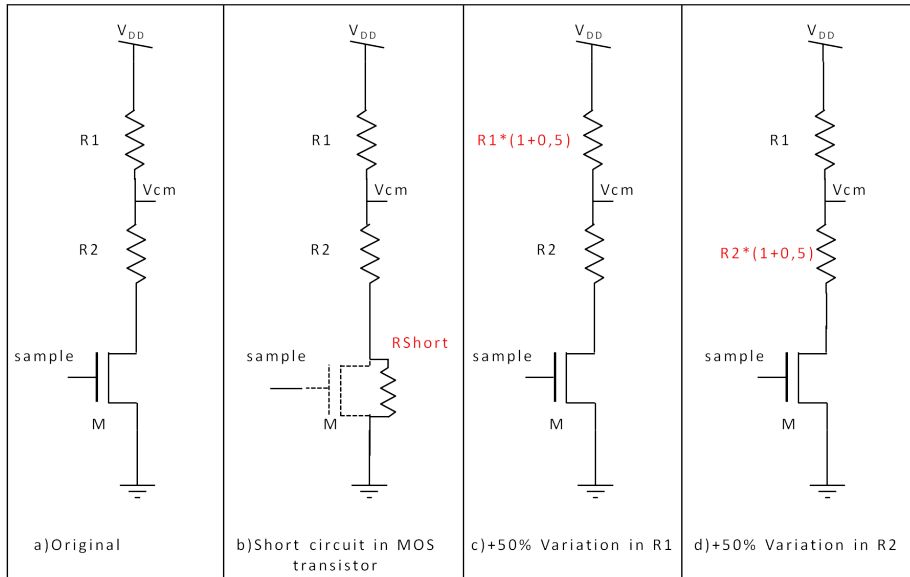


Figure 5.3: Injection of defects 5.1 and 5.2.

RL of each defect. Then the simulator will randomly sample the defects to be simulated. In this example we only have three potential defects, so all of them will be simulated and the random sampling will not take place. These defects are independent and can be simulated in parallel. From the outcome of all the defect simulations we can calculate the RL defect coverage given by the Equation 5.4.

For more details regarding the defect models and the way they are injected into the CUT the reader is referred to the Tessent®DefectSim user manual [166].

5.4 CONCLUSIONS

In this chapter, we addressed the obstacles for performing large-scale defect simulations. The main challenges are to deal with the large number of defects and the resultant time-consuming simulations. We described the defect simulation workflow, as well as the defect model that will be used in Chapter 6 to estimate the L-W defect coverage for our case study. We described the main options of the Tessent®DefectSim mixed-signal defect simulator by Mentor®, that were used to make the defect simulation tractable. These include LWRS, reporting L-W defect coverage, using stop-on-detection and simulating the defects in parallel. Finally, we described the way the defect simulator handles the defect injection and illustrated the defect injection in one simple case study.

RESULTS OF SYMBIST APPLICATION TO THE SAR ADC IP

In the previous chapters, we described the generic *SymBIST* paradigm for A/M-S ICs and the way it is applied to a real SAR ADC IP by STM. In this chapter, we will present the results obtained for the use of *SymBIST* in the context of off-line testing, online testing, and fault diagnosis.

6.1 CHECKER SELF-TEST

The first step is to evaluate the self-test approach of the analog checker in Figure 4.7 proposed in Section 4.3.3. We performed a defect simulation of the checker considering all possible defects within the checker according to the defect model discussed in Section 5.2, for a total of 176 defects. We considered sequences of DC tests that use 18 different combinations of DC values for the four checker inputs, namely V_1 , V_2 , UTL, and LTL, as discussed in Section 4.3.3. The resultant L-W defect coverage is 94.45%. Thereafter, we searched to eliminate tests such that the set of retained tests achieves the same fault coverage. It turns out that $N = 3$ tests suffice to achieve the same fault coverage. These are: $\{V_1, V_2, k\} = \{VDD, VDD, 3\}$, $\{VREF[32], GND, 3\}$, $\{GND, GND, 3\}$.

6.2 OFF-LINE TESTING USING SYMBIST

Herein, we present results for the use of *SymBIST* for off-line post-manufacturing defect-oriented test. In Section 6.2.1, we will present the specific test stimulus used in the off-line test mode. In Section 6.2.2, we set the checker comparison windows for the desired test coverage versus yield loss trade-off. In Section 6.2.3, we show transient analysis of *SymBIST* and, finally, in Section 6.2.4, we will present the L-W defect coverage results for our case study by employing the defect simulator described in Section 5.3.

6.2.1 Test Stimulus

Figure 4.6 shows the on-chip generation of the dynamic test stimulus together with the required re-configurations. This dynamic test stimulus is used in the off-line defect-oriented test mode and for diagnosis. Specifically, for the off-line defect-oriented test mode the same test stimulus is used to exercise all the invariances. As discussed in Section 4.3.1, it is composed of a static and a dynamic part. The static part is a DC value

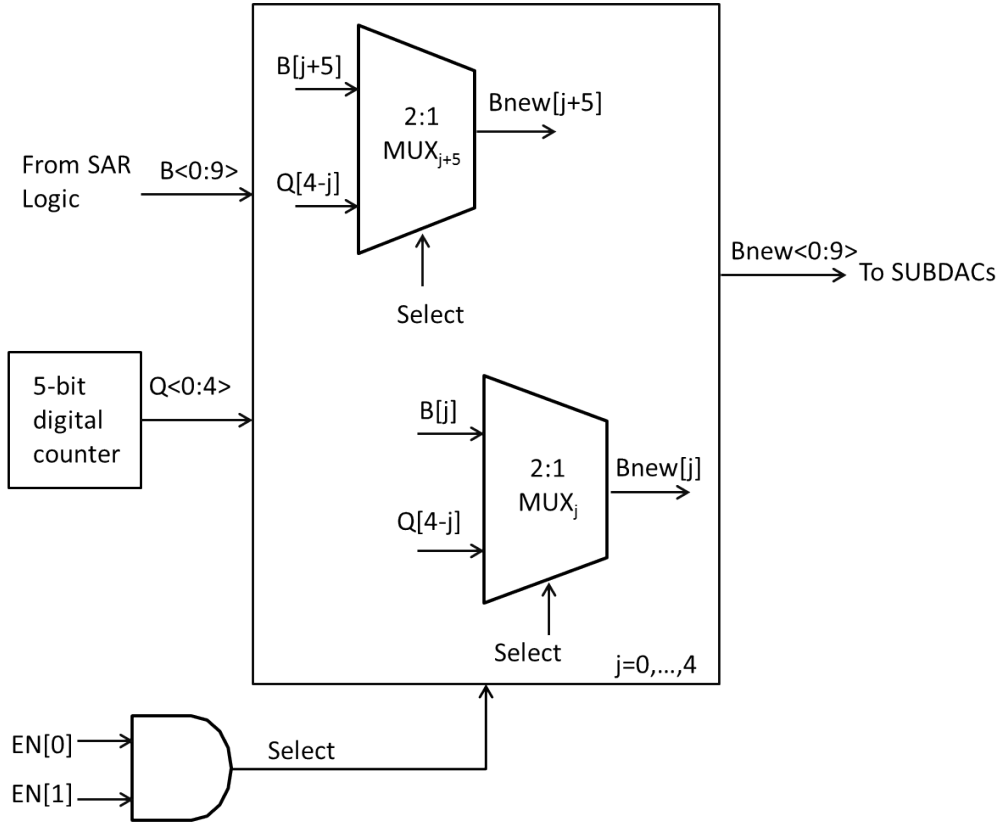


Figure 6.1: Re-configuration for applying the test stimulus in off-line test mode.

applied to the input of the SAR ADC which can be set arbitrarily. Herein, we use $\Delta IN_{DC} = 0.1V$. The dynamic part is a digital signal applied to the input of the two sub-DACs. In particular, the 5-bit digital counter cycles through all possible 2^5 bit combinations.

The simplified non-intrusive re-configuration to enable the off-line test mode is shown in Figure 6.1. Compared to Figure 4.6, we can program the shuffling block to implement just one cycle that is kept constant during the whole off-line test. In off-line test mode, i.e., $EN<0:1> = 11$, the 5-bit digital counter drives simultaneously both sub-DACs. In our experiments, we observed that randomly cycling through the 2^5 bit combinations at the inputs of the sub-DACs, as opposed to incremental counting, results in higher defect coverage. Intuitively, this is because the two sub-DACs are exercised more intensively generating large steps at their outputs. Figure 6.1 shows the configuration that was implemented, e.g. $Bnew[j+5]=Q[4-j]$ for SUBDAC₁ and $Bnew[j]=Q[4-j]$ for SUBDAC₂, $j = 0, \dots, 4$. For example, for SUBDAC₁, the input sequence is $\{2^4, 2^3, 2^4 + 2^3, 2^2, 2^2 + 2^4, \dots\}$.

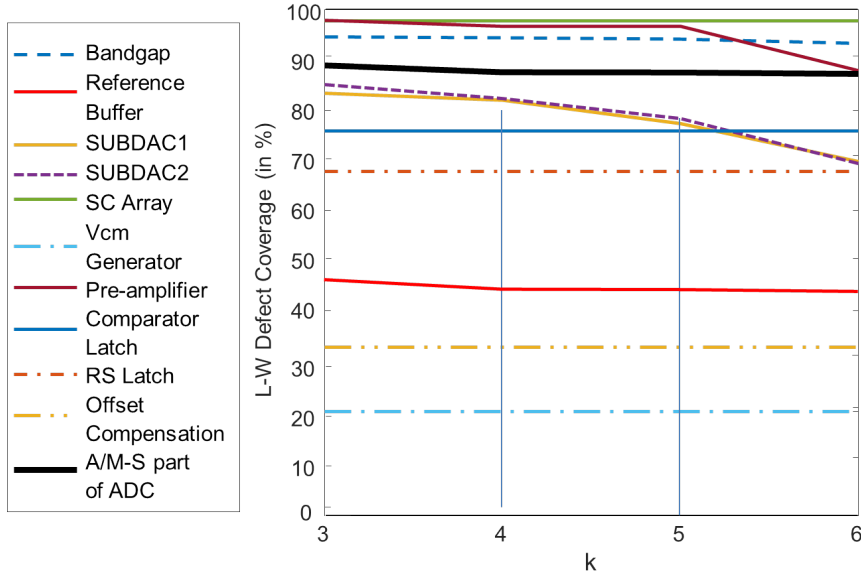


Figure 6.2: L-W defect coverage as a function of tolerance window placement.

6.2.2 Setting the Comparison Window for Desired Test Coverage vs. Yield Loss Trade-off

Herein, we study the effect of the width of the comparison window, which is set by the coefficient k , on the trade-off between yield loss and defect coverage.

Figure 6.2 shows the L-W defect coverage values achieved with *SymBIST* for the individual blocks of the SAR ADC IP and for its complete A/M-S part as a function of k , where k is varied from 3 to 6. μ and σ are computed based on a defect-free MC analysis with 100 runs. For simplicity, Figure 6.2 does not include the 95% confidence intervals.

As it can be seen, the expected drop of L-W defect coverage as we increase k is evident only for three blocks, namely the pre-amplifier and the two sub-DACs, while for the rest of the blocks the curves are practically flat and for the entire A/M-S part only a slight drop is observed. This implies that the majority of the detectable defects result in substantial deviation of the invariance outside its tolerance window, thus by enlarging the window we can reduce yield loss probability without inadvertently increasing test escapes. The fact that the L-W defect coverage curve of the A/M-S part does not follow the drop observed for the pre-amplifier and the two sub-DACs is due to LWRS. In particular, defects within these blocks have lower likelihood of occurrence compared to defects in other blocks and, thereby, they are less frequently sampled to estimate the L-W defect coverage of the A/M-S part.

The observed L-W defect coverage curve of the A/M-S part helps us to draw two important conclusions. First, as already mentioned, we can use a wide comparison window to minimize yield loss without affecting

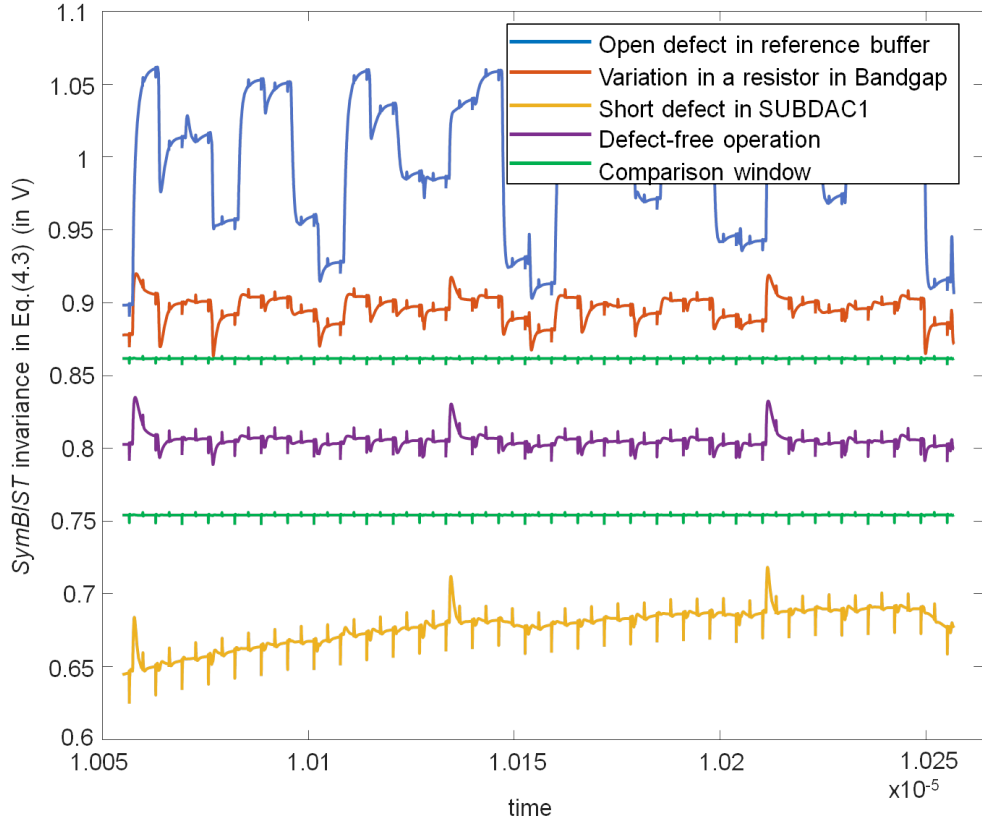


Figure 6.3: Transient simulation of *SymbIST* invariance in Equation (4.3) for different defect scenarios and blocks.

defect coverage. In this regard, a value of $k = 5$ is a good compromise since it guarantees a negligible yield loss. Having verified that invariant signals follow a normal distribution, for $k = 5$ the expected fraction of functional defect-free circuits within the comparison window will be 99.9999426%. In other words, the false positive rate will be $5.74 \cdot 10^{-5}$. Second, the fact that the curve drops with a slight rate as we increase k implies that in a neighborhood of k the *L-W* defect coverage is practically constant. As a result, the *UTL* and *LTL* of the comparison window of the checkers do not have to be precisely set, thus making the *SymbIST* infrastructure overall robust.

In Section 6.2.4, we will analyze in detail the defect coverage for $k = 5$.

6.2.3 *SymbIST* Transient Simulations in Off-line Test Mode

Herein, we show transient simulations of *SymbIST* for the *SAR ADC IP* in off-line test mode. In all simulations, the comparison window is set for $k = 5$. Moreover, the stop-on-detection option was disabled.

Figure 6.3 shows the *SymbIST* invariance in Equation (4.3) during the whole duration of the test stimulus for the nominal defect-free case and

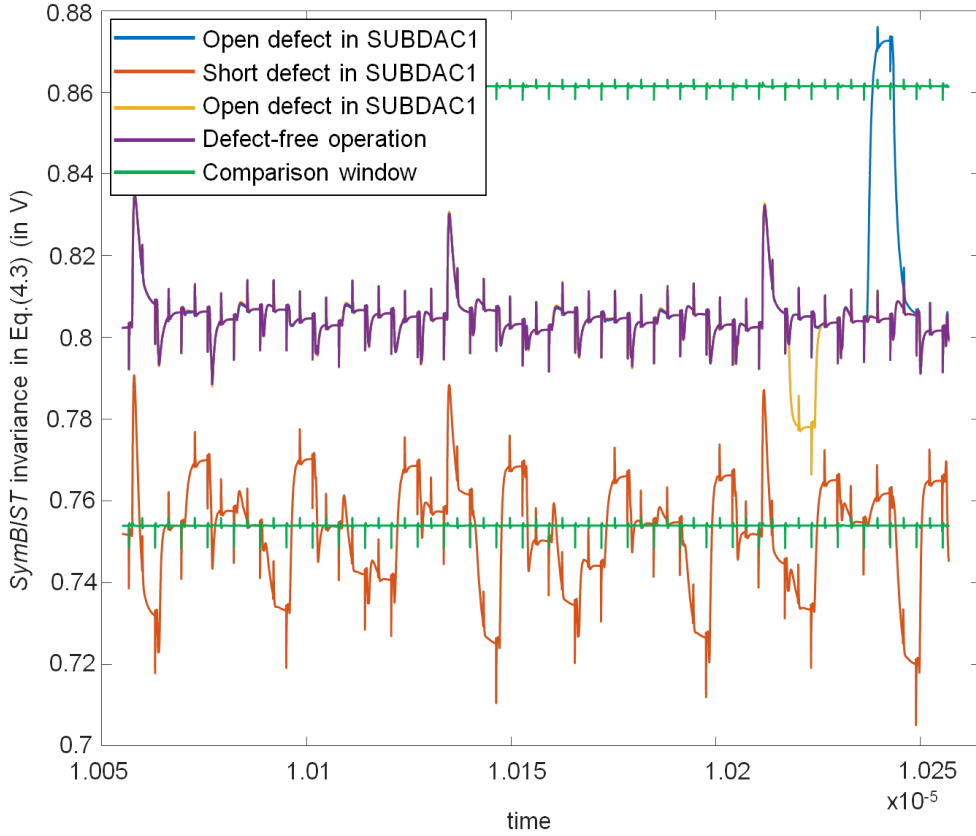


Figure 6.4: Transient simulation of *SymbIST* invariance in Equation (4.3) for different defect scenarios in SUBDAC₁.

for the three randomly chosen defect scenarios. The maximum duration of the test is $0.205\mu\text{s}$ and equals the duration of the dynamic part of the test stimulus which consists of applying 2^5 different bit combinations at the inputs of the two sub-DACs. Since the stop-on-detection option is disabled, after defect detection the test continues until its maximum duration. The instantaneous glitches are due to the switching operation, either due to changes in this digital test stimulus or due to the sampling and conversion operations. As it can be seen from Figure 6.3, in the defect-free case the invariance lies within the comparison window, while in all defect scenarios the invariance is permanently violated during the duration of the test stimulus.

Figure 6.4 shows transient simulations for additional defects in SUBDAC₁ to illustrate scenarios where the invariance is not permanently violated. As it can be seen, the short defect (orange curve) results in invariance violation during several conversion periods. The first open defect (blue curve) results in invariance violation only during one conversion period towards the end of the test stimulus duration, otherwise for the rest of the conversion periods the transient response matches the transient response of the defect-free operation. In contrast, the second

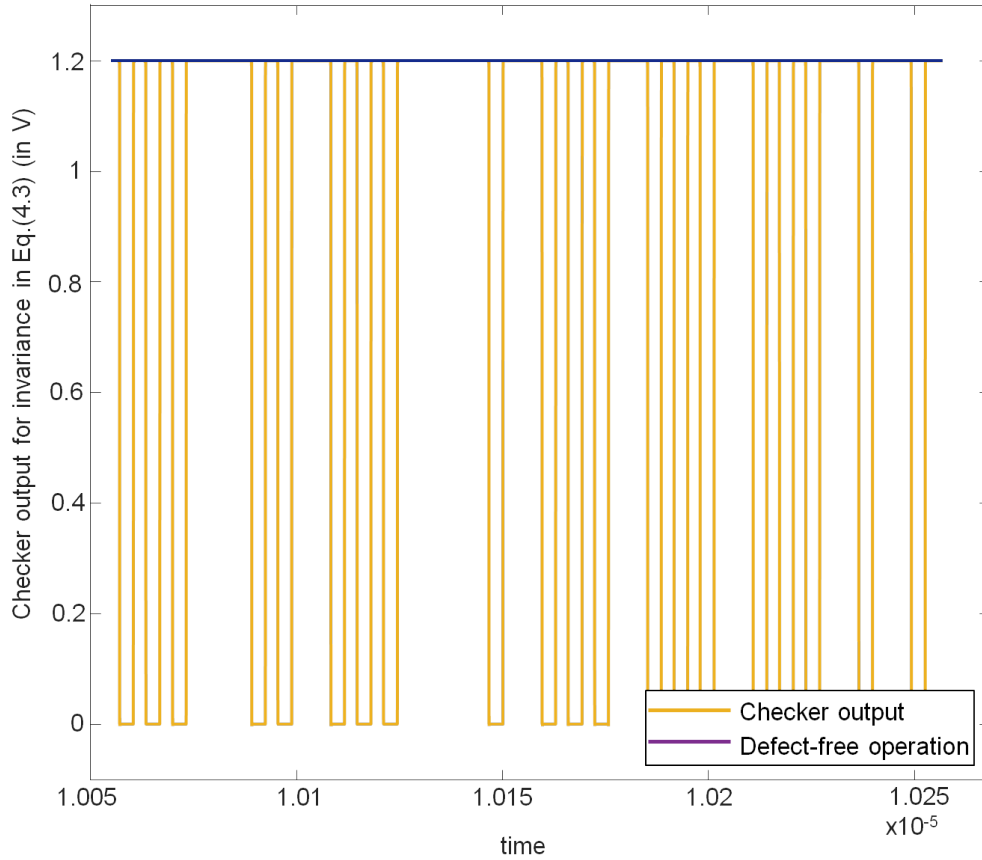


Figure 6.5: Checker output for the short defect simulation in SUBDAC₁ shown in Figure 6.4.

open defect (yellow curve) is activated during one conversion cycle resulting in clear excursion of the invariant signal, but still the signal does not slide outside the window and the defect goes undetected. This last defect simulation shows that certain defects may be undetected due to lenient test limits set at $k = 5$, yet making the test limits strict may result in inadvertent false positives.

Figure 6.5 shows the response of the checker monitoring the invariance in Equation (4.3) for the short defect scenario shown in Figure 6.4. As it can be seen, the output of the checker switches to logical zero at several time stamps during the duration of the test stimulus indicating invariance violation.

6.2.4 Defect Coverage Analysis

Table 6.1 shows for the individual A/M-S blocks of the SAR ADC IP and for its complete A/M-S part the L-W defect coverage values achieved using *SymBIST* with the comparison window set at $k = 5$. Notice that any differences in L-W defect coverage values compared to the results in [167] is due to the fact that herein the simulation considers the full on-chip in-

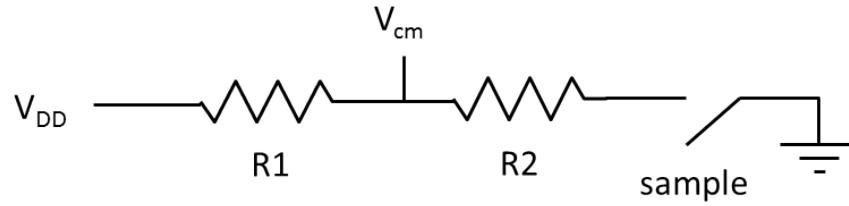
Blocks	# defects	# defects simulated	defect simulation time (sec)	L-W defect coverage for k=5
BandGap	104	104	7305	95.46%
Reference Buffer	160	160	10640	43.32%
SUBDAC ₁	1260	107	7413	77.32%±6.62%
SUBDAC ₂	1260	107	7331	78.01%±6.55%
SC Array	44	44	3139	97.7%
V _{cm} Generator	6	6	591	20%
Preamplifier	24	24	1797	96%
Comparator Latch	38	38	2835	76%
RS Latch	40	40	2899	68%
Offset Compensation circuit	20	20	1400	32.73%
Complete A/M-S part of SAR ADC IP	2956	100	6376	87.56%±4.34%

Table 6.1: L-W defect coverage results with the comparison window set at $k = 5$.

tegration of *SymBIST* at transistor-level into the SAR ADC IP, while in [167] mathematical expressions of the invariances were used in the analysis. For the larger blocks, namely the two sub-DACs, as well as for the complete A/M-S part, we use the *LWRS* option to respect a reasonable defect simulation time budget and, therefore, we report also the corresponding 95% confidence intervals. Table 6.1 includes in addition the total number of defects in a block according to the defect model in Section 5.2, the number of defects simulated which is different than the total number of defects when the *LWRS* option is used as explained in Section 5.3, and the defect simulation time. The defect simulation campaign was performed on a server with 4 cores@3.5 GHz and 7.55 GB RAM. The defect simulation times are proportional to the number of defects simulated, as well to the detection time stamps during the test duration since we are using the stop-on-detection option.

As it can be seen from Table 6.1, for the complete A/M-S part of the SAR ADC IP, the L-W defect coverage is $87.56\% \pm 4.34\%$. As an indicative comparison, for two considerably smaller industrial A/M-S IPs, namely a bandgap and a power-on-reset circuit, the reported defect coverage values are 74% and 51%, respectively [7].

The reported L-W defect coverage of around 87% is considered to be very high. For A/M-S ICs, it is not expected to approach near-100% defect coverage, as is typical of scan-tested digital ICs [7]. The reason is that A/M-S ICs have redundancy and defect tolerance, intentional or not. Undetected defects can be examined one by one, which of course is a very

Figure 6.6: Schematic of V_{cm} generator.

tedious and time-consuming process, in order to report also the modified fault escape rate [8], defined as the percentage of undetected defects that result in at least one specification being violated. As the reason behind an undetected defect is understood, systematic efforts can be made to tune the BIST towards higher defect coverage. For example, this involves designing a new test stimulus to better activate the circuit in the vicinity of the inserted defect and propagating the defective signals at activated defect sites to circuit outputs with sufficient amplitude [7].

On another note, safety standards, e.g., ISO 26262 for automotive, are not written in a quantified way. For BIST certification, the IC manufacturer delivers a safety manual that clearly defines the defect model and the defect coverage accounting method, and describes the results of the analysis done. Moreover, safety-relevant application failure modes defined by the user can be mapped to specific defects, and the analysis should prove that those defects are caught.

To shed more light in the low L-W defect coverage values observed for certain blocks, namely the reference buffer, V_{cm} generator, and offset compensation circuit, we further analyzed their undetected defects. Nevertheless, these rather low L-W defect coverage values do not have a significant impact on the L-W defect coverage for the complete A/M-S part. The reason is that compared to defects in other blocks, most defects within these blocks have considerably lower RL. The V_{cm} generator, shown in Figure 6.6, is a simple voltage divider serially connected to ground with a switch. It is used only during the sampling phase to set V_{cm} equal to $V_{DD} \cdot R_2 / (R_1 + R_2)$, where R_1 and R_2 are the two resistors in the voltage divider. Out of the 6 defects only 1 is undetected, e.g. the absolute defect coverage is 83.33%. The undetected defect is the stuck-on defect in the switch and has a very high RL, thus dominating the L-W defect coverage which is 20%. However, this defect has no effect on the operation of the block since it only forces the V_{cm} generator to operate uninterruptedly, thus it only increases the power consumption of the V_{cm} generator. For the reference buffer, there are two undetected defects with very high RL, namely the $\pm 50\%$ variations in the Miller capacitor of the op-amp inside the reference buffer. This capacitor is used for stability; however, the circuit is working at DC and the comparison levels

$V_{REF} < 0 : 32 >$ manage to settle during the warm-up phase of the ADC despite the $\pm 50\%$ variations. For this reason, these defects have no effect on the operation of this block.

6.3 ONLINE TESTING USING SYMBIST

Since the invariances should hold true independently of the input and the checkers monitor the invariances in real-time without loading the operation of the circuit, *SymBIST* can be used for concurrent error detection with low latency. Herein, we demonstrate *SymBIST* for two scenarios of lifetime failure, namely transient errors and latent defects.

6.3.1 Test Stimulus

During on-line testing we cannot interrupt the operation of the SAR ADC so as to force our desired test stimulus. Therefore in this case the test stimulus is the running input of the SAR ADC, which could be any time-varying signal set by the application. For the purpose of our simulations, we consider a sinusoidal FD analog input with peak-to-peak voltage 1V, common mode 0.6V, and frequency 2MHz.

6.3.2 Transient Error

Transient errors are produced in ICs by signal coupling, electromagnetic interference and power supply or ground bounce [168]. We model a transient error by injecting a short current pulse into a node of the circuit. Figure 6.7 shows the *SymBIST* response to a transient error occurring in the node DAC- of the 10-bit DAC shown in Figure 4.4. The current pulse has amplitude $5\mu\text{A}$ and is injected at around $10.15\mu\text{s}$. Shortly after the invariance in Equation (4.6) slides outside the tolerance window and the global *SymBIST* output switches to logical 0 at time stamps when the test clock is high to indicate the error. As it can be seen, although the pulse has a short duration, *SymBIST* flags an error for all the subsequent conversion periods until a new input analog value is sampled. The reason is that the current pulse charges the capacitors to a different DC level than V_{cm} until the new sampling phase.

A second example of a transient error is shown in Figure 6.8. A current pulse with amplitude $10\mu\text{A}$ is injected in a node inside the bandgap circuit affecting the biasing conditions. Figure 6.8 shows the detection via the *SymBIST* invariant signal in Equation (4.7). *SymBIST* flags an error for a relative long period after the current pulse settles to zero since it requires some time for the bandgap to settle back to the nominal biasing conditions.

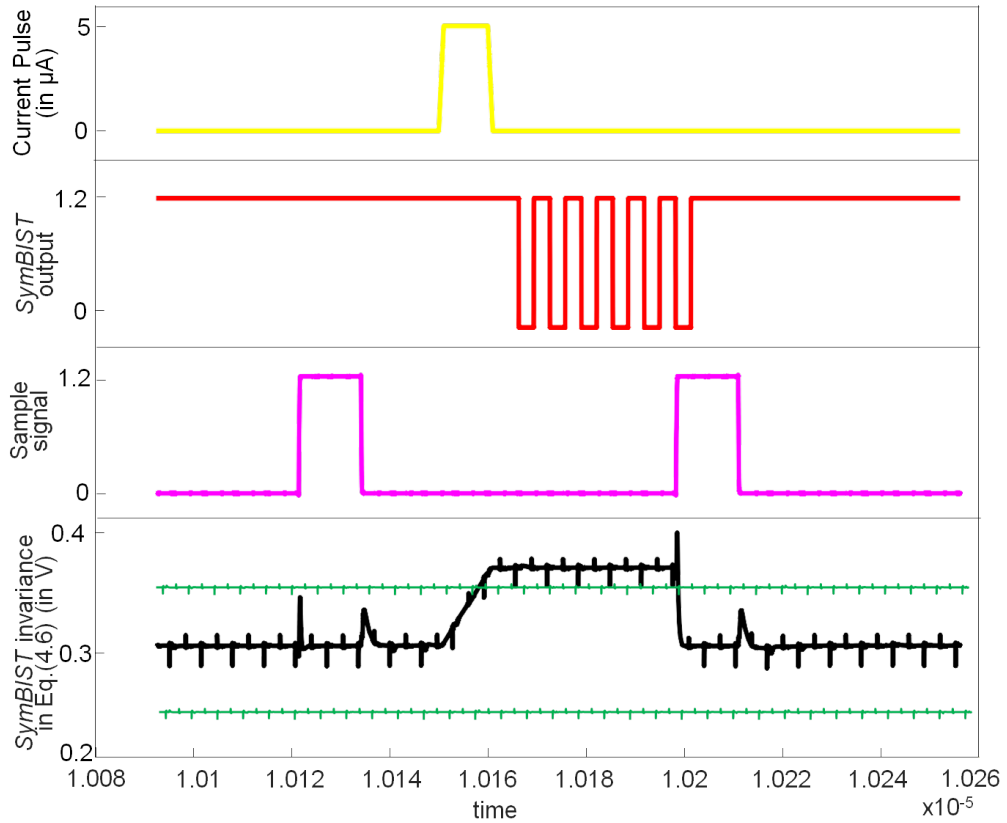


Figure 6.7: *SymbIST* response to transient error injected in the DAC.

6.3.3 Latent Defect

The most common latent defect is the rupture of the gate oxide of MOS transistors known as pinhole which accelerates the TDDB [9]. It has been recently shown that a pinhole can be modeled as a decrease in the effective value of the oxide thickness t_{ox} [9]. Figure 6.9 shows an example where we gradually decrease the t_{ox} of a transistor inside the reference buffer and eventually this decrease is captured by the invariance in Equation (4.3). As it can be seen, as t_{ox} decreases gradually to around 80% of its nominal value, the invariant signal shows an increasing positive offset but still remains within the tolerance window. Further t_{ox} decrease causes an abrupt negative offset to the invariant signal and when t_{ox} drops below 76% of its nominal value the invariant signal abruptly drops below the lower bound of the tolerance window.

6.4 BIST-ASSISTED ANALOG FAULT DIAGNOSIS

We assume that *SymbIST* is already in place for post-manufacturing testing and is re-used “as is”, for analog diagnosis. In Section 6.4.1, we will

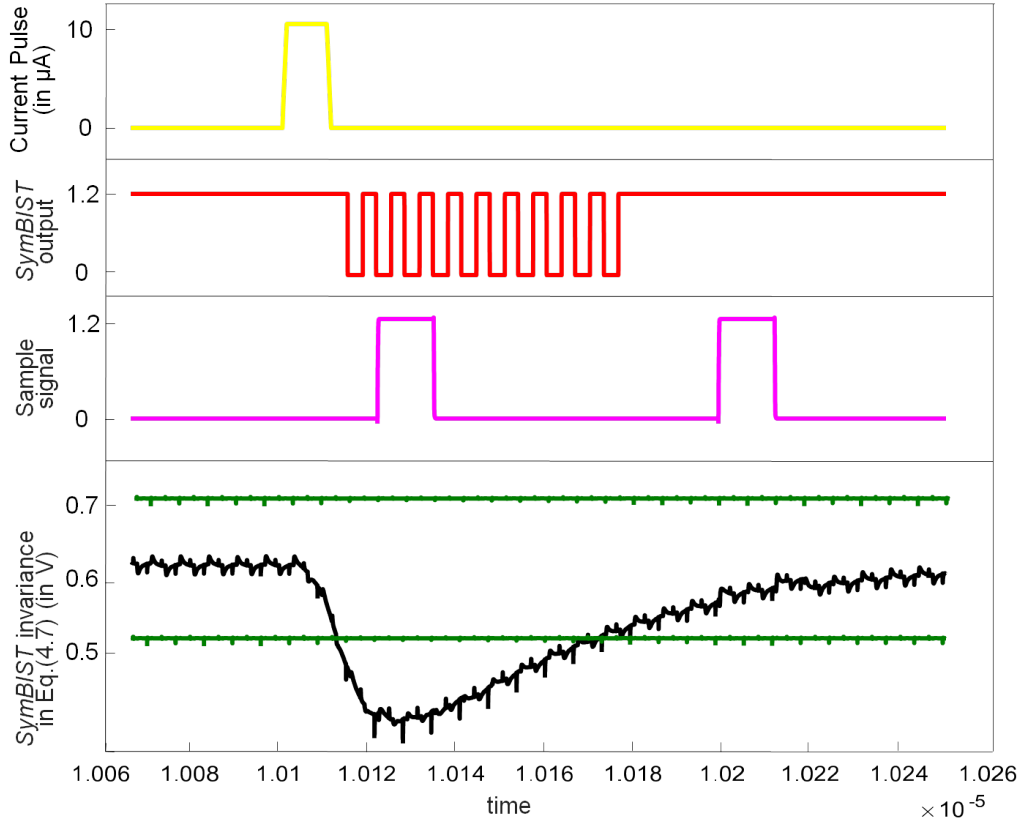


Figure 6.8: *SymBIST* response to transient error injected in the bandgap.

explain the way the *SymBIST* setups are chosen and in Section 6.4.2, we will present the diagnosis results.

6.4.1 *SymBIST* and other Diagnosis Setups

A *SymBIST* setup is composed of the triplet:

$$S^\ell = \{\Delta IN_{DC}^\ell, Q^\ell < 0 : 9 >, \delta^\ell\}, \quad (6.1)$$

where ℓ denotes the setup index and δ^ℓ is a 2×5 vector containing the lower and upper limits of the tolerance window of the checkers monitoring the invariances in Equations (4.3)-(4.8).

Each checker provides a 32-bit diagnostic measurement pattern. Their concatenation results in a diagnostic measurement pattern of length $6 \times 32 = 192$ bits per *SymBIST* setup.

The first *SymBIST* setup is the same one used for post-manufacturing defect-oriented test, as explained in Section 6.2.1. This setup uses $\Delta IN_{DC} = 0.1V$, the tolerance windows for the invariances in Equations (4.3)-(4.8) are placed at $\pm 5 \cdot \sigma$ so as to have negligible yield loss, and the implemented shuffling in Figure 4.6 is $Q[j+5]=W[4-j]$ for SUBDAC₁ and

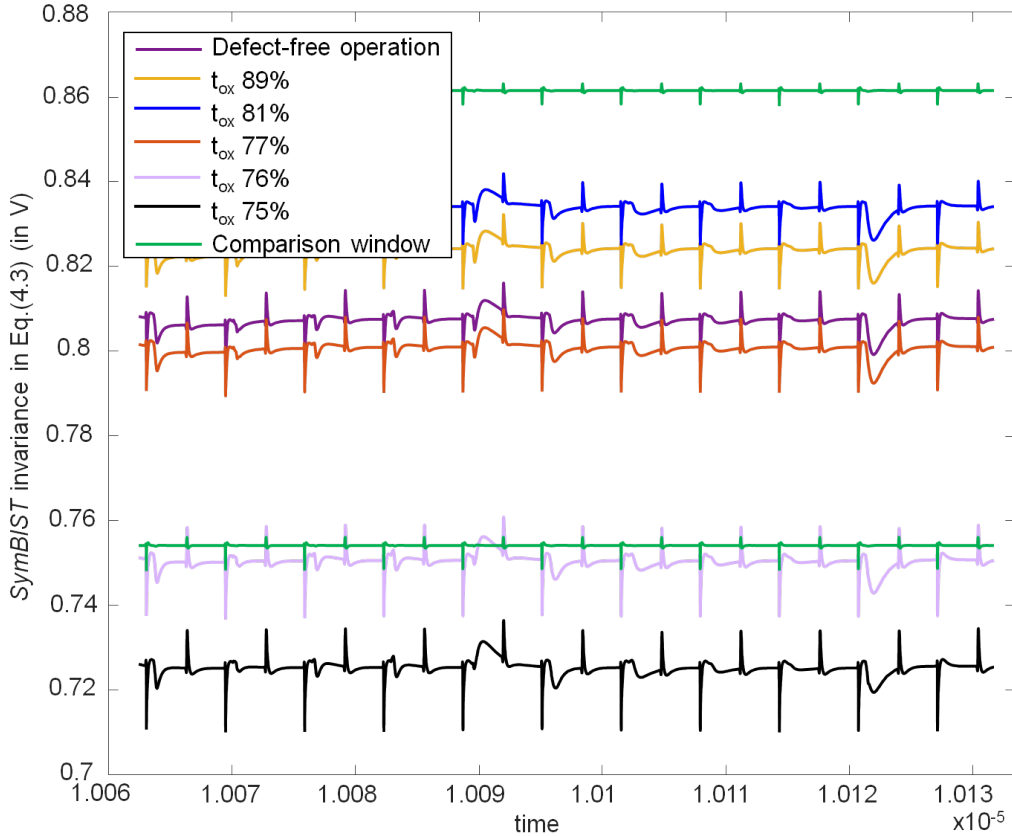


Figure 6.9: *SymbIST* response to a latent defect.

$Q[j]=W[4-j]$ for SUBDAC₂, $j = 0, \dots, 4$. For example, for SUBDAC₁, the input sequence is $\{2^4, 2^3, 2^4 + 2^3, 2^2, 2^2 + 2^4, \dots\}$. Then, additional *SymbIST* setups are used to resolve ambiguity groups. These *SymbIST* setups have a different combination of DC FD input, DAC input sequence, and tolerance window widths for each checker. Generating the fault dictionary matrix in Equation (3.6) for one *SymbIST* setup takes roughly one day. Thus, using an optimization algorithm to search in the space of *SymbIST* setups is computationally expensive, even if this is an off-line analysis. To this end, the next *SymbIST* setup is crafted based on the unresolved ambiguity groups so far with the aim to split them into smaller size groups, preferably uniquely detecting the defects composing them. For example, we can try a different input DAC sequence to split an ambiguity group with defects into the SUBDACs, we can increase (decrease) the upper (lower) tolerance window limit for an ambiguity group where the defects cause the invariance to exceed the nominal upper (lower) tolerance window limit, etc. We stop adding *SymbIST* setups when diagnosis resolution stops improving. After we stop adding *SymbIST* setups, we start adding additional test setups compatible with *SymbIST* in order to resolve the ambiguity groups.

Our first idea is to extract two diagnostic measurement patterns per defect per invariance by considering separately the two tolerance window limits. If an invariance violates permanently either the upper or the lower limit, the diagnostic measurement pattern is in both cases a vector of ones. If we consider separately the two tolerance window limits, then we can distinguish two defects where one violates the upper and the other the lower limit. This effectively doubles the size of the diagnostic measurement pattern per checker. Note that this is also a *SymBIST* setup and the diagnostic measurement is extracted from the *SymBIST* output.

Our second idea is to use the existing VREFP test pin, shown in the top-level architecture of Figure 4.2. This pin outputs the VREF[32] reference voltage of the reference buffer used by the DAC during the conversion. We use the DC VREFP measurement as an analog diagnostic measurement. The nominal VREFP is 0.5V and we consider a nominal tolerance window set at $\pm 5 \cdot \sigma$. Let $VREFP(F_j)$ denote the diagnostic measurement value for defect F_j . Considering a range of 20mV centered at each $VREFP(F_j)$, a defect is detectable if its corresponding range does not overlap with the nominal tolerance window, and two defects are distinguishable if their corresponding ranges do not overlap.

Our third idea is to use internal re-configuration. We use the topology modification approach proposed in [16]. The underlying idea is to connect PU PMOS and PD NMOS transistors that bring an internal node to VDD or ground, respectively. This changes the topology of the circuit and potentially can expose defects differently. This topology modification approach is another BIST approach that can be used to test analog ICs. Hence, *SymBIST* can co-exist with this topology modification approach for diagnosis purposes. The PD and PU transistors will be activated to test the SAR ADC and the *SymBIST* will act as a diagnosis measurement extractor.

6.4.2 Diagnosis Results

Figure 6.10 shows a transistor level simulation for a +50% and a -50% variation defect in a resistor within the Bandgap. We simulated each defect using two different *SymBIST* setups 1 and 4, which differ only in the lower and the upper limits of the tolerance window of the checkers. More specifically, for *SymBIST* setup 1 the limits are set to $\pm 5 \cdot \sigma$ and for *SymBIST* setup 4 the limits are set to $\pm 6 \cdot \sigma$, as shown in the middle subplot with the green and the pink colour, respectively. The middle subplot also shows the invariant signal in Equation 4.4 for the defect-free case with the red curve and the two defective cases with the yellow and the black curve, respectively. The invariant signal of each defect remains the same for both *SymBIST* setups, since these setups only differ in the tolerance window of the checkers and this doesn't affect the invariant signals at all. So the invariant signal of each defect remains constant for the two

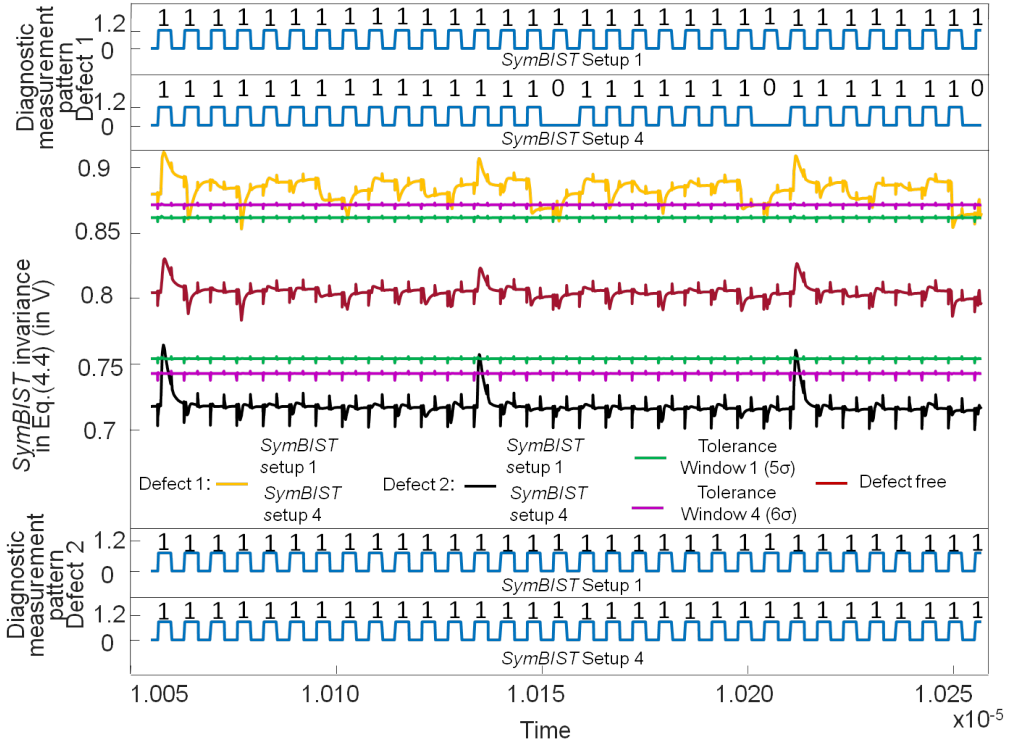


Figure 6.10: Transient simulation of *SymbIST* invariance in Equation 4.4 for $\pm 50\%$ variation defect in a resistor in Bandgap, considering two different *SymbIST* setups.

different *SymbIST* setups, but since the comparison window alters we expect to have a different *SymbIST* response per setup. The top and the bottom subplots show the diagnostic measurement pattern of each defect for these two setups. In *SymbIST* setup 1, for both defects the invariance violates either the lower or the upper limit of the tolerance window 1, thus their diagnostic measurement pattern in the same, i.e., a vector of ones, and the two defects are indistinguishable. With *SymbIST* setup 4, the invariance of defect 1 slides inside the tolerance window 4 at specific time stamps setting the corresponding *SymbIST* output bit equal to zero for these specific time stamps, whereas the invariant signal of defect 2 is permanently violated. Thus, with the inclusion of *SymbIST* setup 4 the two diagnostic measurements are now different and the defects can be distinguished.

It would be possible to distinguish these two specific defects by extracting two diagnostic measurements per defect, as explained in the previous Section 6.4.1, by considering separately the upper and the lower test limit. The first defect permanently violates the upper test limit, whereas the second defect permanently violates the lower test limit. Thus, their diagnostic measurement would be different.

Figure 6.11 shows a second transistor-level transient simulation for two short defects within the SUBDAC₁. The middle subplot shows the

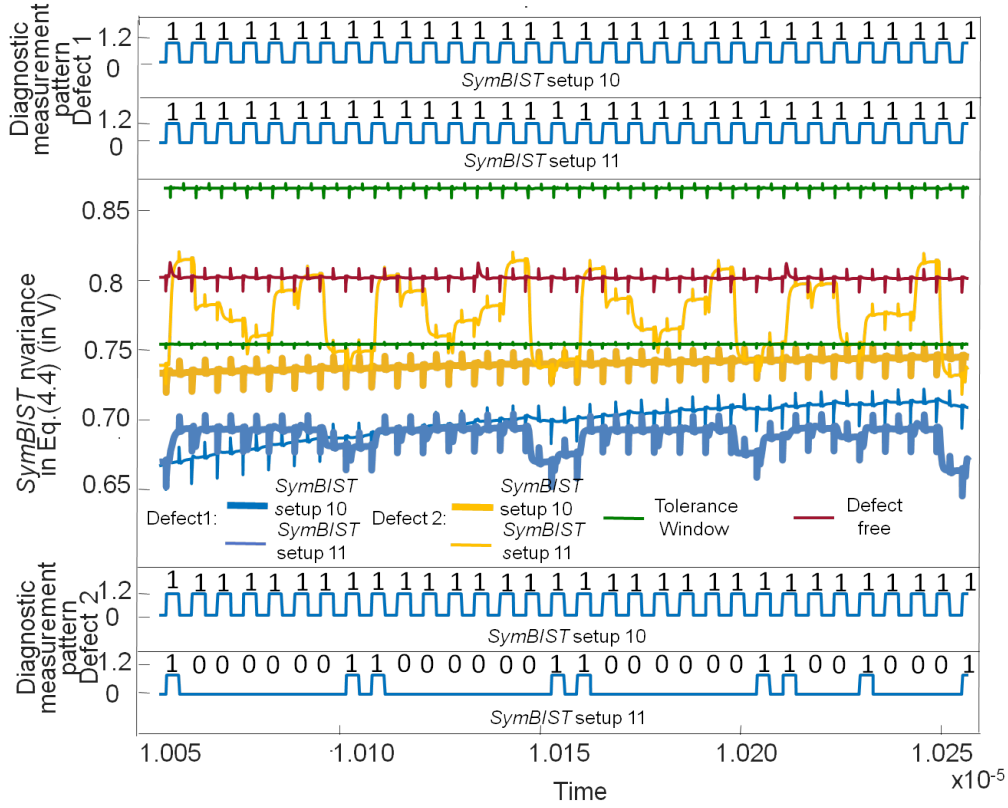


Figure 6.11: Transient simulation of *SymBIST* invariance in Equation (4.4) for two short defects within the SUBDAC, considering two different *SymBIST* setups.

invariant signal in Equation (4.4) for the defect-free and the two defective cases. For each defect we show the invariance for two different *SymBIST* setups 10 and 11, which differ only in the dynamic part of the test stimulus, i.e., the input DAC sequence. The top and bottom subplots show the diagnostic measurement pattern of each defect for these two setups. In *SymBIST* setup 10, for both defects the invariance violates the lower limit of the tolerance window, thus their diagnostic measurement pattern is the same, i.e., a vector of ones, and the two defects are inseparable. With *SymBIST* setup 11, the invariant signal for defect 2 now toggles across the lower limit, whereas the invariant signal for defect 1 is still permanently violated. Thus, the two diagnostic measurement patterns are now different and the defects are successfully distinguished.

Figure 6.12 shows the diagnosis results achieved with *SymBIST* following the strategy described in Section 6.4.1. For a given *SymBIST* setup, we show 3 bars corresponding to the percentage of uniquely diagnosed defects, i.e., ambiguity groups of size 1, the percentage of defects in ambiguity groups of size ≤ 5 , and the percentage of defects in ambiguity groups of size ≤ 10 . The inclusion of additional *SymBIST* setups stops when these three metrics saturate. As it can be seen, saturation occurs after using

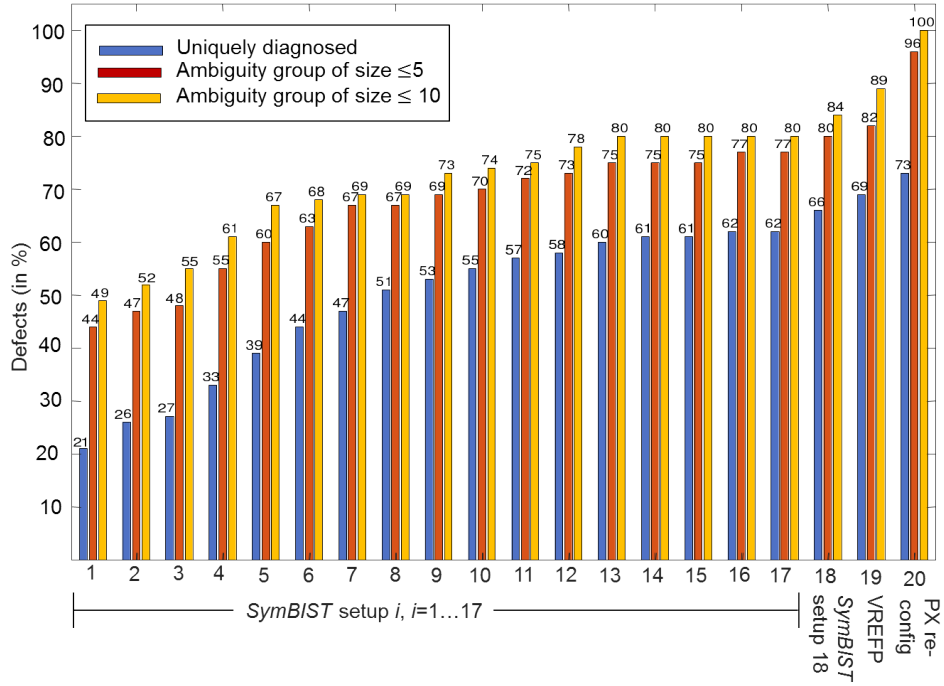


Figure 6.12: Percentage of defects uniquely diagnosed and in ambiguity groups of sizes ≤ 5 and ≤ 10 .

17 different *SymBIST* setups. Using only the first *SymBIST* setup, 21% of the defects are uniquely diagnosed, 44% are in an ambiguity group of size ≤ 5 , and 49% are in an ambiguity group of size ≤ 10 . After using 17 different *SymBIST* setups, these metrics increase to 62%, 77%, and 80%, respectively. Figure 6.13(a) sheds more light into the resultant ambiguity groups and their sizes. For example, 263 defects are uniquely diagnosed, there are $38/2 = 19$ ambiguity groups of size 2, 1 ambiguity group of size 3, etc.

Next, we aimed at further improving the diagnosis result in Figure 6.12 achieved with 17 *SymBIST* setups. We followed the steps presented in Section 6.4.1, by sequentially adding test setups. First we extracted two diagnostic measurement patterns per defect, considering separately the two tolerance window limits. This new *SymBIST* setup uses the same test stimulus and comparison window as *SymBIST* setup 1 and is numbered 18 in Figures 6.12 and 6.13(b). As it can be seen, it improves the diagnosis resolution. For example, the number of uniquely diagnosed defects increases to 277 and the largest ambiguity group has now size 21.

The second idea was to use the existing VREFP test pin. This test is numbered 19 in Figures 6.12 and 6.13(c). It further improves the diagnosis resolution since the number of uniquely diagnosed defects increases to 291 and the largest ambiguity group has now size 17.

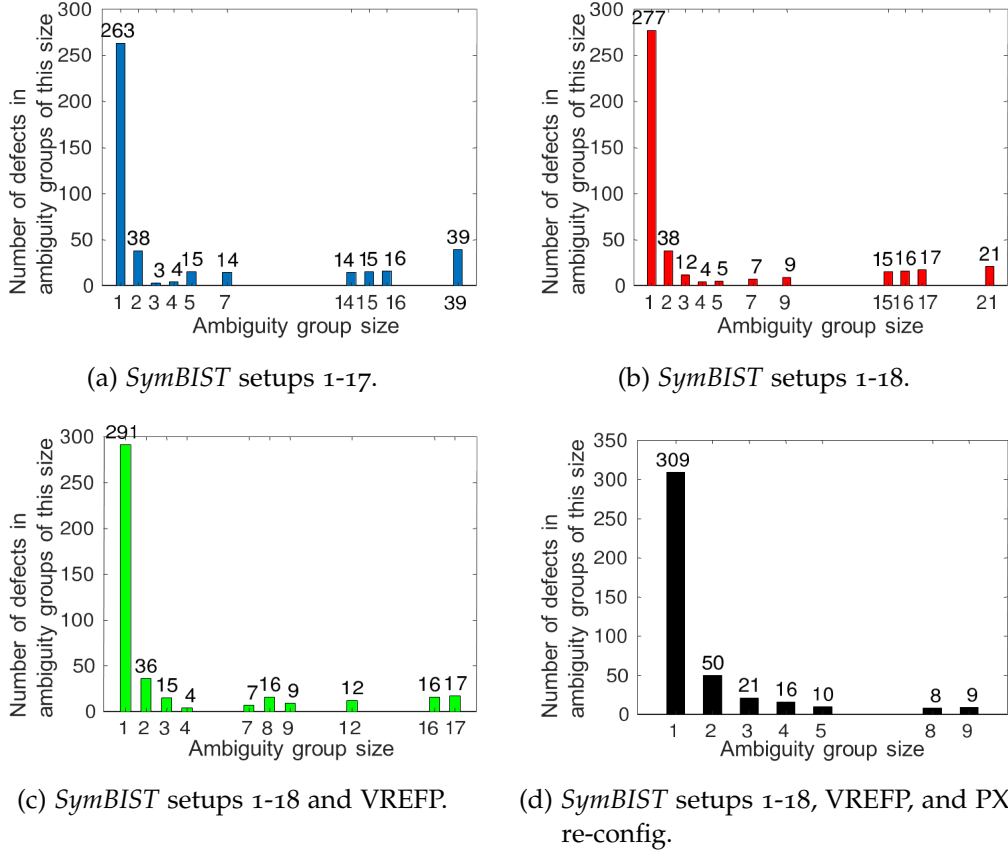


Figure 6.13: Gradual diagnosis resolution improvement.

Our third idea was to use internal re-configuration, by adding **PU** and **PD MOS** transistors. First, we observed that the two largest ambiguity groups of sizes 16 and 17 in Figure 6.13(c) concern defects inside the Bandgap. Thus, we applied topology modifications only in the Bandgap. More specifically, we inserted 13 **PU** and **PD** transistors enabling 13 re-configurations, where in each re-configuration only a single **PU** or **PD** transistor is activated. For each re-configuration, we read out the diagnostic measurements using *SymBIST* setup 1. Since the Bandgap provides biases for all blocks inside the **SAR ADC**, the effect of a defect in the Bandgap can propagate to the outputs of the checkers in a complex way. This test is named “PX re-config” and is numbered 20 in Figures 6.12 and 6.13(d). By adding this *SymBIST* variant, we were able to uniquely diagnose 73% of the defects, while the largest ambiguity group has size 9 and all but $8 + 9 = 17$ defects, i.e., 96% of the defects, are in ambiguity groups of size ≤ 5 , which overall is an excellent diagnosis resolution.

Finally, in total we used $L = 18 + 13 = 31$ *SymBIST* setups, thus maximum diagnosis time is $0.205 \cdot 31 = 6.35 \mu\text{s}$, plus the time of a **DC** measurement from the **VREFP** pin.

It should be noted that the percentages in Figure 6.12 are given in terms of detectable defects using nominal tolerance windows placed at

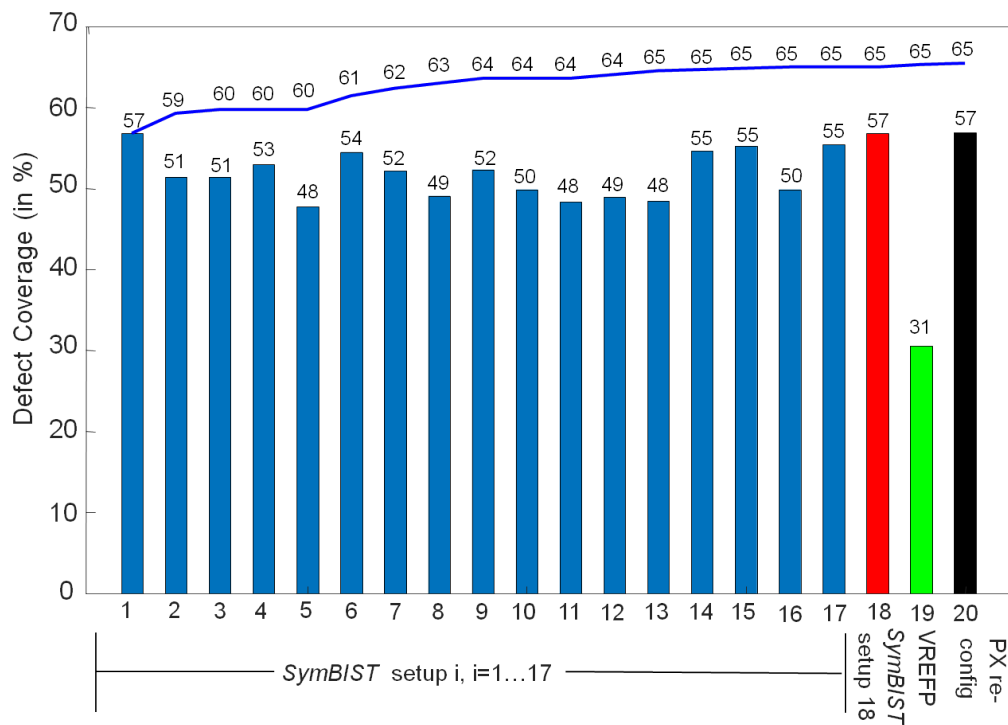


Figure 6.14: Absolute defect coverage per test setup and cumulative absolute defect coverage.

$5 \cdot \sigma$. The bars in Figure 6.14 represent the absolute defect coverage per test setup, whereas the blue line shows the cumulative defect coverage. The first *SymBIST* setup provides the best absolute defect coverage of 57%, while the cumulative defect coverage increases to 65% considering all 17 *SymBIST* setups. The cumulative defect coverage doesn't further improve by adding the three extra test setups, which are explicitly used for diagnosis purposes. Especially the topology modification approach achieves low defect coverage as essentially it only targets defects in the Bandgap and Reference Buffer and the reported defect coverage in Figure 6.14 is achieved from the *SymBIST* application. In practice we are interested in the *L-W* defect coverage that considers the likelihood of defect occurrence [7], which is shown in Section 6.2 to be over 86% for the complete A/M-S part of the SAR ADC IP using the first *SymBIST* setup. Thus, we are diagnosing over 86% of occurring defective cases considering defect likelihoods.

6.5 CONCLUSIONS

In this Chapter we demonstrated *SymBIST* on a real SAR ADC IP, for off-line defect-oriented testing, on-line testing, and analog fault diagnosis. The analog part of the *SymBIST* infrastructure is self-testable achieving *L-W*

defect coverage of over 94%. *SymBIST* achieves L-W defect coverage of around 87% for the complete analog and mixed analog-digital part of the SAR ADC IP. *SymBIST* is also successfully used for in-field concurrent error detection, detecting both transient errors and latent defects. Finally, *SymBIST* can also be applied for analog fault diagnosis resulting in high diagnosis resolution. More specifically, we achieved 73% uniquely detected defects, around 96% defects in ambiguity groups with size less than 5, while the largest ambiguity group has size of 9. In terms of test time, *SymBIST* proved to be very fast, offering a fast diagnosis cycle of 6.35 μs , whereas the maximum duration of the test in off-line test mode is equal to 0.205 μs . For on-line test, *SymBIST* offers real-time error detection with latency of 1 clock cycle.

CONCLUSIONS AND FUTURE WORK

7.1 CONCLUSIONS

In semiconductor manufacturing, MOS devices are scaling rapidly, which allows more and more components to be integrated into a given die area. Concurrently, the evolution of SoCs contributes to the direction of embedding an increasing number of ICs into the applications. Especially for the safety- and mission-critical applications, i.e., automotive, smart health-care, etc., all these advances pose a major challenge to guaranteeing robustness and reliability of the system. Moreover, nowadays Internet of Things (IoT), which relies on electronic systems, starts to play a significant role in such safety-critical applications. For example, in vehicle applications, the vehicle-to-vehicle communication will be implemented by IoT edge devices, and therefore, the functional safety of individual ICs must increase to prevent decrease in the system's functional safety.

The objective of this work was to develop a BIST approach specifically for A/M-S ICs that can be used to enhance the functional safety of an electronic system deployed in a safety-critical application. The proposed BIST approach, called *SymBIST*, was shown to achieve several objectives towards the functional safety goal. First, it can be used for defect-oriented test with high defect coverage towards post-manufacturing test quality improvement. Second, it can be used for on-line test concurrently with the operation towards detecting aging, latent defects, and single event upsets. Third, it can be used for fault diagnosis with high resolution and small diagnosis cycle towards yield learning and implementing corrective actions to avoid failure re-occurrence. In this thesis we demonstrated the above three use cases on a real industrial SAR ADC IP, yet *SymBIST* is virtually applicable to any A/M-S ICs .

7.2 CONTRIBUTIONS OF THE THESIS

More specifically, this thesis made the following contributions:

1. *SymBIST* is a generic BIST paradigm based on constructing or identifying inherent invariances and checking whether those invariances are satisfied. Invariances should hold true, i.e., stay within a tolerance window, only in error-free operation, while their violation indicates abnormal operation, i.e., the presence of a defect, aging, transient errors, etc.

2. We demonstrated the complete *SymBIST* test infrastructure for an industrial SAR ADC IP. The SAR ADC IP was re-designed to embed the *SymBIST* test infrastructure with no performance penalty and an area overhead of around 5%. We demonstrated how to efficiently extract invariances and the design of on-chip checkers for monitoring these invariances. We also demonstrated the design of on-chip digital test stimulus generators and minimal re-configurations required in the defect-oriented and diagnosis operation modes.
3. For the defect-oriented test use case, *SymBIST* offers a fast time which allows to perform an accelerated large-scale defect simulation to assess defect coverage with high confidence. For the SAR ADC IP test time is in the order of sub- μ s and we demonstrated likelihood-weighted defect coverage of over 86% which is considered very high for A/M-S ICs.
4. For the on-line test use case, *SymBIST* can check the invariances in real-time without interrupting the operation of the circuit and irrespective of the input. For the SAR ADC IP, we demonstrated that *SymBIST* is capable of detecting aging, latent defects, and transient errors.
5. For the diagnosis use case, the *SymBIST* response is used as a digital diagnostic measurement. The same test infrastructure is used but different *SymBIST* setups are used towards reducing defect ambiguity. We showed how *SymBIST* can be combined with topology modifications so as to improve diagnosis resolution. Overall, for the SAR ADC IP *SymBIST* achieved 73% correct diagnosis of defects, while over 96% of defects are in ambiguity groups of maximum size 5 and the largest ambiguity group has size 9. In addition, *SymBIST* offers a fast diagnosis cycle in the order of a few μ s.
6. *SymBIST* has the auxiliary benefit that it has a fully digital interface, thus in the context of a SoCs it can connect to any modern test access and control mechanism based on two external test pins.

7.3 FUTURE WORK

Based on the conducted research and the derived conclusions, a number of recommendations for future research are proposed.

The first perspective is to validate *SymBIST* with silicon results. *SymBIST* has been demonstrated so far at transistor-level. The next step will be to re-design the layout of the SAR ADC IP with the extra hardware of *SymBIST* and then fabricate a test chip. We can provoke failures in the test chip for example by burn-in test, by artificially switching-in defects, and by

injecting current pulses in internal nodes. The goal is to demonstrate in silicon the capability of *SymBIST* for detecting failures.

Future work will also concentrate on applying *SymBIST* to additional A/M-S ICs, including other ADC types and architectures, PLLs, and Time-to-Digital Converters (TDCs), etc. Moreover, future work should deal with case studies in other-technology nodes.

Our diagnosis results are encouraging, but they can be further improved, for example we can aim at improving the percentage of uniquely diagnosed defects and the size of ambiguity groups. In order to achieve this, more invariances and *SymBIST* setups can be constructed to further enrich the diagnostic measurement pattern and improve diagnosis resolution. Another aspect that needs to be studied is the effect of process variations in the diagnostic accuracy. *SymBIST* provides a robust digital diagnostic measurement pattern; however, in the presence of process variations a defect may be mapped to more than one possible diagnostic measurement patterns. To address this problem, one can envision employing a multi-class classifier to perform the diagnosis using *SymBIST* responses as features.

Part II

APPENDIX

BIBLIOGRAPHY

- [1] S. Sunter, K. Jurga, P. Dingenen, and R. Vanhooren, "Practical random sampling of potential defects for analog fault simulation," in *Proc. IEEE International Test Conference*, Paper 2.3, 2014 (cit. on pp. xv, 51–54).
- [2] H. F. Church and B. C. Roberts, "Failure mechanisms of electronic components," in *Fourth Annual Symposium on the Physics of Failure in Electronics*, 1965, pp. 156–178 (cit. on p. 3).
- [3] F. Fantini and C. Morandi, "Failure modes and mechanisms for VLSI ICs—a review," *IEE Proceedings G (Electronic Circuits and Systems)*, vol. 132, no. 3, 74–81(7), 1985 (cit. on p. 3).
- [4] Q. Huang, C. Fang, S. Mittal, and R. D. (Shawn) Blanton, "Towards smarter diagnosis: a learning-based diagnostic outcome previewer," *ACM Transactions on Design Automation of Electronic Systems*, vol. 25, no. 5, 2020 (cit. on pp. 5, 17).
- [5] G. Gielen, W. Dobbelaere, R. Vanhooren, A. Coyette, and B. Esen, "Design and test of analog circuits towards sub-ppm level," in *Proc. IEEE International Test Conference*, Paper 19.3, 2014 (cit. on pp. 5, 6).
- [6] H. Hashempour, J. Dohmen, B. Tasic, B. Kruseman, C. Hora, M. Beurden, and Y. Xing, "Test time reduction in analogue/mixed-signal devices by defect oriented testing: an industrial example," in *Proc. Design, Automation & Test in Europe Conference*, 2011 (cit. on p. 6).
- [7] S. Sunter, K. Jurga, and A. Laidler, "Using mixed-signal defect simulation to close the loop between design and test," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 12, pp. 2313–2322, 2016 (cit. on pp. 6, 20, 49–52, 65, 66, 76).
- [8] V. G. Gil, A. J. G. Arteaga, and G. Léger, "Assessing AMS-RF test quality by defect simulation," *IEEE Transactions on Device and Materials Reliability*, vol. 19, no. 1, pp. 55–63, 2019 (cit. on pp. 6, 20, 66).
- [9] J. Gomez, N. Kama, A. Coyette, R. Vanhooren, W. Dobbelaere, and G. Gielen, "Pinhole latent defect modeling and simulation for defect-oriented analog/mixed-signal testing," in *Proc. IEEE VLSI Test Symposium*, 2020 (cit. on pp. 6, 68).

- [10] F. Su and P. Goteti, "Improving analog functional safety using data-driven anomaly detection," in *Proc. IEEE International Test Conference*, Paper 13.3, 2018 (cit. on p. 7).
- [11] K. Huang, H.-G. Stratigopoulos, S. Mir, C. Hora, Y. Xing, and B. Kruseman, "Diagnosis of local spot defects in analog circuits," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 10, pp. 2701–2712, 2012 (cit. on pp. 7, 19).
- [12] A. Vassighi and M. Sachdev, "Burn-in as a reliability screening test. in: Thermal and Power Management of Integrated Circuits. Series on Integrated Circuits and Systems," in: Springer, Boston, MA, 2006 (cit. on p. 7).
- [13] W. Dobbelaere, R. Vanhooren, W. D. Man, K. Matthijs, A. Coyette, B. Esen, and G. G. E. Gielen, "Analog fault coverage improvement using final-test dynamic part average testing," in *Proc. IEEE International Test Conference*, 2016 (cit. on p. 7).
- [14] L.-C. Wang, S. Siatkowski, C. Shan, M. Nero, N. Sumikawa, and L. Winemberg, "Some considerations on choosing an outlier method for automotive product lines," in *Proc. IEEE International Test Conference*, Paper 14.3, 2017 (cit. on p. 7).
- [15] H.-G. Stratigopoulos and C. Streitwieser, "Adaptive test with test escape estimation for mixed-signal ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 10, pp. 2125–2138, 2018 (cit. on p. 7).
- [16] A. Coyette, B. Esen, W. Dobbelaere, R. Vanhooren, and G. Gielen, "Automatic generation of test infrastructures for analog integrated circuits by controllability and observability co-optimization," *Integration, the VLSI Journal*, vol. 55, pp. 393–400, 2016 (cit. on pp. 9, 71).
- [17] P. Malcovati, F. Maloberti, C. Fiocchi, and M. Pruzzi, "Curvature-compensated BiCMOS bandgap with 1 – V supply voltage," *Solid-State Circuits, IEEE Journal of*, vol. 36, pp. 1076–1081, Aug. 2001 (cit. on p. 10).
- [18] K. Arabi and B. Kaminska, "Testing analog and mixed-signal integrated circuits using oscillation-test method," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 7, pp. 745–753, 1997 (cit. on p. 10).
- [19] K. Arabi and B. Kaminska, "Design for testability of embedded integrated operational amplifiers," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 4, pp. 573–81, 1998 (cit. on p. 10).

- [20] K. Arabi and B. Kaminska, "Oscillation-test methodology for low-cost testing of active analog filters," *IEEE Transactions on Instrumentation and Measurement*, vol. 48, no. 4, pp. 798–806, 1999 (cit. on p. 10).
- [21] M. S. Zarnik, F. Novak, and S. Macek, "Design of oscillation-based test structures for active RC filters," *IEE Proceedings-Circuits, Devices, and Systems*, vol. 147, no. 5, pp. 297–302, 2000 (cit. on p. 10).
- [22] G. Huertas, D. Vázquez, E. J. Peralías, A. Rueda, and J. L. Huertas, "Practical oscillation-based test of integrated filters," *IEEE Design & Test of Computers*, vol. 19, no. 6, pp. 64–72, 2002 (cit. on p. 10).
- [23] J. Roy and J. A. Abraham, "A comprehensive signature analysis scheme for oscillation-test," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 10, pp. 1409–1423, 2003 (cit. on p. 10).
- [24] K. Suenaga, E. Isern, R. Picos, S. Bota, M. Roca, and E. García-Moreno, "Application of predictive oscillation-based test to a CMOS opAmp," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 8, pp. 2076–2082, 2010 (cit. on p. 10).
- [25] S. Callegari, F. Pareschi, G. Setti, and M. Soma, "Complex oscillation-based test and its application to analog filters," *IEEE Transactions on Circuits and systems I: Regular Papers*, vol. 57, no. 5, pp. 956–969, 2010 (cit. on p. 10).
- [26] G. Huertas, D. Vázquez, A. Rueda, and J. L. Huertas, *Oscillation-Based Test in Mixed-Signal Circuits*. Springer, 2006 (cit. on p. 10).
- [27] K. Arabi and B. Kaminska, "Efficient and accurate testing of analog-to-digital converters using oscillation-test method," in *Proc. European Design and Test Conference*, 1997, pp. 348–352 (cit. on p. 10).
- [28] K. Arabi and B. Kaminska, "Oscillation built-in self test (OBIST) scheme for functional and structural testing of analog and mixed-signal integrated circuits," in *Proc. IEEE International Test Conference*, 1997, pp. 786–795 (cit. on p. 10).
- [29] G. Huertas, D. Vazquez, E. J. Peralias, A. Rueda, and J. L. Huertas, "Oscillation-based test in oversampling $\Sigma\Delta$ modulators," *Microelectronics Journal*, vol. 33, pp. 799–806, 2002 (cit. on p. 10).
- [30] A. Goyal, M. Swaminathan, and A. Chatterjee, "Low-cost specification based testing of RF amplifier circuits using oscillation principles," *Journal of Electronic Testing: Theory and Applications*, vol. 26, no. 1, pp. 13–24, 2010 (cit. on p. 10).

- [31] V. Beroulle, Y. Bertrand, L. Latorre, and P. Nouet, "Evaluation of the oscillation-based test methodology for micro-electro-mechanical systems," in *Proceedings 20th IEEE VLSI Test Symposium (VTS 2002)*, 2002, pp. 439–444 (cit. on p. 10).
- [32] H.-G. Stratigopoulos and S. Mir, "Analog test metrics estimates with PPM accuracy," in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 2010, pp. 241–247 (cit. on p. 10).
- [33] G. Huertas, D. Vázquez, E. J. Peralías, A. Rueda, and J. L. Huertas, "Testing mixed-signal cores: a practical oscillation-based test in an analog macrocell," *IEEE Design & Test of Computers*, vol. 19, no. 6, pp. 73–82, 2002 (cit. on p. 11).
- [34] K. Arabi and B. Kaminska, "Parametric and catastrophic fault coverage of analog circuits in oscillation-test methodology," in *Proc. IEEE VLSI Test Symposium*, 1997, pp. 166–171 (cit. on p. 11).
- [35] A. K. Lu, G. W. Roberts, and D. A. Johns, "A high quality analog oscillator using oversampling D/A converters technique," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 41, no. 7, pp. 437–444, 1994 (cit. on p. 11).
- [36] M. M. Elsayed and E. Sánchez-Sinencio, "A low THD, low power, high output-swing time-mode-based tunable oscillator via digital harmonic-cancellation technique," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 5, pp. 1061–1071, 2010 (cit. on p. 11).
- [37] B. K. Vasan, S. K. Sudani, D. J. Chen, and R. L. Geiger, "Low-distortion sine wave generation using a novel harmonic cancellation technique," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 5, pp. 1122–1134, 2013 (cit. on p. 11).
- [38] H. Malloug, M. J. Barragan, and S. Mir, "Practical harmonic cancellation techniques for the on-chip implementation of sinusoidal signal generators for mixed-signal BIST applications," *Journal of Electronic Testing: Theory and Applications*, vol. 34, no. 3, pp. 263–279, 2018 (cit. on pp. 11, 12).
- [39] B. Dufort and G. W. Roberts, "On-chip analog signal generation for mixed-signal built-in self-test," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 3, pp. 318–30, 1999 (cit. on p. 12).
- [40] M. F. Toner and G. W. Roberts, "A BIST scheme for an SNR test of a sigma-delta ADC," in *Proc. IEEE International Test Conference*, 1993, pp. 805–814 (cit. on p. 12).
- [41] M. F. Toner and G. W. Roberts, "A BIST scheme for a SNR, gain tracking, and frequency response test of a sigma-delta ADC," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 1, pp. 1–15, 1995 (cit. on p. 12).

- [42] H.-C. Hong and S.-C. Liang, "A decorrelating design-for-digital-testability scheme for $\Sigma\Delta$ modulators," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 1, pp. 60–73, 2009 (cit. on p. 12).
- [43] M. Barragan, R. Alhakim, H.-G. Stratigopoulos, M. Dubois, S. Mir, H. L. Gall, N. Bhargava, and A. Bal, "A fully-digital BIST wrapper based on ternary test stimuli for the dynamic test of a 40nm CMOS 18-bit stereo audio $\Sigma\Delta$ ADC," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 11, pp. 1876–1888, 2016 (cit. on p. 12).
- [44] H. Chauhan, Y. Choi, M. Onabajo, I.-S. Jung, and Y.-B. Kim, "Accurate and efficient on-chip spectral analysis for built-in testing and calibration approaches," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 3, pp. 49–506, 2014 (cit. on p. 12).
- [45] H. Mattes, S. Sattler, and C. Dworski, "Controlled sine wave fitting for ADC test," in *Proc. IEEE International Test Conference*, 2004, pp. 963–971 (cit. on p. 12).
- [46] F. Azais, S. Bernard, Y. Bertrand, X. Michel, and M. Renovell, "A low-cost adaptive ramp generator for analog BIST applications," in *Proc. IEEE VLSI Test Symposium*, 2001, pp. 266–271 (cit. on p. 12).
- [47] B. Provost and E. Sánchez-Sinencio, "On-chip ramp generators for mixed-signal BIST and ADC self-test," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 2, pp. 263–273, 2003 (cit. on p. 12).
- [48] R. Holcer, L. Michaeli, and J. Saliga, "DNL ADC testing by the exponential shaped voltage," *IEEE Transactions on Instrumentation and Measurement*, vol. 52, no. 3, pp. 946–949, 2003 (cit. on p. 12).
- [49] L. Jin, K. Parthasarathy, T. Kuyel, D. Chen, and R. L. Geiger, "Accurate testing of analog-to-digital converters using low linearity signals with stimulus error identification and removal," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 3, pp. 1188–1199, 2005 (cit. on p. 12).
- [50] F. Azais, S. Bernard, Y. Bertrand, and M. Renovell, "Optimizing sinusoidal histogram test for low cost ADC BIST," *Journal of Electronic Testing: Theory and Applications*, vol. 17, no. 3-4, pp. 255–266, 2001 (cit. on p. 12).
- [51] S. Goyal and A. Chatterjee, "Linearity testing of A/D converters using selective code measurement," *Journal of Electronic Testing: Theory and Applications*, vol. 24, no. 6, pp. 567–576, 2008 (cit. on p. 12).

- [52] H. Xing, H. Jiang, D. Chen, and R. L. Geiger, "High-resolution ADC linearity testing using a fully digital-compatible BIST strategy," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 8, pp. 2697–2705, 2009 (cit. on p. 12).
- [53] A. Laraba, H.-G. Stratigopoulos, S. Mir, and H. Naudet, "Exploiting pipeline ADC properties for a reduced-code linearity test technique," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 10, pp. 2391–2400, 2015 (cit. on p. 12).
- [54] G. Renaud, M. J. Barragan, A. Laraba, H.-G. Stratigopoulos, S. Mir, H. L. Gall, and H. Naudet, "A 65nm CMOS ramp generator design and its application towards a BIST implementation of the reduced-code static linearity test technique for pipeline ADCs," *Journal of Electronic Testing: Theory and Applications*, vol. 32, no. 4, pp. 407–421, 2016 (cit. on p. 12).
- [55] R. S. Feitoza, M. J. Barragan, S. Mir, and D. Dzahini, "Reduced-code static linearity test of SAR ADCs using a built-in incremental $\Sigma\Delta$ converter," in *Proc. IEEE International Symposium on On-Line Testing and Robust System Design*, 2018, pp. 29–34 (cit. on p. 12).
- [56] T. Chen, X. Jin, R. L. Geiger, and D. Chen, "USER-SMILE: ultrafast stimulus error removal and segmented model identification of linearity errors for ADC built-in self-test," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 7, pp. 2059–2069, 2018 (cit. on p. 12).
- [57] G. Léger and A. Rueda, "Low-cost digital detection of parametric faults in cascaded $\Sigma\Delta$ modulators," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 7, pp. 1326–1338, 2009 (cit. on p. 13).
- [58] A. Gines and G. Léger, " $\Sigma\Delta$ testability for pipeline A/D converters," in *Proc. Design, Automation and Test in Europe Conference*, 2014 (cit. on pp. 13, 49).
- [59] M. Jarwala, L. Duy, and M. S. Heutmaker, "End-to-end test strategy for wireless systems," in *Proc. IEEE International Test Conference*, 1995, pp. 940–946 (cit. on p. 13).
- [60] M. S. Heutmaker and D. K. Le, "An architecture for self-test of a wireless communication system," *IEEE Communications Magazine*, vol. 37, no. 6, pp. 98–102, 1999 (cit. on p. 13).
- [61] D. Lupea, U. Pursche, and H.-J. Jentschel, "RF-BIST: loopback spectral signature analysis," in *Proc. Design, Automation, & Test in Europe Conference*, 2003, pp. 478–483 (cit. on p. 13).
- [62] S. Ozev and C. Olgaard, "Wafer-level RF test and DfT for VCO modulating transceiver architectures," in *Proc. IEEE VLSI Test Symposium*, 2004, pp. 217–222 (cit. on p. 13).

- [63] J.-S. Yoon and W. R. Eisenstadt, "Embedded loopback test for RF ICs," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 5, pp. 1715–1720, 2005 (cit. on p. 13).
- [64] A. Valdes-Garcia, J. Silva-Martinez, and E. Sanchez-Sinencio, "On-chip testing techniques for RF wireless transceivers," *IEEE Design & Test of Computers*, vol. 23, no. 4, pp. 268–277, 2006 (cit. on p. 13).
- [65] E. S. Erdogan and S. Ozev, "Detailed characterization of transceiver parameters through loop-back-based BIST," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 6, pp. 901–911, 2010 (cit. on p. 13).
- [66] J. J. Dabrowski and R. M. Ramzan, "Built-in loopback test for IC RF transceivers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 6, pp. 933–946, 2010 (cit. on p. 13).
- [67] M. Cimino, H. Lapuyade, Y. Deval, T. Taris, and J.-B. Bégueret, "Design of a 0.9V 2.45 GHz self-testable and reliability-enhanced CMOS LNA," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 5, pp. 1187–1194, 2008 (cit. on p. 13).
- [68] L. Abdallah, H.-G. Stratigopoulos, S. Mir, and C. Kelma, "RF front-end test using built-in sensors," *IEEE Design & Test of Computers*, vol. 28, no. 6, pp. 76–84, 2011 (cit. on p. 13).
- [69] A. Gopalan, M. Margala, and P. R. Mukund, "A current based self-test methodology for RF front-end circuits," *Microelectronics Journal*, vol. 36, no. 12, pp. 1091–1102, 2005 (cit. on p. 13).
- [70] A. Valdes-Garcia, R. Venkatasubramanian, J. Silva-Martinez, and E. Sanchez-Sinencio, "A broadband CMOS amplitude detector for on-chip RF measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 7, pp. 1470–1477, 2008 (cit. on p. 13).
- [71] Y.-C. Huang, H.-H. Hsieh, and L.-H. Lu, "A low-noise amplifier with integrated current and power sensors for RF BIST applications," in *IEEE VLSI Test Symposium*, 2007, pp. 401–408 (cit. on p. 13).
- [72] S. S. Akbay and A. Chatterjee, "Built-in test of RF components using mapped feature extraction sensors," in *IEEE VLSI Test Symposium*, 2005, pp. 243–248 (cit. on p. 13).
- [73] S. Bhattacharya and A. Chatterjee, "A DFT approach for testing embedded systems using DC sensors," *IEEE Design & Test of Computers*, vol. 23, no. 6, pp. 464–475, 2006 (cit. on p. 13).
- [74] P. F. D. Mota and J. M. D. Silva, "A true power detector for RF PA built-in calibration and testing," in *Proc. Design, Automation, & Test in Europe Conference*, 2011, pp. 1–6 (cit. on p. 13).

- [75] C. Zhang, R. Gharpurey, and J. A. Abraham, "Built-in self test of RF subsystems with integrated sensors," *Journal of Electronic Testing: Theory and Applications*, vol. 28, no. 5, pp. 557–569, 2012 (cit. on p. 13).
- [76] A. Zjajo, M. J. B. Asian, and J. P. de Gyvez, "BIST method for die-level process parameter variation monitoring in analog/mixed-signal integrated circuits," in *Proc. Design, Automation & Test in Europe Conference*, 2007, pp. 1301–1306 (cit. on p. 13).
- [77] L. Abdallah, H.-G. Stratigopoulos, S. Mir, and C. Kelma, "Experiences with non-intrusive sensors for RF built-in test," in *Proc. IEEE International Test Conference*, Paper 17.1, 2012 (cit. on p. 13).
- [78] J. Altet, A. Rubio, E. Schaub, S. Dilhaire, and W. Claeys, "Thermal coupling in integrated circuits: application to thermal testing," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 1, pp. 81–91, 2001 (cit. on p. 13).
- [79] J. Altet, A. Rubio, J. L. Rosselló, and J. Segura, "Structural RFIC device testing through built-in thermal monitoring," *IEEE Communications Magazine*, vol. 41, no. 9, pp. 98–104, 2003 (cit. on p. 13).
- [80] M. Onabajó, J. Altet, E. Aldrete-Vidrio, D. Mateo, and J. Silva-Martinez, "Electrothermal design procedure to observe rf circuit power and linearity characteristics with a homodyne differential temperature sensor," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, no. 99, pp. 1–1, 2011 (cit. on p. 13).
- [81] J. Altet, D. Mateo, D. Gómez, X. Perpinyà, M. Vallvehi, and X. Jordà, "DC temperature measurements for power gain monitoring in RF power amplifiers," in *Proc. IEEE International Test Conference*, Paper 17.3, 2012 (cit. on p. 13).
- [82] R. Voorakaranam, S. Cherubal, and A. Chatterjee, "A signature test framework for rapid production testing of RF circuits," in *Proc. Design, Automation and Test in Europe Conference*, 2002, pp. 186–191 (cit. on p. 13).
- [83] S. S. Akbay, A. Halder, A. Chatterjee, and D. Keezer, "Low-cost test of embedded RF/Analog/Mixed-signal circuits in SOPs," *IEEE Transactions on Advanced Packaging*, vol. 27, no. 2, pp. 352–363, 2004 (cit. on p. 13).
- [84] S. Cherubal, R. Voorakaranam, A. Chatterjee, J. McLaughlin, J. L. Smith, and D. M. Majernik, "Concurrent RF test using optimized modulated RF stimuli," in *IEEE International Conference on VLSI Design*, 2004, pp. 1017–1022 (cit. on p. 13).

- [85] S. S. Akbay, J. L. Torres, J. M. Rumer, A. Chatterjee, and J. Amtsfield, "Alternate test of RF front ends with IP constraints: frequency domain test generation and validation," in *Proc. IEEE International Test Conference*, 2006, pp. 4.4.1–4.4.10 (cit. on p. 13).
- [86] S. Ellouz, P. Gamand, C. Kelma, B. Vandewiele, and B. Allard, "Combining internal probing with artificial neural networks for optimal RFIC testing," in *Proc. IEEE International Test Conference*, 2006, pp. 4.3.1–4.3.9 (cit. on p. 13).
- [87] H.-G. Stratigopoulos and Y. Makris, "Error moderation in low-cost machine-learning-based Analog/RF testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 2, pp. 339–351, 2008 (cit. on p. 13).
- [88] H.-G. Stratigopoulos, S. Mir, E. Acar, and S. Ozev, "Defect filter for alternate RF test," in *Proc. IEEE European Test Symposium*, 2009, pp. 101–106 (cit. on p. 13).
- [89] M. J. Barragán, R. Fiorelli, G. Léger, A. Rueda, and J. L. Huertas, "Alternate test of LNAs through ensemble learning of on-chip digital envelope signals," *Journal of Electronic Testing: Theory and Applications*, vol. 27, no. 3, pp. 277–288, 2011 (cit. on p. 13).
- [90] H.-G. Stratigopoulos and S. Mir, "Adaptive alternate analog test," *IEEE Design & Test of Computers*, vol. 29, no. 4, pp. 71–79, 2012 (cit. on p. 13).
- [91] H. Ayari, F. Azais, S. Bernard, M. Compte, V. Kerzerho, O. Potin, and M. Renovell, "Making predictive analog/RF alternate test strategy independent of training set size," in *Proc. IEEE International Test Conference*, Paper 10.1, 2012 (cit. on p. 13).
- [92] M. J. and G. Leger, "Efficient selection of signatures for analog/RF alternate test," in *Proc. IEEE European Test Symposium*, 2013 (cit. on p. 13).
- [93] L. Abdallah, H.-G. Stratigopoulos, S. Mir, and J. Altet, "Defect-oriented non-intrusive RF test using on-chip temperature sensors," in *Proc. IEEE VLSI Test Symposium*, 2013 (cit. on p. 14).
- [94] S. Sunter and A. Roy, "On-chip digital jitter measurement, from megahertz to gigahertz," *IEEE Design & Test of Computers*, vol. 21, no. 4, pp. 314–321, 2004 (cit. on p. 14).
- [95] S. Sunter and A. Roy, "BIST for phase-locked loops in digital applications," in *Proc. IEEE International Test Conference*, 1999, pp. 532–540 (cit. on p. 14).

- [96] P. Dudek, S. Szczepanski, and J. Hatfield, "A high resolution CMOS time-to-digital converter utilizing a Vernier delay line," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 2, pp. 240–247, 2000 (cit. on p. 14).
- [97] A. Chan and G. Roberts, "A synthesizable, fast and high-resolution timing measurement device using a component-invariant Vernier delay line," in *Proc. IEEE International Test Conference*, 2001, pp. 858–867 (cit. on p. 14).
- [98] S. Tabatabaei and A. Ivanov, "Embedded timing analysis: a SoC infrastructure," *IEEE Design & Test of Computers*, vol. 19, no. 3, pp. 22–34, 2002 (cit. on p. 14).
- [99] J.-L. Huang, J.-J. Huang, and Y.-S. Liu, "A low-cost jitter measurement technique for bist applications," *Journal of Electronic Testing: Theory and Applications*, vol. 22, no. 3, pp. 219–228, 2006 (cit. on p. 14).
- [100] H. Le Gall, R. Alhakim, M. Valka, S. Mir, H.-G. Stratigopoulos, and E. Simeu, "High frequency jitter estimator for SoCs," in *2015 20th IEEE European Test Symposium (ETS)*, 2015, pp. 1–6 (cit. on p. 14).
- [101] B. R. Veillette and G. Roberts, "On-chip measurement of the jitter transfer function of charge-pump phase-locked loops," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, pp. 483–491, 1998 (cit. on p. 14).
- [102] J. Kim, "On-chip measurement of jitter transfer and supply sensitivity of PLL/DLLs," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 6, pp. 449–453, 2009 (cit. on p. 14).
- [103] A. Asquini, F. Badets, S. Mir, J.-L. Carbonero, and L. Bouzaida, "PFD output monitoring for RF PLL BIST," in *Proc. IEEE International Mixed-Signals, Sensors, and Systems Test Workshop*, 2008 (cit. on p. 14).
- [104] S.-W. Hsiao, N. Tzou, and A. Chatterjee, "A programmable BIST design for PLL static phase offset estimation and clock duty cycle detection," in *Proc. IEEE VLSI Test Symposium*, 2013 (cit. on p. 14).
- [105] S.-W. Hsiao, X. Wang, and A. Chatterjee, "Analog sensor based testing of phase-locked loop dynamic performance parameters," in *Proc. IEEE Asian Test Symposium*, 2013, pp. 50–55 (cit. on p. 14).
- [106] F. Azais, Y. Bertrand, M. Renovell, A. Ivanov, and S. Tabatabaei, "An all-digital dft scheme for testing catastrophic faults in plls," *IEEE Design Test of Computers*, vol. 20, no. 1, pp. 60–67, 2003 (cit. on p. 15).

- [107] S. Kim and M. Soma, "An all-digital built-in self-test for high-speed phase-locked loops," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 48, no. 2, pp. 141–150, 2001 (cit. on p. 15).
- [108] M. Ince, E. Yilmaz, W. Fu, J. Park, K. Nagaraj, L. Winemberg, and S. Ozev, "Digital built-in self-test for phased locked loops to enable fault detection," in *2019 IEEE European Test Symposium (ETS)*, 2019, pp. 1–6 (cit. on p. 15).
- [109] J. L. Huertas, A. Rueda, and D. Vasquez, "Testable switched-capacitor filters," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 7, pp. 719–724, 1993 (cit. on p. 15).
- [110] A. Chatterjee, "Concurrent error detection and fault-tolerance in linear analog circuits using continuous checksums," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 1, no. 2, pp. 138–150, 1993 (cit. on p. 15).
- [111] H.-G. D. Stratigopoulos and Y. Makris, "Concurrent detection of erroneous responses in linear analog circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 5, pp. 878–891, 2006 (cit. on p. 15).
- [112] N. J. Stessman, B. Vinnakota, and R. Harjani, "System-level design for test of fully differential analog circuits," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 10, pp. 1526–1534, 1996 (cit. on pp. 15, 25).
- [113] V. Kolarik, S. Mir, M. Lubaszewski, and B. Courtois, "Analog checkers with absolute and relative tolerances," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 5, pp. 607–612, 1995 (cit. on pp. 15, 16).
- [114] H.-G. D. Stratigopoulos and Y. Makris, "An adaptive checker for the fully differential analog code," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 6, pp. 1421–1429, 2006 (cit. on pp. 15, 16).
- [115] H.-G. Stratigopoulos, S. Mir, and A. Bounceur, "Evaluation of analog/RF test measurements at the design stage," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 4, pp. 582–590, 2009 (cit. on p. 16).
- [116] A. Bounceur, S. Mir, and H.-G. Stratigopoulos, "Estimation of analog parametric test metrics using copulas," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 9, pp. 1400–1410, 2011 (cit. on p. 16).
- [117] H.-G. Stratigopoulos, "Test metrics model for analog test development," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 7, pp. 1116–1128, 2012 (cit. on p. 16).

- [118] H.-G. Stratigopoulos, P. Faubet, Y. Courant, and F. Mohamed, "Multidimensional analog test metrics estimation using extreme value theory and statistical blockade," in *Proc. Design Automation Conference*, 2013 (cit. on p. 16).
- [119] H.-G. Stratigopoulos and S. Sunter, "Fast Monte Carlo-based estimation of analog parametric test metrics," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1977–1990, 2014 (cit. on p. 16).
- [120] M. J. Barragan, H.-G. Stratigopoulos, S. Mir, H. L. Gall, N. Bhargava, and A. Bal, "Practical simulation flow for evaluating analog/mixed-signal test techniques," *IEEE Design & Test*, vol. 33, no. 6, pp. 46–54, 2016 (cit. on p. 16).
- [121] M. Andraud, H.-G. Stratigopoulos, and E. Simeu, "One-shot non-intrusive calibration against process variations for analog/RF circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 11, pp. 2022–2035, 2016 (cit. on p. 16).
- [122] A. Antonopoulos, G. Volanis, Y. Lu, and Y. Makris, "Post-production calibration of analog/RF ICs: recent developments and a fully integrated solution," in *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*, 2019, pp. 77–80 (cit. on p. 16).
- [123] C. Maxey, G. Creech, S. Raman, J. Rockway, K. Groves, T. Quach, L. Orlando, and A. Mattamana, "Mixed-signal SoCs with in situ self-healing circuitry," *IEEE Design & Test of Computers*, vol. 29, no. 6, pp. 27–39, 2012 (cit. on p. 17).
- [124] S. Bowers, K. Sengupta, B. Parker, and A. Hajimiri, "Integrated self-healing for mm-wave power amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 3, pp. 352–363, 2013 (cit. on p. 17).
- [125] S. Lee, C. Shi, J. Wang, A. Sanabria, H. Osman, J. Hu, and E. Sánchez-Sinencio, "A built-in self-test and in situ analog circuit optimization platform," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 10, pp. 3445–3458, 2018 (cit. on p. 17).
- [126] S. Holst and H. Wunderlich, "Adaptive debug and diagnosis without fault dictionaries," in *IEEE European Test Symposium*, 2007, pp. 7–12 (cit. on p. 17).
- [127] B. Benware, C. Schuermyer, M. Sharma, and T. Herrmann, "Determining a failure root cause distribution from a population of layout-aware scan diagnosis results," *IEEE Design & Test of Computers*, vol. 29, no. 1, pp. 8–18, 2012 (cit. on p. 17).

- [128] J. Tikkanen, S. Siatkowski, N. Sumikawa, L.-C. Wang, and M. S. Abadir, "Yield optimization using advanced statistical correlation methods," in *Proc. IEEE International Test Conference*, 2014 (cit. on p. 17).
- [129] P. M. et al., "Cell-aware diagnosis: defective inmates exposed in their cells," in *IEEE European Test Symposium*, 2016 (cit. on p. 17).
- [130] S. Mhamdi, P. Girard, A. Virazel, A. Bosio, E. Faehn, and A. Ladhari, "Cell-aware defect diagnosis of customer returns based on supervised learning," *IEEE Transactions on Device and Materials Reliability*, vol. 20, no. 2, pp. 329–340, 2020 (cit. on p. 17).
- [131] E. S. Erdogan, S. Ozev, and P. Cauvet, "Diagnosis of assembly failures for system-in-package RF tuners," in *IEEE International Symposium on Circuits and Systems*, 2008, pp. 2286–2289 (cit. on p. 17).
- [132] N. Sen and R. Saeks, "Fault diagnosis for linear systems via multifrequency measurements," *IEEE Transactions on Circuits and Systems*, vol. 26, no. 7, pp. 457–465, 1979 (cit. on p. 18).
- [133] H. Dai and M. Souders, "Time-domain testing strategies and fault diagnosis for analog systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 39, no. 1, pp. 157–162, 1990 (cit. on p. 18).
- [134] M. Slamani and B. Kaminska, "Analog circuit fault diagnosis based on sensitivity computation and functional testing," *IEEE Design & Test of Computers*, vol. 9, no. 1, pp. 30–39, 1992 (cit. on p. 18).
- [135] S. Chakrabarti, S. Cherubal, and A. Chatterjee, "Fault diagnosis for mixed-signal electronic systems," in *Proc. IEEE Aerospace Conference*, 1999, pp. 169–179 (cit. on p. 18).
- [136] K. Huang, H.-G. Stratigopoulos, and S. Mir, "Fault diagnosis of analog circuits based on machine learning," in *Proc. Design, Automation & Test in Europe Conference*, 2010, pp. 1761–1766 (cit. on p. 18).
- [137] E. F. Cota, M. Negreiros, L. Carro, and M. Lubaszewski, "A new adaptive analog test and diagnosis system," *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 2, pp. 223–227, 2000 (cit. on pp. 18, 19).
- [138] S. Kook, A. Banerjee, and A. Chatterjee, "Dynamic specification testing and diagnosis of high-precision $\Sigma\Delta$ ADCs," *IEEE Design & Test of Computers*, vol. 30, no. 4, pp. 36–48, 2013 (cit. on pp. 18, 19).

- [139] F. Ferguson and J. Shen, "A CMOS fault extractor for inductive fault analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 7, no. 11, pp. 1181–1194, 1988 (cit. on p. 19).
- [140] R. Spina and S. Upadhyaya, "Linear circuit fault diagnosis using neuromorphic analyzers," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 44, no. 3, pp. 188–196, 1997 (cit. on p. 19).
- [141] M. Aminian and F. Aminian, "A modular fault-diagnosis system for analog electronic circuits using neural networks with wavelet transform as a preprocessor," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 5, pp. 1546–1554, 2007 (cit. on p. 19).
- [142] S. S. Somayajula, E. Sanchez-Sinencio, and J. P. de Gyvez, "Analog fault diagnosis based on ramping power supply current signature clusters," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 43, no. 10, pp. 703–712, 1996 (cit. on pp. 19, 20).
- [143] Y. Maidon, B. W. Jarvis, N. Dutton, and S. Lesage, "Diagnosis of multifaults in analogue circuits using multilayer perceptrons," *IEE Proceedings - Circuits, Devices & Systems*, vol. 144, no. 3, pp. 149–154, 1997 (cit. on pp. 19, 20).
- [144] J. A. Starzyk, D. Liu, Z. -H. Liu, D. E. Nelson, and J. O. Rutkowski, "Entropy-based optimum test points selection for analog fault dictionary techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 3, pp. 754–761, 2004 (cit. on pp. 19, 20).
- [145] S. Mir, M. Lubaszewski, and B. Courtois, "Fault-based ATPG for linear analog circuits with minimal size multifrequency test sets," *Journal of Electronic Testing: Theory and Applications*, vol. 9, no. 1-2, pp. 43–57, 1996 (cit. on p. 20).
- [146] V. Zivkovic and A. Schaldenbrand, "Requirements for industrial analog fault-simulator," in *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, 2019, pp. 61–64 (cit. on p. 20).
- [147] T. M. et al., "Automatic fault simulators for diagnosis of analog systems," in *International Symposium on On-Line Testing and Robust System Design*, 2020 (cit. on p. 20).
- [148] S. Mir, M. Lubaszewski, and B. Courtois, "Unified built-in self-test for fully differential analog circuits," *Journal of Electronic Testing*, vol. 9, no. 1, pp. 135–151, 1996 (cit. on p. 20).
- [149] S. Mir, M. Lubaszewski, V. Kolarik, and B. Courtois, "Fault-based testing and diagnosis of balanced filters," *Analog Integrated Circuits and Signal Processing*, vol. 11, no. 1, pp. 5–19, 1996 (cit. on p. 20).

- [150] F. Maloberti, *Data Converters*. Springer, 2007 (cit. on p. 31).
- [151] M. Portolan, "Automated testing flow: the present and the future," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2952–2963, 2020 (cit. on p. 43).
- [152] S. Sunter, J.-F. Côté, and J. Rearick, "Streaming access to ADCs and DACs for mixed-signal ATPG," *IEEE Design & Test*, vol. 33, no. 6, pp. 38–45, 2016 (cit. on pp. 43–45).
- [153] "IEEE standard for access and control of instrumentation embedded within a semiconductor device," *IEEE Std 1687-2014*, 2014 (cit. on p. 44).
- [154] "IEEE standard for access and control of instrumentation embedded within a semiconductor device," *IEEE Std 1687-2014*, pp. 1–283, 2014 (cit. on p. 45).
- [155] "IEEE standard for terminology and test methods for analog-to-digital converters," *IEEE Std 1241-2010 (Revision of IEEE Std 1241-2000)*, pp. 1–139, 2011 (cit. on p. 47).
- [156] B. Esen, A. Coyette, G. Gielen, W. Dobbelaere, and R. Vanhooren, "Effective DC fault models and testing approach for open defects in analog circuits," in *Proc. IEEE International Test Conference*, Paper 3.2, 2016 (cit. on p. 50).
- [157] R. M. McDermott, "Random fault analysis," in *18th Design Automation Conference*, 1981, pp. 360–364 (cit. on p. 51).
- [158] V. D. Agrawal, "Sampling techniques for determining fault coverage in lsi circuits," *Journal of Digital Systems*, vol. 5, no. 3, pp. 189–202, 1981 (cit. on p. 51).
- [159] V. D. Agrawal, "Fault sampling revisited," *IEEE Design Test of Computers*, vol. 7, no. 4, pp. 32–35, 1990 (cit. on p. 51).
- [160] M. G. McNamer, S. C. Roy, and H. T. Nagle, "Statistical fault sampling," *IEEE Transactions on Industrial Electronics*, vol. 36, no. 2, pp. 141–150, 1989 (cit. on p. 51).
- [161] M. B. Santos, F. M. Goncalves, I. C. Teixeira, and J. P. Teixeira, "Defect-oriented verilog fault simulation of soc macros using a stratified fault sampling technique," in *Proceedings 17th IEEE VLSI Test Symposium*, 1999, pp. 326–332 (cit. on p. 51).
- [162] F. M. Goncalves and J. P. Teixeira, "Sampling techniques of non-equally probable faults in VLSI systems," in *Proceedings. 16th IEEE VLSI Test Symposium*, 1998, pp. 283–288 (cit. on p. 51).

- [163] J. T. De Sousa, F. M. Goncalves, J. P. Teixeira, C. Marzocca, F. Corsi, and T. W. Williams, "Defect level evaluation in an ic design environment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 10, pp. 1286–1293, 1996 (cit. on p. 54).
- [164] W. G. Cochran, *Sampling Techniques*, 3rd ed. John Wiley & Sons, 1977, pg.52 (Theorem 3.3) (cit. on p. 54).
- [165] P. Figueiredo and J. Vital, "The MOS capacitor amplifier," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 51, no. 3, pp. 111–115, 2004 (cit. on p. 55).
- [166] *Tessent defectsim user's manual*, Mentor Graphics Corporation, Wilsonville, OR, USA, September 2016 (cit. on p. 57).
- [167] A. Pavlidis, M. .-M. Louërat, E. Faehn, A. Kumar, and H. .-G. Stratigopoulos, "Symmetry-based A/M-S BIST (SymBIST): demonstration on a SAR ADC IP," in *Design, Automation & Test in Europe Conference Exhibition*, 2020, pp. 282–285 (cit. on pp. 64, 65).
- [168] Á. Gómez-Pau, S. Banerjee, and A. Chatterjee, "Real-time transient error and induced noise cancellation in linear analog filters using learning-assisted adaptive analog checksums," in *2014 IEEE 20th International On-Line Testing Symposium (IOLTS)*, 2014, pp. 25–30 (cit. on p. 67).

DECLARATION

Paris, July 2021

Antonios Pavlidis