System-Level Testing of Embedded Analogue Cores in SoC

Liquan Fang

2003

Ph.D. thesis
University of Twente

Twente University Press

Also available in print:
http://www.tup.utwente.nl/catalogue/book/index.jsp?isbn=9036519195

System-Level Testing

of

Embedded Analogue Cores in SoC

Promotiecommissie

Voorzitter:	Prof.dr.ir. A. J. Mouthaan	Univ. Twente / EWI
Promotor:	Prof.dr.ir. T. Krol	Univ. Twente / EWI
Assistent-promotor:	Dr.ir. H. G. Kerkhoff	Univ. Twente / EWI
Leden:	Prof.dr.ir H. Wallinga	Univ. Twente / EWI
	Prof.ir. A.J.M. van Tuijl	Univ. Twente / EWI
	Prof.dr. G. Gielen	KU Leuven
	Prof.dr. J. Figueras	Univ. Catalunya

The research described in this thesis was partially funded by the Dutch Technology Foundation (STW). The research was carried out at Testable Design & Testing of Microsystems research group (TDT), MESA+ research institute, University of Twente.

Twente University **Press**

Publisher:
Twente University Press,
P.O. Box 217, 7500 AE Enschede, the Netherlands,
www.tup.utwente.nl

Print: Océ Facility Services, Enschede

ISBN 9036519195

SYSTEM-LEVEL TESTING
OF
EMBEDDED ANALOGUE CORES IN SOC

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof.dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op donderdag 12 juni 2003 te 13.15 uur

door

Liquan Fang
geboren op 23 januari 1973
te Xiaoshan, Zhejiang, China

Dit proefschrift is goedgekeurd door
de promotor Prof.dr.ir. T. Krol en
de assistent-promotor Dr.ir. H. G. Kerkhoff

To Zhichun, Daniel
and my parents

Contents

Acknowledgement v

Summary vii

Abbreviations ix

Chapter 1. Introduction 1

1.1 The role of IC testing 2
1.2 Digital testing and analogue testing 2
1.2.1 Digital testing 2
1.2.2 Analogue testing 3
1.2.3 Mixed-signal testing 4
1.3 System-on-a-Chip (SoC) testing 5
1.4 Our approach for testing analogue embedded cores in SoC 6
1.5 Controllable input range and observable output range 9
1.6 Test-signal backtrace 11
1.7 Outline of the thesis 11
1.8 References 12

Chapter 2. Tolerance-box generation and propagation 15

2.1 Introduction 16
2.2 Core-based tolerance-box generation and propagation 17
2.3 Sensitivity-analysis based tolerance-box generation and propagation 19
2.3.1 Parameter-deviation model 19
2.3.2 Tolerance-box analysis for a stand-alone core 22
2.3.3 Propagation algorithm .... .25
2.4 Example circuit and simulation results 28
2.4.1 Example circuit 28
2.4.2 Parameter-deviation model of the example circuit 30
2.4.3 Simulation results 32
2.5 Conclusions 35
2.6 References 35

Chapter 3. Test-signal backtrace 37

3.1 Introduction 38
3.2 PID feedback-loop based test-signal backtrace approach 40
3.2.1 General structure 40
3.2.2 Theoretical background 41
3.2.3 Steady-state error 43
3.2.4 Sensitivity to a disturbance signal 44
3.2.5 Parameter tuning 45
3.3 Time and frequency domain simulation with the LPF 47
3.3.1 Example circuit and parameter tuning 47
3.3.2 Signal backtrace in the time and frequency domain 49
3.4 Test-signal backtracing 50
3.4.1 Structure of the test path 51
3.4.2 Test-signal backtracing for the test signals of the FILTER 51
3.4.3 Test-signal backtracing for the test signals of the VIC 54
3.4.4 Experimental results 55
3.5 Conclusions 56
3.6 References 57

Chapter 4. Mixed-level modelling for analogue fault simulation 59

4.1 Introduction 60
4.2 Defect Oriented Test (DOT) 61
4.3 Mixed-level modelling for analogue fault simulation 63
4.4 Example circuit and the mixed-level model construction 66
4.4.1 An actual industrial design: the data transceiver 66
4.4.2 The function description of the RECEIVER block 68
4.4.3 The mixed-level model of the RECEIVER block 70
4.4.4 Verification of the functional behaviour of the mixed-level model 71
4.5 Fault-simulation results 74
4.6 Conclusions 76
4.5 References 76

Chapter 5. Mixed-signal P1500-compatible core-based testing architecture. 79

5.1 Introduction 80
5.2 Core-based testing 81
5.2.1 Digital boundary-scan standard IEEE 1149.1 81
5.2.2 Mixed-signal test bus standard IEEE 1149.4 82
5.2.3 Digital core-based testing using the proposed standard IEEE P1500 83
5.2.4 The proposed mixed-signal core-based testing architecture 84
5.3 The hardware implementation of the proposed core-based testing architecture 86
5.3.1 Analogue input wrapper cell 87
5.3.2 Analogue output wrapper cell 89
5.3.3 Wrapper instruction register 89
5.3.4 TAP controller 91
5.3.4 User defined test access mechanism 91
5.4 Example circuit and simulation results 91
5.4.1 Example circuit 92

5.4.2 Simulation results for normal operation 93
5.4.3 Simulation results for the embedded core testing case 99
5.4.4 Simulation results for the interconnection testing case 101
5.4.5 Silicon area overhead of the proposed architecture 101
5.5 Conclusions 102
5.6 References 103

Chapter 6. Conclusions and recommendations 105

6.1 Conclusions 106
6.2 Original contributions of this thesis 108
6.3 Recommendations for future research 108
6.4 Reference 109

Appendix A. Parameter-deviation model and tolerance-box propagation algorithm 111

A.1 Parameter-deviation model 111
A.2 Tolerance-box propagation algorithm 112

Appendix B. DC convergence of the proposed test-signal backtrace approach 117

List of publications 123

Acknowledgement

Upon the completion of this thesis, I would like to thank all the people who have contributed to this thesis in one way or other. Specially, I would like to thank:

• my assistant promotor, Dr. ir. Hans G. Kerkhoff, for his invaluable guidance, advice, encouragement and conscientious review of the manuscript of this thesis. His constant confidence in the face of research uncertainties and his persistent questioning of our methods have been and will always be inspiring to me.

• my promotor, Prof. dr. ir. Thijs Krol, for his continuous encouragement and support during the last two years of this research work.

• Prof. dr. ir. Hans Wallinga for giving me a lot of support on my research work and personal life.

• Guido Gronthoud of Philips Research and Yizi Xing of Philips Semiconductors for giving me the opportunity to have the practical training in Philips and helping me a lot in other ways.

• my officemates Milan Stancic and Nur Engin for the nice and interesting discussions in a wide spectrum including mixed-signal testing, cars, child-care, computers, food, politics, etc.: the good mood in the office was sometimes the best thing keeping me going.

• my other TDT AIO colleagues Arun Antony Joseph, Herman Vermaak**,** Octavian Petre, Frank te Beest and Vladimir Zivkovic for the enjoyable discussions and the wonderful time we had.

• Rob Tijink for designing some circuits for this research work and providing high quality support for the CAD tools.

• Marcel Weusthof for helping me carry out the measurements.

• students M. C. Huijgen and M. H. ten Berge for implementing the mixed-signal core-based test architecture in CMOS technology.

• Cor Bakker and Frederik Reenders for their high quality maintenance of the PC network and fast and friendly support with all PC problems.

• Egbert Holl and Han Speek, for the maintenance of the HP-network.

• the administrative staff of ICE, the secretaries Annemiek Janssen, Miranda Wijk, Marie-Christine Predery, Margie Rhemrev, and the financial administrators Sophie Kreulen and Joke Vollenbroek. Thanks for keeping things organized and the work atmosphere enjoyable.

• Jay Kolhatker for the nice and interesting discussions we had (in the wide spectrum including the high-tech stuff, computers, food, language etc.)

- Ronald Tangelder for having the resourceful and enjoyable discussions during my first two years in the Netherlands.

- all the other members of ICE, who not only created a pleasant work environment but also gave me a lots of support for my work during the last four years.

- my Chinese friends for the wonderful time we had together.

Last but not the least, I want to thank my wife Zhichun and my son Daniel. They encourage and support me in every possible way to accomplish this Ph.D thesis. My parents' continuous encouragement and support also contribute immensely to my past and my today.

Summary

The increasing time-to-market (TTM) pressures have created the demand for rapid design of single complex chips based on reusing the design and test data. On the other hand, the availability of shrinking process technologies makes it possible to integrate a number of digital and analogue functional blocks into a single chip, mixed-signal System-on-a-Chip (SoC). Due to the complexity of a SoC and the limited test access (lack of controllability and observability) for embedded analogue cores in the pin-limited SoC, there are a lot of technical challenges in the testing of embedded analogue cores in SoC.

A hierarchical approach has been employed for testing of embedded analogue cores in our research. In this approach, the test signals for each standalone analogue core in the SoC are selected first. Next, some test translation schemes are employed to translate the core-level test signals into system-level test signals. The research presented in this thesis deals with the system-level testing of embedded analogue cores in SoC, i.e. how to translate the core-level test patterns into system-level.

The main task of the test-translation procedure is to propagate (backward and forward) those core-level test-input signals and test-output responses of each stand-alone core to the primary SoC inputs and outputs. Meanwhile, the corresponding tolerance boxes, which are the result of the allowed process-parameter variations during manufacturing, are also propagated to the primary SoC outputs for setting the corresponding optimal test thresholds. These thresholds are used to determine whether the chip under test is good or faulty during final production testing.

Tolerance-box generation in SoC testing is a very CPU-time consuming procedure with the traditional Monte-Carlo approach. By using the sensitivity analysis technique, this thesis proposes a new fast core-based tolerance-box generation and propagation approach for the testing of embedded analogue cores. In this approach, sensitivity analysis for each stand-alone analogue core is carried out first. Then, by using the proposed propagation algorithm, the tolerance box for the complete test path can be obtained. As the original points in our approach, the concept of the sensitivity is extended to core sensitivity and a new model for different categories of parameter deviations is proposed. The application of this approach to an example circuit shows that the proposed approach is very effective for the tolerance-box generation in the testing of embedded analogue cores.

Test signal backtracing is the procedure to determine a stimulus at the primary SoC inputs that will produce the desired test signal at the inputs of the embedded cores under test. A PID (Proportional Integral Derivative) feedback loop based backtrace method has been proposed in our research. With this proposed method, the test input signal for the embedded core could be backtraced to the primary SoC inputs in the time and frequency domain. Moreover, the computational effort for the backtrace procedure is low since it is quite easy to implement the PID controller in the high-level language during simulation. The presented theoretical analysis, simulation and measurement results show that the test-signals of the embedded cores can successfully be backtraced to the primary SoC inputs.

Analogue fault simulation is a CPU-time consuming procedure. A new general structure for mixed-level modelling with three stages is proposed in this thesis to speed up analogue fault simulation. In this structure, the original transistor-level circuits are reused for the input stage and output stage. The functional stage is an equation-based part to represent the function of the original fault-free block. This structure has been applied to one block of an actual industrial chip. The fault simulations have been carried out using this mixed-level model. The results show that this kind of mixed-level modelling can effectively reduce the fault-simulation time, while providing the same results for fault simulation.

During test translation, some core-level test patterns might not be able to translate to the system-level due to the limited test access of the embedded cores. In order to solve this problem, a mixed-signal P1500-compatible core-based testing architecture is proposed in this thesis. The digital cores with the P1500 test wrapper can be directly used in this architecture to get extra test access because of its good compatibility. The new analogue input and output wrapper cells and analogue test buses have been designed and used to provide test access for the analogue testing of embedded mixed-signal cores. As an example, one analogue test path of a SoC including the new core-based architecture was evaluated by means of simulation. The simulation results show that the performance degradation and silicon area overhead of the extra DfT are acceptable for testing of embedded analogue cores.

Summarising, with the methods proposed in this thesis, the core-level test patterns for embedded analogue cores can be translated into the system-level test patterns. In other words, the research presented in this thesis provides an effective solution to the system-level testing of embedded analogue cores in SoC.

Abbreviations

ADC	Analogue to Digital Converter
ABM	Analogue Boundary Module
AITB	Analogue Input Test Bus
AOTB	Analogue Output Test Bus
ATAP	Analogue Test Access Port
ATE	Automatic Test Equipment
ATPG	Automatic Test Pattern Generation
AWG	Arbitrary Waveform Generator
BIST	Build-In Self Test
BPF	Band Pass Filter
BSC	Boundary Scan Cell
DAC	Digital to Analogue Converter
DfT	Design for Test
DOT	Defect Oriented Test
DSP	Digital Signal Processing
ECUT	Embedded Core Under Test
EDA	Electronic Design Automation
FI	Functional Input
FO	Functional Output
IC	Integrated Circuit
IP	Intellectual Property
LPF	Low Pass Filter
MNA	Modified Node Analysis
PCB	Printed Circuit Board
PID	Proportional Integral Derivative
PLL	Phase Locked Loop
PPM	Part Per Million
PWL	Piece Wise Linear
RF	Radio Frequency
SNR	Signal-to-Noise Ratio
SoC	System on a Chip
TAM	Test Access Mechanism
TAP	Test Access Port
TBIC	Test Bus Interface Circuit
TCK	Test Clock
TDI	Test Data Input
TDO	Test Data Output
THD	Total Harmonic Distortion

TMS	Test Mode Selection
TTM	Time-To-Market
VIC	Voltage Current Converter
WCI	Wrapper Control Interface
WIP	Wrapper Interface Port
WIR	Wrapper Instruction Register
WSI	Wrapper Serial Input
WSO	Wrapper Serial Output

Chapter 1

Introduction

This chapter starts with a short introduction into the area of IC (integrated circuits) testing. It is followed by a brief overview of digital, analogue and mixed-signal testing. Then, System-on-a-Chip (SoC) testing is introduced. Next, the main challenges in testing of embedded analogue cores in a SoC are discussed and later on our hierarchical approach is introduced. In this approach, the test patterns for the standalone cores are generated first, then these core-level test patterns are translated into system-level test patterns by using the methods proposed in this thesis. Finally, the outline of the thesis is given in the last section of this chapter.

1.1 The role of IC testing

IC testing is a procedure to select a *faulty* chip or a *good* chip by applying the proper test stimuli generated by a test program to the chip under test, measuring the test responses and comparing these test responses with pre-defined thresholds. A *faulty* chip is here defined as the chip that has manufacturing defects and/or whose operations are out of the original design specifications.

The main reason for the necessity of production testing is the existence of unpredictable and uncontrollable phenomena in different steps of the manufacturing process. As the results of these imperfections in the manufacturing process, several types of defects can exist in manufactured ICs. The most common causes of defects in manufacturing are human errors, equipment failures, process instabilities, material instabilities, substrate in-homogeneity and lithographical spot defects [Sac98]. Those defects can cause various types of malfunctioning, depending on the IC topology and the nature of the defect.

On the other hand, nowadays consumers today demand high performance and quality in any of the electronic components that they buy [Rob97]. Low prices and years of problem-free operation with minimal maintenance are now the norm. In order for manufacturers to deliver such products, extensive testing procedures must be followed. This is to ensure that only good products are delivered to the consumers and that bad parts are rejected. A commonly mentioned rule of thumb of test is *the rule of ten* which suggests that the cost of detecting a bad component in a manufactured part increases tenfold at each level of assembly. Early discovering its presence of a defect chip is therefore most desirable. As a result, IC testing is very important for chip manufacturers.

In the semiconductor industry, the cost to fabricate a transistor has fallen dramatically, from one-tenth of a dollar cent in 1983 to less than one-thousandth of a cent today [Gar01]. At the same time, the cost of testing each transistor has remained relatively stable. As a result, it is expected that testing a transistor in the near future (around 2012) will cost the same amount of money as manufacturing it [Int01].

1.2 Digital testing and analogue testing

1.2.1 Digital testing

For conventional digital testing, the structural fault-model based testing approach and the standard stuck-at fault model have been widely used for a substantial period of time. The distinction between *good* and *faulty* is relatively clear for digital circuitry. As a result of that, CAD tools for automatic test-generation and test-circuitry insertion are already available for conventional digital testing since more than two decades. Since digital testing is not within the scope of this thesis, the details with regard to the digital testing are not discussed here. More information on the issue of digital testing, for instance delay fault testing and IDDQ testing, can be found in other literature e.g. the proceedings of International Test Conference.

1.2.2 Analogue testing

An obvious difference between digital testing and analogue testing is that the test stimuli and/or the test responses are analogue signals in the latter [Mil98]. This difference makes analogue testing much more difficult than digital testing. As commonly known, the number of values in an analogue signal is infinite, though the range is limited. This means that the search space for potential test stimuli is infinite during the analogue test-pattern generation process.

To complicate matters even more, the specification of an analogue circuit allows some variations in the circuit characteristics because of the possible process-parameter variations during manufacturing. For instance, the nominal value for the gain of an amplifier is 10dB. But 9.99 dB is also acceptable. Therefore, these output ranges called tolerance-boxes should be determined during the test-development stage [Tia97]. During the final production testing, it has to be determined if the output is within a certain *range*, not if it has a specific value. This also increases the complexity of analogue testing.

In general, each analogue circuit has its own unique specification during design. It is therefore reasonable that a *good* chip can be selected during the final chip testing by checking whether the chip conforms to the original design specification. This test methodology is called specification-based testing. Specification-based tests offer the advantage of ensuring that a circuit that passes the test process will meet the requirements of the user. The test generation, however, is manual and derived from the specifications. Its disadvantage is that the implementation of the test program can be very expensive for certain circuits. For instance high-frequency measurements are required for RF circuit testing.

The alternative test development methodology is called structural fault-model based test. In this approach, the manufacturing defects are modelled by certain fault models (e.g. bridging fault model), which can be inserted into the circuit schematic to emulate potential faulty circuits. For those faulty circuits, circuit simulations are carried out with test signals as inputs and the outputs are compared to the output of the fault-free circuit. If the difference between a faulty circuit output and a fault-free circuit output exceeds a pre-defined threshold, then the fault can be detected by applying the corresponding test pattern. In this way, the test patterns for final production testing can be determined. The procedure of simulating the fault-free circuit and all the faulty circuits with all the potential test-input signals is called analogue fault simulation. The main advantages of the structural fault model-based test are that low-cost test patterns can potentially be generated, the performance of the test patterns can be evaluated with their fault coverage and it makes automatic test-pattern generation feasible.

One crucial issue in analogue structural testing is that the analogue fault simulation always requires extremely long simulation times. This is because the number of possible faults in the fault list is large and analogue circuit simulation at the transistor level is very time-consuming. A possible solution to reduce the fault-simulation time is to use high-level models for the faulty-free blocks in the fault simulation. This means that the fault-free blocks are simulated using high-level models in order to reduce the simulation time.

Meanwhile, the faulty block is simulated at transistor level for simple and accurate injection of the fault into the fault-free netlist [Kaa98]. Crucial for this method is the generation of suitable high-level models of the fault-free analogue functional blocks or cores.

Another important item for structural fault model-based testing is the generation of the tolerance box. During analogue fault simulation, the fault-free circuit and faulty circuit have different tolerance boxes. The tolerance boxes for the fault-free circuit and each faulty circuit have to be generated. Hence, the total number of tolerance-boxes is ($n+1$) if the number of faults in the fault list is n. Since normally the fault list is quite large in analogue fault simulation, it is very important to generate the tolerance boxes fast and efficiently [Fan02a].

1.2.3 Mixed-signal testing

Besides the pure digital blocks and analogue blocks, it is quite often that there are mixed-signals blocks (e.g. ADC, DAC, PLL) in the mixed-signal ICs. During mixed-signal testing, some test stimuli and test responses are in the digital domain and others are in the analogue domain. More details with regard to ADC, DAC and PLL testing can be found in [Bus01].

For mixed-signal testing, *DSP-based testing* is a very important approach [Rob96, Mah87]. The basic structure of a mixed-signal DSP-based test system is shown in Figure 1.1.

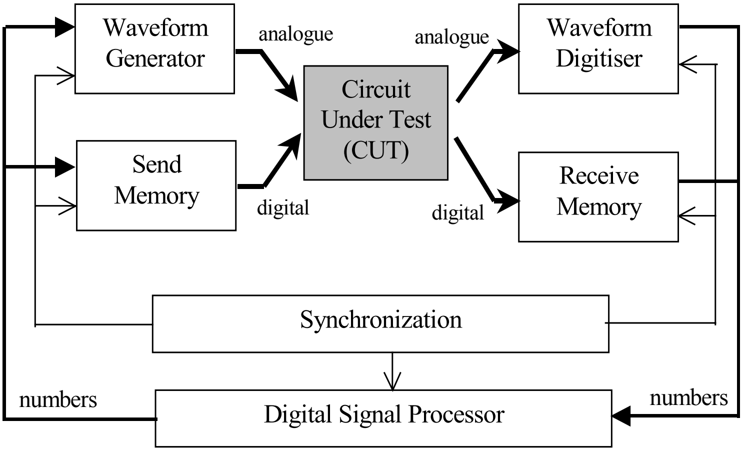


Figure 1.1: Fundamental structure of a DSP-based test system [Mah87].

In this structure, the *Synchronization* block provides the clock and phase synchronization for the whole system, analogue a well as digital parts, and the *Digital Signal Processor* block is the computing engine of the whole system. The basic idea behind DSP-based testing is that an analogue signal, for instance a sinus, is numerically computed by the digital computing engine and then applied to the *Waveform Generator* for conversion into the analogue domain. The resulting analogue signal is then applied to the *Circuit Under*

Test (CUT) from which its response is digitised by the *Waveform Digitiser* and passed on to the *Digital Signal Processor* for further processing. Depending on the measurement that is required, the appropriate software has to be loaded.

The procedure just described assumes that the *CUT* is an all-analogue chip. For the chip that produces digital output responses (e.g., ADC), the output digital signals are collected by a temporary RAM called the *Receive Memory* and then sent to the *Digital Signal Processor*. Similarly, if the *CU*T input is digital instead of analogue (e.g. DAC), the stimulus vectors would not be sent to the *Waveform Generator*, but to the *Send Memory* and replayed to the *CUT* input under the timing control of the *Synchronization* block (Figure 1.1).

In general, the DSP-based testing method offers a level of flexibility and repeatability that would be impossible with conventional analogue measurements. The analysis of captured data in the digital domain has also changed the measurement equipment capabilities substantially by the introduction of the concept of virtual instruments.

1.3 System-on-a-Chip (SoC) testing

The increasing time-to-market (TTM) pressures have created the demand for rapid design of single complex chips based on reusing the design and test data. On the other hand, the availability of shrinking process technologies makes it possible to integrate a number of digital and analogue functional blocks into a single chip, mixed-signal System-on-a-Chip (SoC) [Dol02, Nag99, Oze00]. These functional blocks can be standard interface blocks (e.g. analog/digital converters, phase-locked loop), reusable IP (Intellectual Property) cores (e.g. memory or microprocessors), and custom-designed “user blocks”. The SoC design concept makes the design process much more efficient than the conventional design methods and decreases the TTM because the system designer can order the cores from the core-providers, for instance on the internet [Dol02], on the Internet and reuse them in their SoC directly.

Digital core-based testing, which is being described by the proposed IEEE standard P1500 [P1500], is an effective test method for embedded digital cores in a SoC. In this approach, the test patterns for the stand-alone cores are generated by the core provider first. Afterwards, so-called test-wrapper cells as described in IEEE P1500 are used to provide test access for the embedded cores to apply the pre-generated test patterns [P1500].

Although the analogue circuitry part of mixed-signal SoCs comprises no more than 10-20% of the total circuitry of a chip, it takes up 70% of the test-development time [Nag99]. Besides test development, also in production, the analogue parts of a mixed-signal SoC can take a significant amount of time to test. In [Sma97] it was reported that analogue testing accounts for 50% of the production testing time. Besides the complexity of the SoC, another key factor that contributes to the SoC mixed-signal test complexity is the limited test access (lack of controllability and observability) for embedded analogue/mixed-signal cores in pin-limited SoCs. This hampers direct test-signal application and test-response evaluation. Therefore, how to develop effectively the efficient test patterns for testing the embedded analogue cores is a key issue in mixed-signal SoC testing.

Due to the complexity of testing the embedded analogue cores, a hierarchical approach is employed based on the previous concept of core-based testing. In the hierarchical approach, the test signals for each standalone analogue core in the SoC are selected first. Next, some test translation schemes are employed to translate the core-level test signals into system-level test signals.

The main previous work on system-level testing of embedded analogue/mixed-signals cores in SoC is presented in [Oze00, Oze01]. In this approach, by using the high-level models of the preceding cores and succeeding cores, the test signals at the inputs of the embedded core under test (ECUT) can be propagated to the primary SoC inputs and the test responses at the outputs of the ECUT can be propagated forwards to the primary SoC outputs. Therefore, high-level models of the embedded cores should be available if one wants to apply this test-translation scheme. In addition, the accuracy of the test signal/response propagation depends on the accuracy of the high-level models. Moreover, the tolerance-box propagation issue is not considered in the results of [Oze00] and [Oze01]. In our approach proposed in this thesis, however, high-level models of other cores are not required for the test translation and the tolerance-boxes are also propagated to the primary SoC outputs for setting the corresponding test thresholds.

1.4 Our approach for testing analogue embedded cores in SoC

In our approach, the core-based testing concept is used for testing embedded analogue cores in SoC. The flow diagram of our proposed approach is shown in Figure 1.2. In this approach, the test signals for stand-alone cores are suggested by the test engineer or obtained from the core-provider first. Then, the core-level test patterns are translated into system-level ones because the test stimuli can only be applied at the primary SoC inputs and the test responses can only be observed at the primary SoC outputs during final production testing.

For the analogue cores whose test patterns are not supplied by the core-provider, e.g. locally designed cores, the test patterns have to be generated by the test engineer at the core-user side. In our approach, the controllable input range and observable output range of the embedded analogue cores is derived first, based on the topology of the SoC and the operation ranges of the preceding and succeeding cores. Then, our test-pattern generation method based on analogue testability analysis, DOT and the bridging-fault model [Sta03] is employed to select the core-level test multi-frequency AC signals. During the core-level test-pattern generation, the controllable input range and observable output range can narrow the search space for potential test patterns. Consequently, the fault-simulation time can be reduced. How to obtain the controllable input range and the observable output range of the analogue embedded cores is discussed in section 1.5.

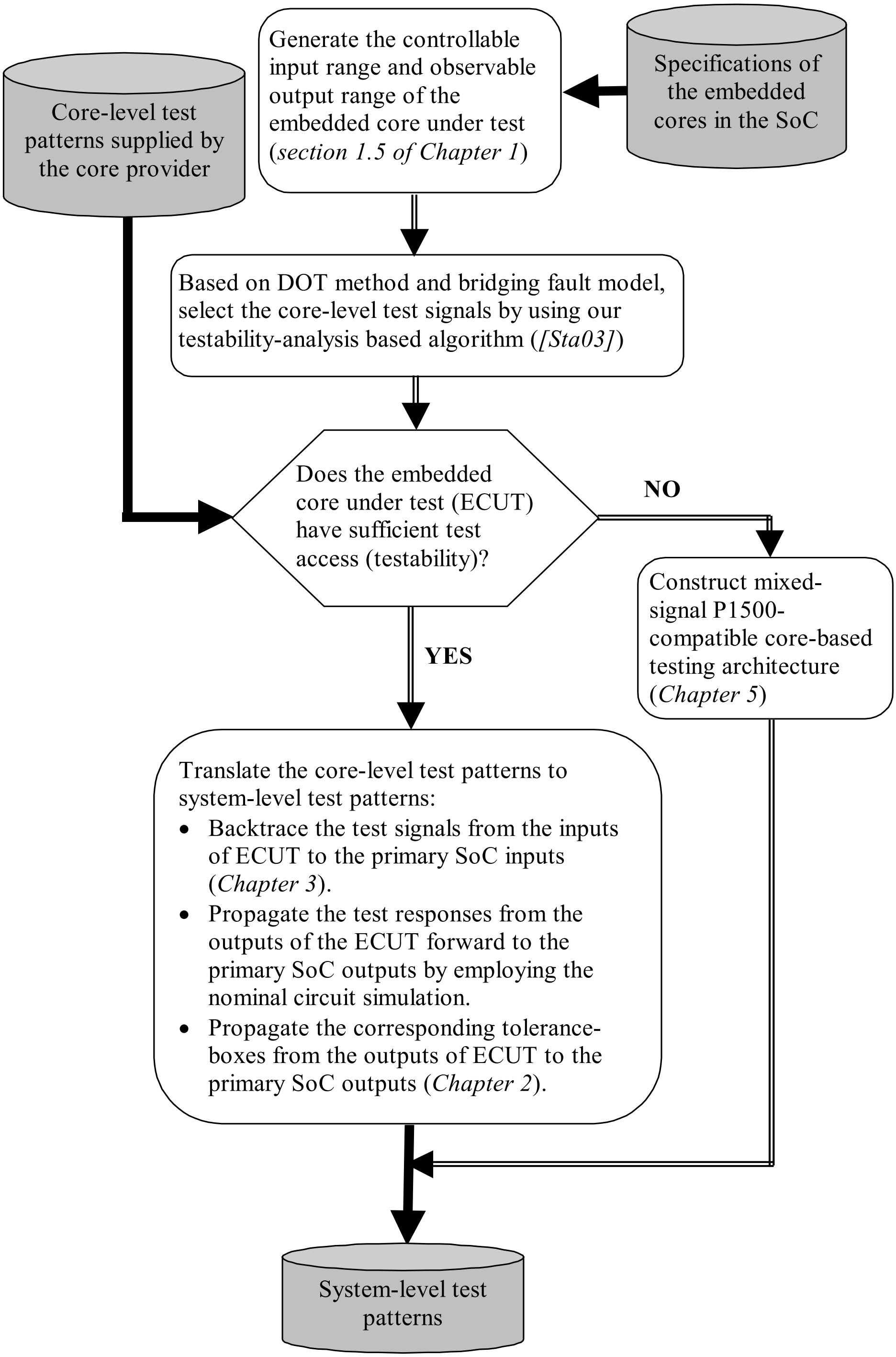

Core-level test patterns supplied by the core provider
Generate the controllable input range and observable output range of the embedded core under test (*section 1.5 of Chapter 1*)
Specifications of the embedded cores in the SoC
Based on DOT method and bridging fault model, select the core-level test signals by using our testability-analysis based algorithm (*[Sta03]*)
Does the embedded core under test (ECUT) have sufficient test access (testability)?
NO
YES
Construct mixed-signal P1500-compatible core-based testing architecture (*Chapter 5*)
Translate the core-level test patterns to system-level test patterns:
• Backtrace the test signals from the inputs of ECUT to the primary SoC inputs (*Chapter 3*).
• Propagate the test responses from the outputs of the ECUT forward to the primary SoC outputs by employing the nominal circuit simulation.
• Propagate the corresponding tolerance-boxes from the outputs of ECUT to the primary SoC outputs (*Chapter 2*).
System-level test patterns

Figure 1.2: The flow diagram of the proposed approach.

The main task of the test-translation procedure is to propagate (backwards and forwards) those core-level test-input signals and test-output responses of each stand-alone core to the primary SoC inputs and outputs as shown in Figure 1.3. Meanwhile, the corresponding tolerance boxes, which are the result of the allowed process-parameter variations during manufacturing, are also propagated to the primary SoC outputs for setting the corresponding test thresholds. These thresholds are set based on the tolerance-boxes of the faulty circuits and fault-free circuit. They are used to determine whether the chip under test is good or faulty during final production testing.

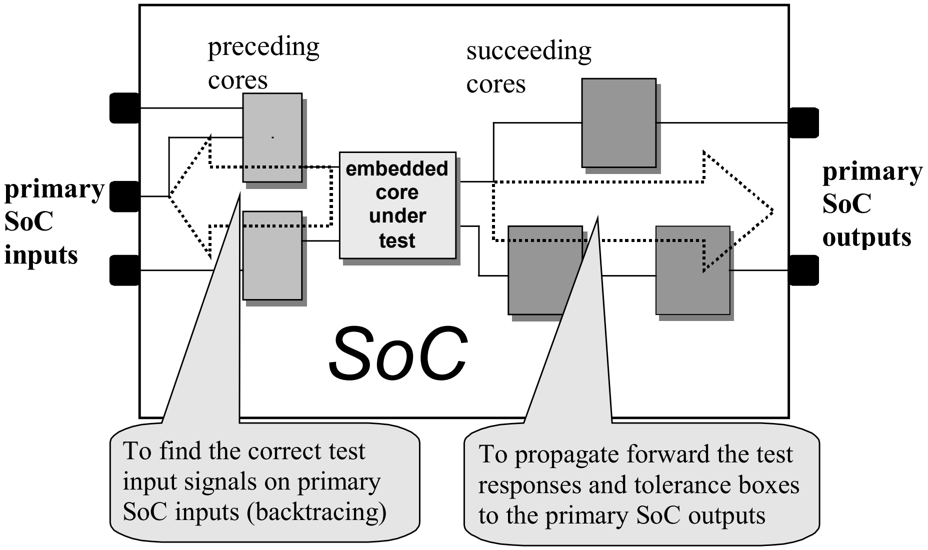

preceding cores
succeeding cores
primary SoC inputs
embedded core under test
primary SoC outputs
SoC
To find the correct test input signals on primary SoC inputs (backtracing)
To propagate forward the test responses and tolerance boxes to the primary SoC outputs

Figure 1.3: Illustration of translating the core-level test patterns to system-level ones.

More details about the test-signal backwards propagation (i.e. backtrace) is given in section 1.6. The test-response forward propagation can be carried out by employing normal circuit simulation. With regard to the tolerance-box generation and propagation, the traditional approach to obtain the tolerance-box is to perform Monte Carlo simulations at transistor level for the complete circuit. However, this becomes very CPU-time consuming for a SoC due to the complexity of a SoC. In our approach, a sensitivity-analysis based approach is proposed to reduce the CPU time used for the generation and propagation of the tolerance boxes [Fan02a].

During the test translation, it is also possible that some core-level test patterns cannot be realized for the embedded situation due to the limited test access. In order to test those embedded cores, extra controllable paths should be added to the SoC to allow access to inputs and outputs of each embedded core from the primary SoC pins. In our approach [Fan00], a new mixed-signal P1500-compatible core-based testing architecture is proposed to provide extra test access for the testing of low testability embedded cores.

1.5 Controllable input range and observable output range

The controllable input range and observable output range of embedded analogue cores can be illustrated as following. Assume that there are three embedded analogue cores connected as depicted in Figure 1.4. The controllable input range for *Core B* is imposed by the operating range and behaviour of *Core A*. The controllable input range for *Core C* is determined by *Core A* and *Core B*. On the other hand, the whole output range is assumed to be observable for *Core C*. However, the observable output range of *Core B* is not the whole output range any more. It can be constrained by the operating range and behaviour of *Core C*. Similarly, the observable output range of *Core A* can be constrained by the operating range and behaviour of *Core B* and *Core C*.

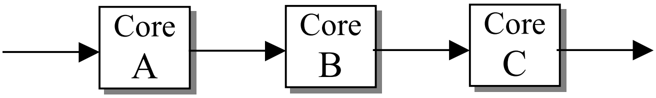


Figure 1.4: Illustration of a test path with three embedded cores.

The controllable input range and the observable output range of embedded cores can be obtained by combining the operational range of the preceding and succeeding cores. The following discussion shows how to obtain the controllable input range and the observable output range. Without the loss of any generality, the operational frequency ranges of three embedded cores shown in Figure 1.4 are assumed to be [0,5MHz], [1kHz,100kHz] and [0,1MHz] respectively (see Table 1.1).

Table 1.1: The operational frequency range (-3dB point) of three cores.

Core	AC operational range
Core A	Frequency: [0, 5MHz]
Core B	Frequency: [1kHz , 100kHz]
Core C	Frequency: [0, 1MHz]

With the data listed in Table 1.1, the controllable input frequency range and the observable output frequency range can be calculated as shown in Figure 1.5 by combining the corresponding operation frequency ranges of the preceding cores and the succeeding cores. Note that both the controllable input frequency range of *Core A* and the observable output frequency range of *Core C* are assumed to be [0, infinite). During the calculation, the controllable input frequency range of *Core B* can be obtained to be [0, 5MHz] by combining the range [0, infinite) and [0, 5MHz]. The controllable input frequency range of *Core C* can be calculated to be [1kHz, 100kHz] by combining [0, infinite), [0, 5MHz] and [1kHz, 100kHz]. Similarly, the observable output frequency range of *Core A* and *Core B* can be obtained to be [1kHz, 100kHz] and [0, 1MHz] respectively.

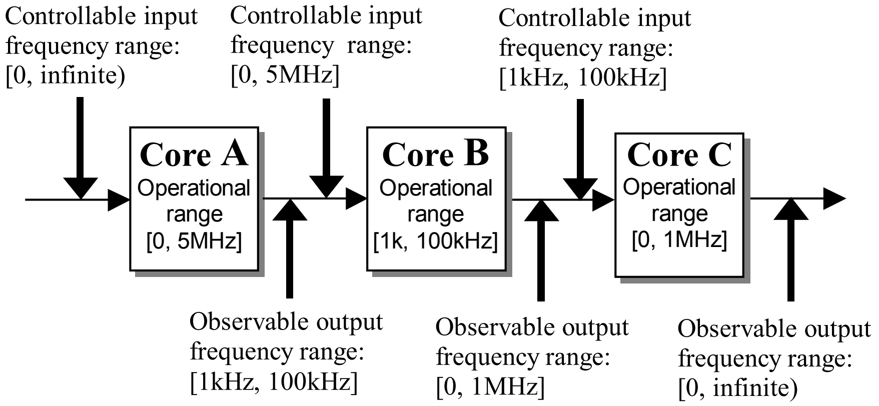

Controllable input frequency range: [0, infinite)
Controllable input frequency range: [0, 5MHz]
Controllable input frequency range: [1kHz, 100kHz]
Core A
Operational range [0, 5MHz]
Core B
Operational range [1k, 100kHz]
Core C
Operational range [0, 1MHz]
Observable output frequency range: [1kHz, 100kHz]
Observable output frequency range: [0, 1MHz]
Observable output frequency range: [0, infinite)

Figure 1.5: Illustration of the generation of the resulting controllable input ranges and the observable output ranges of the cores.

The operational frequency ranges of the embedded cores are assumed to be available during the above calculation. Actually, these operation ranges are available together with other specifications if the core is supplied by the core-provider. For instance, the analogue core HiAOP16 (High-Fi performance Audio-amplifier for driving speakers/headphones) is provided by the company Dolphin Integration [Dol02] with the following specifications available to the core user. The operational frequency range of this core is clearly specified to be from 100Hz to 24kHz in this table.

Table 1.2: The specification of an analogue core (partly)

Item	Specification
Output load	16 Ohm series to analog ground (VMC)
Output signal	single-ended
Internal gain	fixed by resistive feedback network
Output voltage range	0.71 Vrms
SNR	90 dB
Frequency band (-3dB point)	**(100Hz, 24kHz)**
THD	70 dB with input frequency = 1 kHz
Power consumption	2 400 μW with no load
Technological process	0.35 μm TSMC / UMC
Single power supply	2.7 V to 3.6 V
Temperature range	-40 °C to 85 °C

The controllable input range and observable output range of the analogue embedded cores are quite important for selecting the test signals for the standalone embedded cores. Due to the fact that the operation ranges of the embedded cores are provided as one part of the core specifications, the controllable input range and the observable output range can be obtained by combining the operation ranges of the preceding and succeeding cores in the SoC as having been illustrated in the above-mentioned example.

1.6 Test-signal backtrace

The test-signal backtrace procedure illustrated in Figure 1.6 can be defined as follows: given the DC, AC or transient signal desired at the input of the embedded core under test (ECUT), one requires to find the signal values or waveforms at the primary SoC inputs that result in the desired test signals at the inputs of ECUT.

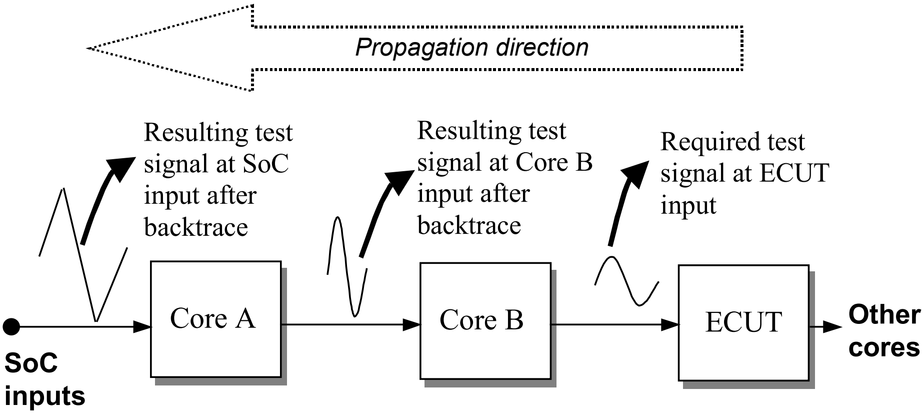

Propagation direction
Resulting test signal at SoC input after backtrace
Resulting test signal at Core B input after backtrace
Required test signal at ECUT input
Core A
Core B
ECUT
Other cores
SoC inputs

Figure 1.6: The illustration of the test-signal backtrace procedure.

In digital ATPG (automatic test-pattern generation), such a procedure finds an assignment of primary chip input values that cause a desired logic value to appear on a specified line in the circuit. While digital backtrace procedures are relatively easy to implement with the help of Boolean mathematics, there is no generic approach for backtracing analogue signals in mixed-signal SoC testing [Voo99]. The main reason is that the analogue test signal is continuous and the respective input-output relationships of the components used in analogue circuits are often expressed by nonlinear equations. In our approach [Fan02b], a new PID (Proportional Integral Derivative) [Ast95] feedback-loop based test-signal backtrace method is proposed.

1.7 Outline of the thesis

This thesis is focussed on the system-level testing of analogue embedded cores in a mixed-signal SoC environment. The body of the thesis is organized as follows:

In *Chapter 2,* a fast tolerance-box generation method is proposed for the testing of analogue embedded cores. In this approach, the sensitivity analysis for each stand-alone analogue core is carried out first. Subsequently, the tolerance box for the entire analogue test path can be obtained by using our proposed propagation algorithm. The key features of this approach are: the introduction of the concept of core sensitivity and the development of a new model for different categories of parameter deviations to solve the problem that parameter deviations for different cores can be possibly correlated. The theoretical analysis of this approach and the application to an example circuit are also presented and further mathematical details can be found in *Appendix A*.

A promising solution to the test-signal backtracing problem is presented in *Chapter 3*. It is referred to as the PID feedback-loop based test-signal backtrace approach. Motivated by concepts in control theory, a feedback loop with PID controller is constructed for test-signal backtracing in this approach. With this method, the test-input signals of an embedded analogue core under test can be easily propagated backwards to the primary SoC inputs. The computational effort of the proposed approach is also quite low. Its application to an example circuit and the simulation results as well as measurement results are also presented. The theoretical analysis of this approach is provided in *Appendix B*.

The technique of reducing the massive analogue fault-simulation time by using high-level modeling of fault-free cores is discussed in *Chapter 4*. The crucial issue for using this technique is the generation of the high-level models suitable for fault simulation. A new mixed-level model structure with three stages is proposed [Fan01]. In this structure, the original transistor-level circuits are reused for the input stage and output stage. The functional stage is an equation-based part to represent the function of the original fault-free block. This structure can guarantee the proper operation of the complete model if preceded or succeeded by a faulty block/core. This approach has been applied to an actual industrial chip and the fault simulations with the mixed-level model have been carried out.

Chapter 5 proposes a new mixed-signal P1500-compatible core-based testing architecture to provide extra test access for embedded cores that have low testability. In this architecture, the ideas developed in IEEE P1500 [P1500] are extended for embedded analog/mixed-signal cores. By using a wrapper-cell structure and analogue test buses, the digital/analogue test stimuli can be transported from the external SoC inputs to the inputs of the ECUT, and subsequently the digital/analogue test responses can be propagated from the ECUT to the external SoC outputs. In this architecture, the digital input and output wrapper cells can be identical to the ones suggested within P1500. New analogue input/output wrapper cells and an enhanced TAP (Test Access Port) controller have been developed. The proposed new core-based testing architecture has been applied to a locally designed SoC and the simulation results together with the overhead calculation are presented in this chapter.

Chapter 6 summarizes the research results and highlights our original contributions. Recommendations for further research work in the field of testing of embedded analogue cores in mixed-signal SoC are also given in this chapter.

1.8 References

[Ast95] K. Astrom and T. Hagglund, “PID controllers: theory, design and tuning,” Instrument of Society America, ISBN:1-55617-516-7, USA, 1995.

[Bus01] M. L. Bushnell and V. D. Agrawal, “Essentials of electronic testing for digital, memory & mixed-signal VLSI circuits,” Kluwer Academic Publishers, ISBN: 0-7923-7991-8, 2001.

[Dol02] Dolphin Integration, "Jazz Line of analog and mixed-signal virtual components," http://www.dolphin.fr/flip/analog/analog_overview.html

[Fan00] L. Fang, M. Stancic and H. G. Kerkhoff, “Mixed-signal core-based testing,” *Inf. Proc. of European Test Workshop (ETW) 2000*, pp. 279-280, May 2000.

[Fan01] L. Fang, G. Gronthoud and H. G. Kerkhoff, “Reducing analogue fault-simulation time by using high-level modeling in Dotss for an industrial design”, *Proc. of IEEE European Test Workshop*, ISBN 0-7695-1877, Stockholm, Sweden, pp. 61-67, 2001.

[Fan02a] L. Fang and H. G. Kerkhoff, "Sensitivity analysis based tolerance-box generation and propagation in mixed-signal SoC testing", *Proc. of 7th IEEE European Test Workshop*, Corfu, Greece, pp. 255-260, May 2002.

[Fan02b] L. Fang and H. G. Kerkhoff, “PID feedback based backtrace method for mixed-signal testing,” *Proc. of 8th IEEE International Mixed-Signal Testing Workshop*, Switzerland, pp. 147-152, 2002.

[Gar01] R. Garcia, “Distributed test strategy cuts cost,” *Test & Measurement World (on line)*, November 2001.

[Int01] “International technology roadmap for semiconductors, 2001 edition,” http://public.itrs.net/Files/2001ITRS/Home.htm

[Kaa98] V. Kaal, "Structural test generation for analog macros", Ph.D. thesis, University of Twente, the Netherlands, 1998.

[Mah87] M. Mahoney, “DSP-based testing of analogue and mixed-signal circuits,” IEEE Computer Society Press, ISBN: 0-8186-0785-8, 1987.

[Mil98] L. S. Milor, “A tutorial introduction to research on analog and mixed-signal circuit testing,” *IEEE Trans. Circuits Systems—II: Analog and Digital Signal Processing*, vol. 45, no. 10, pp. 1389-1407, 1998.

[Nag99] N. Nagi, “System-on-a-chip mixed-signal test: issues, current industry practices and future trends,” *Proc. of 5th International Mixed Signal Testing Workshop,* Whistler, BC, Canada, pp. 201-211, 1999.

[Oze00] S. Ozev, I. Bayraktaroglu and A. Orailoglu, "Test synthesis for mixed-signal SOC paths," *Proc. of Design, Automation and Test in Europe (DATE) Conference*, pp. 128-133, 2000.

[Oze01] S. Ozev and A. Orailoglu, “System-level test synthesis for mixed-signal designs,” *IEEE Trans. Circuits and Systems—II: Analog and Digital Signal Processing*, vol. 48, no. 6, pp. 588-599, June 2001.

[P1500] Homepage of IEEE P1500 working group, http://grouper.ieee.org/groups/1500/.

[Rob96] G. W. Roberts, “Metrics, techniques and recent developments in mixed-signal testing,” *Proc. of IEEE/ACM International Conference on Computer-Aided Design*, pp. 514-521, Nov. 1996.

[Rob97] G. W. Roberts, “Improving the testability of mixed-signal integrated circuits,” *Proc. of IEEE Custom Integrated Circuits Conference*, pp. 214-221, May 1997.

[Sac98] M. Sachdev, “Defect Oriented Testing for CMOS analog and digital circuits”, Kluwer Academic Publishers, ISBN 0-7923-8083-5, 1998.

[Sta03] M. Stancic, “Test selection for analogue cores based on the testability transfer function,” Ph.D thesis, University of Twente, Enschede, the Netherlands, 2003.

[Sma97] C. H. Small, “Mixed-signal methods shift gears for tomorrow’s systems-on-a-chip,” *Computer Design*, October 1997.

[Tia97] M. W. Tian and C. J. R. Shi, "Rapid frequency domain analog fault simulation under parameter tolerances," *Proc. of Design Automation Conference (DAC97)*, pp. 275-280, 1997.

[Voo99] R. Voorakaranam and A. Chatterjee, "Hierarchical test generation for analog circuits using incremental test development," *Proc. of IEEE VLSI Test Symposium (VTS)*, pp. 296–301, April 1999.

Chapter 2

Tolerance-box generation and propagation

In this chapter, a sensitivity-analysis based approach is proposed to reduce the CPU time used for the tolerance-box generation and propagation in mixed-signal System-on-Chip (SoC) testing. The sensitivity analysis for each stand-alone analogue core is carried out first in this approach. Next the tolerance box for the entire analogue test path can be obtained by using the proposed propagation algorithm. Compared to the normal sensitivity analysis, the concept of core sensitivity is introduced for the proposed propagation algorithm. Moreover, in order to solve the problem that the parameter deviations for different cores can be possibly correlated, a new model for different categories of parameter deviations is also proposed. The presented simulation results show that this new approach is much faster (about 140 times) than the CPU-time consuming Monte Carlo simulation approach while the loss of accuracy is smaller than 1.36%.

2.1 Introduction

As discussed in Chapter 1, the tolerance box is a fundamental issue for analogue testing because the specification of an analogue circuit allows for some variations in the circuit characteristics. These variations are a result of the allowed deviations of the process parameters. During analogue fault simulation, as a result, the responses for both good and faulty circuits will lie in bands. They are called good-response tolerance box and faulty-response tolerance box respectively [Tia97, Kaa98]. These tolerance-boxes are quite important for selecting the test signals and setting the optimal test thresholds. During test pattern generation, the test signal is selected if the overlap of the corresponding good-response and faulty-response tolerance-boxes is minimal. Meanwhile, the optimal test thresholds, which determine whether the chip under test is good or faulty during production testing, are set to minimize the detect error based on the tolerance-boxes.

Generally, the fault-free circuit and faulty circuit have different tolerance boxes. Therefore, for the fault-free circuit and each faulty circuit, the corresponding tolerance box should be generated. The total number of tolerance-boxes is therefore $(n+1)$ if the number of faults in the fault list is n. Since normally the fault list is quite large in analogue fault simulation, it is very important to generate the tolerance box fast and efficiently.

The traditional and most widely used approach in the past to obtain the tolerance box is to perform Monte Carlo simulations at transistor level for the complete circuit. With this technique, the simulation is repeated for random combinations of values chosen from within the range of each parameter. Unfortunately, if the number of iterations for the simulation is not very large, the Monte Carlo simulation always underestimates the tolerance box [Shi99]. Therefore, this can be the source to produce non-robust test sets resulting that a fault undetectable at a given test point may be claimed to be detectable. Accurately determining the bounds on the circuit performance requires a large number of simulations. Therefore, this method becomes very CPU-time consuming if the chip becomes large.

In recent years, some methods have been proposed to speed up the tolerance-box generation in analogue fault simulation. The interval-analysis method has been used to calculate tolerance boxes for linear analogue circuits in [Tia96, Tia97, Tia00, Shi99]. In order to improve the efficiency and accuracy of interval-arithmetic based tolerance analysis, a genetic algorithm has been used for optimisation in [Fem99]. In [Ham93], sensitivity analysis is used to calculate the tolerance for analogue circuit testing.

However, because the number of transistors in mixed-signal SoC and the number of faults in the fault list are quite large, all the above-mentioned methods are not suitable for the tolerance-box generation in mixed-signal SoC testing. A solution to the fast tolerance-box generation and propagation in mixed-signal SoC testing is still not available.

As we know, the average simulation time of the SPICE-based circuit simulators for linear circuits is in the order of $O(M^3)$ with M being the number of nodes. For nonlinear circuits it becomes even worse, and hence it is advantageous to break up a single set of nodes into a number of smaller sub-circuits [Vla94]. Moreover, with the increase of M, the advantage of breaking up the large circuit into small sub-circuits becomes more significant. On the other hand, due to the complexity of SoC, the chip is always partitioned into several smaller functional cores [Nag99, Oze00], or called analogue virtual components in [Dol02]. Therefore, with respect to the tolerance-box generation in large mixed-signal SoC, the total simulation time can be reduced if the tolerance-box for the complete test path can be obtained by performing the simulations of each stand-alone (relatively small) analogue core and combining those tolerance boxes with a simple propagation algorithm.

Motivated by this idea, a novel fast sensitivity-analysis based tolerance-box generation and propagation algorithm for the analogue test paths in mixed-signal SoC consisting of analogue embedded cores is proposed in this chapter. In our approach, the sensitivity analysis is carried out for each stand-alone core first. Then, by using the proposed propagation algorithm, the tolerance box for the whole test paths with analogue embedded cores can be obtained.

2.2 Core-based tolerance-box generation and propagation

In general, the core-based tolerance-box generation approach consists of the following two steps. In the first step, the tolerance box for each stand-alone analogue core in the test path is generated. In the second step, the tolerance-box for the whole analogue test path is obtained by using an algorithm to combine the tolerance boxes obtained in the first step. As discussed in the previous section, the total simulation time used for tolerance-box generation can be reduced because the simulation is carried out on relatively small cores, instead of the large whole SoC [Fan01a, Fan01b, Fan02].

However, during manufacturing the whole chip is facing the same process conditions, the parameter deviations of different cores are therefore correlated, which are mainly caused by the process deviations [Saa00]. Because of this correlation, the tolerance boxes for different cores in the same SoC are correlated. This correlation issue will be a big hamper in the propagation step of the core-based tolerance-box generation approach. Hence, in order to apply this core-based tolerance-box generation approach, the crucial problem with regard to the correlation of the parameter deviations for different cores should be solved.

Additionally, there are some other general requirements for applying the core-based tolerance-box generation approach. First of all, the computational effort for generating the tolerance-box for stand-alone cores should not be high. Second, the data format of the obtained tolerance box for a stand-alone core should be suitable for propagation. Third, the computational effort for propagation should be low.

In order to respect the above-mentioned requirements, a sensitivity-analysis based tolerance-box generation and propagation method is proposed. The flow chart of this approach is shown in Figure 2.1. The sensitivity analysis is employed in this approach because it is already known that the sensitivity analysis can be used to estimate the worst-case response of any analogue circuit and this method is known to be not computationally expensive [Shi99, Ogr94]. Compared to the normal sensitivity analysis, two new items are proposed to make the sensitivity-analysis approach able to be used for core-based tolerance-box propagation. First, the concept of core sensitivity is proposed to enable the propagation. Second, a new model of different sorts of parameter deviations is proposed to solve the problem that the parameter deviations for different cores could be correlated.

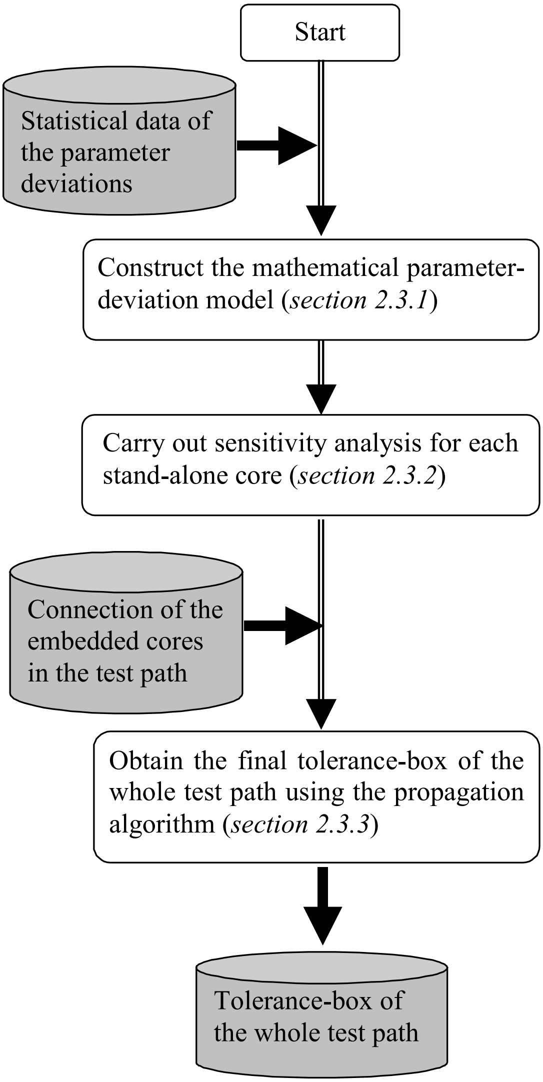

Start
Statistical data of the parameter deviations
Construct the mathematical parameter-deviation model (*section 2.3.1*)
Carry out sensitivity analysis for each stand-alone core (*section 2.3.2*)
Connection of the embedded cores in the test path
Obtain the final tolerance-box of the whole test path using the propagation algorithm (*section 2.3.3*)
Tolerance-box of the whole test path

Figure 2.1: The flow chart of the proposed tolerance-box generation and propagation approach.

The details of the items in the flow chart shown in Figure 2.1 are discussed in the next section. First, the mathematical model of the parameter deviations for different embedded analogue cores in SoC is proposed in subsection 2.3.1. Afterwards, the sensitivity-analysis based tolerance-box generation method for a stand-alone core is discussed in subsection 2.3.2. Finally, in subsection 2.3.3, the concept of core sensitivity is defined and the propagation algorithm is proposed.

2.3 Sensitivity-analysis based tolerance-box generation and propagation

In this section, the details of the proposed tolerance-box generation and propagation approach are discussed, including the parameter-deviation model, the tolerance-box analysis for a stand-alone core and the propagation algorithm.

2.3.1 Parameter-deviation model

As indicated in section 2.2, the parameter deviations of different embedded cores in the same SoC could be correlated. As a result of that, the tolerance boxes of different cores could be correlated. This correlation issue is hampering the application of the propagation algorithm in the core-based tolerance-box generation approach. In order to solve this problem, a mathematical model of the parameter deviations for different embedded analogue cores in SoC is proposed in this subsection.

As an illustration without loss of any generality, core A, core B, core C and core D are connected as shown in Figure 2.2. In this figure, the new mathematical model for different categories of parameter deviations is also shown. In Figure 2.2, $\Delta p_i^G (i = 1,2,\cdots,n)$ are the deviations of the possible *global* parameters for all the cores (G). The deviations of the *local* parameters for core A, core B, core C and core D are $\Delta p_i^A (i = 1,2,\cdots,n_A)$, $\Delta p_i^B (i = 1,2,\cdots,n_B)$, $\Delta p_i^C (i = 1,2,\cdots,n_C)$ and $\Delta p_i^D (i = 1,2,\cdots,n_D)$ respectively. One point that should be made clearly here is that the global parameter deviations and local parameter deviations defined in this model have only a mathematical meaning. The real physical parameter deviations in each core are the combinations of local parameter deviations and global parameter deviations. For instance, the real parameter deviation in core B is a specific mathematical combination of $\Delta p_i^G (i = 1,2,\cdots,n)$ and $\Delta p_i^B (i = 1,2,\cdots,n_B)$. Moreover, the deviations $\Delta p_i^G (i = 1,2,\cdots,n)$, $\Delta p_i^A (i = 1,2,\cdots,n_A)$, $\Delta p_i^B (i = 1,2,\cdots,n_B)$ $\Delta p_i^C (i = 1,2,\cdots,n_C)$ and $\Delta p_i^D (i = 1,2,\cdots,n_D)$ are made to be mutually independent when the model is constructed. In other words, the local parameter deviations in different cores and the global parameter deviations in this model are independent from each other.

Although the local parameter deviations in different cores are mutually independent in this model, as proved in Appendix A, the correlation between the parameter deviations of different cores can be modelled using the same global parameter deviations.

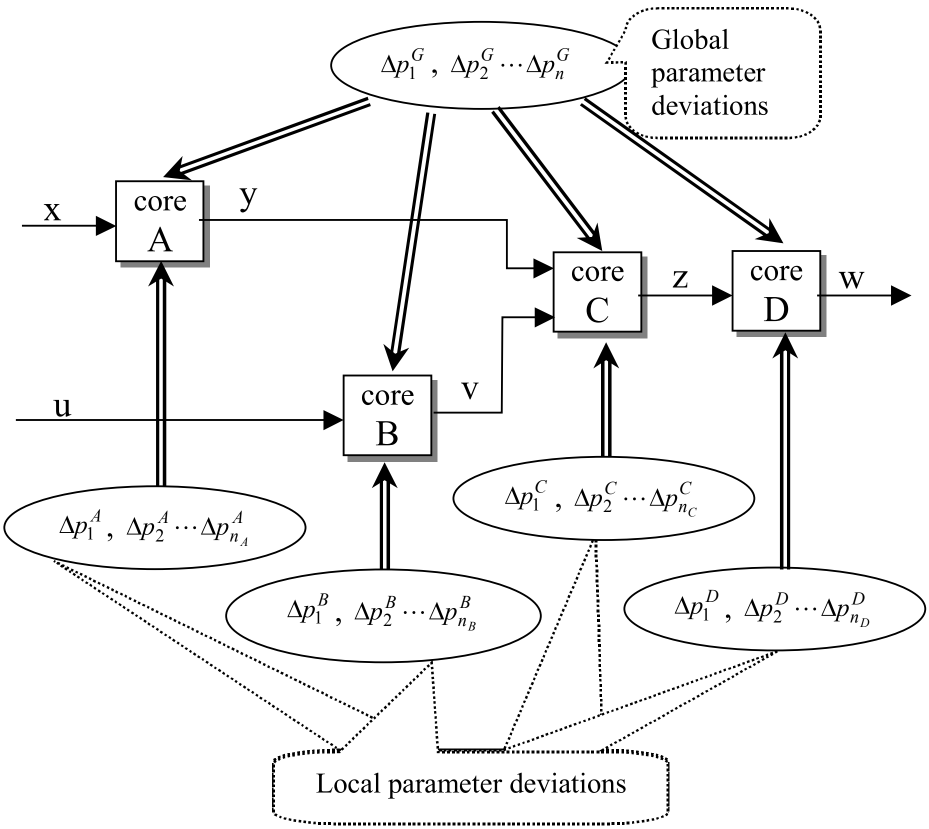

Δp_1^G , $\Delta p_2^G \cdots \Delta p_n^G$
Global parameter deviations
x
core A
y
core C
z
core D
w
u
core B
v
Δp_1^A , $\Delta p_2^A \cdots \Delta p_{n_A}^A$
Δp_1^B , $\Delta p_2^B \cdots \Delta p_{n_B}^B$
Δp_1^C , $\Delta p_2^C \cdots \Delta p_{n_C}^C$
Δp_1^D , $\Delta p_2^D \cdots \Delta p_{n_D}^D$
Local parameter deviations

Figure 2.2: Illustration of the generic system structure with different categories of parameter deviations.

In order to show how to use the equations presented in Appendix A to construct the parameter-deviation model, a simple example is presented as following. Consider a test path containing three embedded analogue cores: the Pre_amplifier, the FILTER and the VIC (Figure 2.6). The deviation of the *channel length* of the MOS transistors in these three embedded cores is the parameter deviation considered here. First of all, a parameter-deviation model is constructed for this example. In this model, the global parameter deviation is ΔL^G and the local parameters deviations for the Pre_amplifier, the FILTER and the VIC are ΔL^L_{amp}, ΔL^L_{fil} and ΔL^L_{vic} respectively. The real physical channel length deviation in these three cores are ΔL_{amp}, ΔL_{fil} and ΔL_{vic}. The mean value and variance of ΔL_{amp}, ΔL_{fil} and ΔL_{vic} are specified in the available statistical data as following:

$$\mu_{\Delta L_{amp}} = \mu_{\Delta L_{fil}} = \mu_{\Delta L_{vic}} = 0 \tag{2.1}$$

$$(\sigma_{\Delta L_{amp}})^2 = (\sigma_{\Delta L_{fil}})^2 = (\sigma_{\Delta L_{vic}})^2 = 2.79 \times 10^{-16} \tag{2.2}$$

Because these three cores are embedded in the same SoC, the transistor channel length deviations in these three cores are correlated. The correlation coefficients between them are given as:

$$\rho_{\Delta L_{amp},\Delta L_{fil}} = 0.93 \tag{2.3}$$

$$\rho_{\Delta L_{fil},\Delta L_{vic}} = 0.93 \tag{2.4}$$

$$\rho_{\Delta L_{amp},\Delta L_{vic}} = 0.9 \tag{2.5}$$

According to equation (A.1) in appendix A, the real physical channel length deviation in these three embedded cores can be expressed as:

$$\Delta L_{amp} = a_1 \cdot \Delta L^G + b_1 \cdot \Delta L^L_{amp} \tag{2.6}$$

$$\Delta L_{fil} = a_2 \cdot \Delta L^G + b_2 \cdot \Delta L^L_{fil} \tag{2.7}$$

$$\Delta L_{vic} = a_3 \cdot \Delta L^G + b_3 \cdot \Delta L^L_{vic} \tag{2.8}$$

Using equations (A.2), (2.6), (2.7) and (2.8), the mean value of the real length deviations in these three cores can be expressed as:

$$\mu_{\Delta L_{amp}} = a_1 \cdot \mu_{\Delta L^G} + b_1 \cdot \mu_{\Delta L^L_{amp}} \tag{2.9}$$

$$\mu_{\Delta L_{fil}} = a_2 \cdot \mu_{\Delta L^G} + b_2 \cdot \mu_{\Delta L^L_{fil}} \tag{2.10}$$

$$\mu_{\Delta L_{vic}} = a_3 \cdot \mu_{\Delta L^G} + b_3 \cdot \mu_{\Delta L^L_{vic}} \tag{2.11}$$

Similarly, using equations (A.3), (2.6), (2.7) and (2.8), the variances of the real length deviations in these three cores can be expressed as:

$$(\sigma_{\Delta L_{amp}})^2 = (a_1)^2 \cdot (\sigma_{\Delta L^G})^2 + (b_1)^2 \cdot (\sigma_{\Delta L^L_{amp}})^2 \tag{2.12}$$

$$(\sigma_{\Delta L_{fil}})^2 = (a_2)^2 \cdot (\sigma_{\Delta L^G})^2 + (b_2)^2 \cdot (\sigma_{\Delta L^L_{fil}})^2 \tag{2.13}$$

$$(\sigma_{\Delta L_{vic}})^2 = (a_3)^2 \cdot (\sigma_{\Delta L^G})^2 + (b_3)^2 \cdot (\sigma_{\Delta L^L_{vic}})^2 \tag{2.14}$$

Let the mean values of the global parameter deviation and the local parameter deviations in the parameter-deviation model be zero, i.e.:

$$\mu_{\Delta L^G} = \mu_{\Delta L^L_{amp}} = \mu_{\Delta L^L_{fil}} = \mu_{\Delta L^L_{vic}} = 0 \tag{2.15}$$

According to equations (2.9), (2.10) and (2.11), the required mean values of the channel length deviation defined in equation (2.1) can be satisfied. Let the variances of the global parameter deviation and the local parameter deviations be:

$$(\sigma_{\Delta L^G})^2 = (\sigma_{\Delta L^L_{amp}})^2 = (\sigma_{\Delta L^L_{fil}})^2 = (\sigma_{\Delta L^L_{vic}})^2 = 2.79 \times 10^{-16} \tag{2.16}$$

In order to satisfy the required variances of the channel length deviation given in equation (2.2), using equations (2.12), (2.13) and (2.14) one can obtain:

$$(a_i)^2 + (b_i)^2 = 1 \quad (i = 1,2,3) \tag{2.17}$$

Additionally, using equation (A.8) and the required correlation coefficients defined in equations (2.3), (2.4) and (2.5), one can obtain:

$$a_1 \cdot a_2 = 0.93 \tag{2.18}$$

$$a_2 \cdot a_3 = 0.93 \tag{2.19}$$

$$a_1 \cdot a_3 = 0.9 \tag{2.20}$$

Using equations (2.17), (2.18), (2.19) and (2.20), the values for $a_i (i = 1,2,3)$ and $b_i (i = 1,2,3)$ can be determined as following:

$$a_1 = 0.949 \quad , \quad b_1 = 0.315 \tag{2.21}$$

$$a_2 = 0.980 \quad , \quad b_2 = 0.199 \tag{2.22}$$

$$a_3 = 0.949 \quad , \quad b_3 = 0.315 \tag{2.23}$$

The above example shows how to construct the parameter-deviation model based on the available statistical data of the real physical parameter deviations. On the other hand, it also shows that the correlation of the real physical parameter deviations in different cores can be modelled with the proposed parameter-deviation model.

2.3.2 Tolerance-box analysis for a stand-alone core

In the previous subsection, how to construct the parameter-deviation model with the global parameter deviations and the local parameter deviations is introduced. After the parameter-deviation model is ready, how to perform the tolerance-box analysis for each core is discussed in this subsection.

Suppose that $f(p_1, p_2,..., p_n)$ is the circuit output performance function with parameters $p_i \ (i = 1,2,\cdots,n)$. For instance, f can be the DC output value and $p_i \ (i = 1,2,\cdots,n)$ can be the length and width of the MOSFET channels, gate oxide thickness and sheet resistance value etc.. The small change in the sensitivity of the performance function f with respect to a parameter p_i can be defined as its partial derivative with respect to p_i calculated at the nominal point [Ogr94]:

$$s_{p_i}^{f} = \frac{\partial f}{\partial p_i} \tag{2.24}$$

where the nominal point is defined as the performance function value with nominal parameter values. This sensitivity is a measure of the absolute change in the circuit output performance per unit absolute change in a parameter. So, given a differential approximation by increments, we can estimate the output performance increment caused by one parameter deviation Δp_i as:

$$\Delta f_i = s_{p_i}^f \cdot \Delta p_i \tag{2.25}$$

where Δp_i is a random variable with certain probability distribution as illustrated in Figure 2.3. Assume the deviation for each parameter p_i $(i = 1,2,\cdots,n)$ is Δp_i, then the deviation of the performance under all the parameter deviations can be expresses as [Saa00]:

$$\Delta f = \sum_{i=1}^{n} (s_{p_i}^f \cdot \Delta p_i) \tag{2.26}$$

Since $\Delta p_i (i = 1,2,\cdots,n)$ are random variables, Δf given in the above equation is also a random variable.

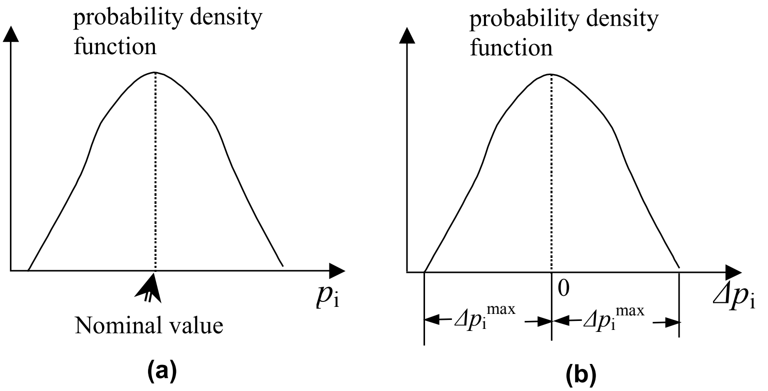


Figure 2.3: (a) Illustration of the probability density function of one parameter p_{i} (e.g. the channel length of NMOS transistor), (b) illustration of the probability density function of the corresponding parameter deviation Δp_i.

In the case the statistical data of $\Delta p_i (i = 1,2,\cdots,n)$ is not available and only the ranges of $\Delta p_i (i = 1,2,\cdots,n)$ are available, then by using equation (2.26) the lower and upper bounds of the output performance f under the parameter deviations can be given by:

$$f_{\min} = f_0 + \min\{\Delta f\} = f_0 + \min\left\{\sum_{i=1}^{n} (s_{p_i}^f \cdot \Delta p_i)\right\} \tag{2.27}$$

and

$$f_{\max} = f_0 + \max\{\Delta f\} = f_0 + \max\left\{\sum_{i=1}^{n} (s_{p_i}^f \cdot \Delta p_i)\right\} \tag{2.28}$$

where f_0 is the nominal value of the performance function, $\max\{\Delta f\} \geq 0$ and $\min\{\Delta f\} \leq 0$.

Let X and Y be two independent random variables, then the following two equations hold [Pee93]:

$$\max\{X + Y\} = \max\{X\} + \max\{Y\} \tag{2.29}$$

$$\min\{X+Y\}=\min\{X\}+\min\{Y\} \tag{2.30}$$

Hence if the deviations $\Delta p_i (i=1,2,\cdots,n)$ are mutually independent, equations (2.27) and (2.28) can be written as:

$$f_{\min}=f_0+\sum_{i=1}^{n}\min\left\{s_{p_i}^{f}\cdot\Delta p_i\right\} \tag{2.31}$$

$$f_{\max}=f_0+\sum_{i=1}^{n}\max\left\{s_{p_i}^{f}\cdot\Delta p_i\right\} \tag{2.32}$$

For the special case that all the parameter deviations are symmetrical as illustrated in Figure 2.3, i.e.:

$$\min\{\Delta p_i\}=-\Delta p_i^{\max} \tag{2.33}$$

$$\max\{\Delta p_i\}=\Delta p_i^{\max} \tag{2.34}$$

Then,

$$\min\left\{s_{p_i}^{f}\cdot\Delta p_i\right\}=-\left|s_{p_i}^{f}\cdot\Delta p_i^{\max}\right| \tag{2.35}$$

$$\max\left\{s_{p_i}^{f}\cdot\Delta p_i\right\}=\left|s_{p_i}^{f}\cdot\Delta p_i^{\max}\right| \tag{2.36}$$

where $\left|s_{p_i}^{f}\cdot\Delta p_i^{\max}\right|$ is the absolute value of $s_{p_i}^{f}\cdot\Delta p_i^{\max}$. With equations (2.35) and (2.36), equations (2.31) and (2.32) can be rewritten as:

$$f_{\min}=f_0-\sum_{i=1}^{n}\left|s_{p_i}^{f}\cdot\Delta p_i^{\max}\right| \tag{2.37}$$

and

$$f_{\max}=f_0+\sum_{i=1}^{n}\left|s_{p_i}^{f}\cdot\Delta p_i^{\max}\right|. \tag{2.38}$$

In the case the statistical data for $\Delta p_i (i=1,2,\cdots,n)$ is available, the following discussion holds. Assume that $\Delta p_i (i=1,2,\cdots,n)$ are mutually independent, and their corresponding mean values are $\mu_i (i=1,2,\cdots,n)$ and variances are $\sigma_i^2 (i=1,2,\cdots,n)$ respectively. By using equation (2.26), the mean value and variance of Δf can be derived to be [Pee93]:

$$\mu_{\Delta f}=\sum_{i=1}^{n}(s_{p_i}^{f}\cdot\mu_i) \tag{2.39}$$

$$\sigma_{\Delta f}^2=\sum_{i=1}^{n}((s_{p_i}^{f})^2\cdot\sigma_i^2) \tag{2.40}$$

Then $f_{\min}$ and $f_{\max}$ defined in equation (2.27) and (2.28) can be estimated with the following equations:

$$f_{\min} = f_0 - c \cdot \sigma_{\Delta f} \tag{2.41}$$

$$f_{\max} = f_0 + c \cdot \sigma_{\Delta f} \tag{2.42}$$

where c is a constant and dependent on the required confidence of the estimation. The larger value c has, the bigger confidence the estimation has.

On the other hand, the tolerance box is defined as the band containing all the possible circuit output performances when the parameters are changed with all the possible values. Therefore, the tolerance box can be expressed with the band $[f_{\min}, f_{\max}]$, where $f_{\min}$ and $f_{\max}$ are defined in equation (2.27) and (2.28) respectively.

During the sensitivity analysis, it is not easy to calculate the partial derivative as required in equation (2.24). The approximate value for the sensitivity can be calculated with the following equation [Yoo98, Ham93]:

$$s_{p_i}^{f} = \frac{\Delta f}{\Delta p_i} \tag{2.43}$$

where Δp_i is the deviation for parameter p_i and Δf is the corresponding performance deviation from the nominal point. Since normally the deviation of the parameter also has a certain range or probability distribution, the sensitivity can be calculated with the following equation [Fan02]:

$$s_{p_i}^{f} = \frac{\dfrac{\Delta f_1}{\min\{\Delta p_i\}} + \dfrac{\Delta f_2}{\max\{\Delta p_i\}}}{2} \tag{2.44}$$

where $\min\{\Delta p_i\}$ and $\max\{\Delta p_i\}$ are the minimal value and maximal value of the parameter deviations respectively, and Δf_1 and Δf_2 are the corresponding circuit output deviations. For instance, if the 3σ value of a Gaussian distribution is assumed for the maximal parameter deviation, then $\min\{\Delta p_i\} = -3\sigma$ and $\max\{\Delta p_i\} = 3\sigma$. In the example presented in this chapter, equation (2.44) is the ***final*** equation used to calculate the circuit output performance sensitivity with respect to the parameter deviations.

2.3.3 Propagation algorithm

In the previous two subsections, the sensitivity-analysis based tolerance-box generation for each stand-alone core and the mathematical model of the parameter deviations in different cores has been discussed. Based on the discussion in these two subsections, the final tolerance-box propagation algorithm is proposed in this subsection.

In order to make propagation possible, the concept of the conventional parameter sensitivity is extended to the *core sensitivity* first. The *core sensitivity* is defined as the deviation of the circuit performance with regard to the deviation of the core input signal. For instance, for core A, the *core sensitivity* can be defined as [Fan02]:

$$s_x^A \overset{def}{=} \frac{\Delta y_x}{\Delta x} \tag{2.45}$$

where Δx is the small deviation of the core input signal x around the nominal value x_0 and Δy_x is the corresponding deviation of the output y around the nominal output value y_0.

Based on the mathematical derivation presented in Appendix A, the final output deviation of the whole system under all the parameter deviations can be derived from equation (A.14) as:

$$\Delta w = \sum_{i=1}^{n} \left(s_{G,p_i}^D + s_z^D \cdot s_{G,p_i}^C + s_z^D \cdot s_y^C \cdot s_{G,p_i}^A + s_z^D \cdot s_v^C \cdot s_{G,p_i}^B \right) \cdot \Delta p_i^G$$

$$+ \sum_{i=1}^{n_A} (s_z^D \cdot s_y^C \cdot s_{p_i}^A \cdot \Delta p_i^A) + \sum_{i=1}^{n_B} (s_z^D \cdot s_v^C \cdot s_{p_i}^B \cdot \Delta p_i^B) + \sum_{i=1}^{n_C} (s_z^D \cdot s_{p_i}^C \cdot \Delta p_i^C)$$

$$+ \sum_{i=1}^{n_D} (s_{p_i}^D \cdot \Delta p_i^D) + s_x^A \cdot s_y^C \cdot s_z^D \cdot \Delta x + s_u^B \cdot s_v^C \cdot s_z^D \cdot \Delta u \tag{2.46}$$

Here, $s_{G,p_i}^A (i=1,2,\cdots,n)$, $s_{G,p_i}^B (i=1,2,\cdots,n)$, $s_{G,p_i}^C (i=1,2,\cdots,n)$ and $s_{G,i}^D (i=1,2,\cdots,n)$ are the sensitivities with respect to the global parameter deviations. $s_{p_i}^A (i=1,2,\cdots,n_A)$, $s_{p_i}^B (i=1,2,\cdots,n_B)$ $s_{p_i}^C (i=1,2,\cdots,n_C)$ and $s_{p_i}^D (i=1,2,\cdots,n_C)$ are the sensitivities with respect to the corresponding local parameter deviations. s_x^A, s_u^B, s_y^C, s_v^C and s_z^D are the core sensitivities.

In the above equation, all the global parameter deviations are used for each core. Actually, the sensitivities with respect to some global parameter deviations are zero if these global parameter deviations have no influence on the output performance of the core. Therefore different cores could have a different number of effective global parameter deviations.

Since the parameter deviations $\Delta p_i^G (i=1,2,\cdots,n)$, $\Delta p_i^A (i=1,2,\cdots,n_A)$, $\Delta p_i^B (i=1,2,\cdots,n_B)$, $\Delta p_i^C (i=1,2,\cdots,n_C)$ and $\Delta p_i^D (i=1,2,\cdots,n_D)$ are mutually independent in the parameter-deviation model, they by using equations (2.29), (2.30) and (2.46) the minimum deviation of the output of the core D can be expressed as:

$$\min\{\Delta w\} = \sum_{i=1}^{n} \min\left\{ \left(s_{G,p_i}^D + s_z^D \cdot s_{G,p_i}^C + s_z^D \cdot s_y^C \cdot s_{G,p_i}^A + s_z^D \cdot s_v^C \cdot s_{G,p_i}^B \right) \cdot \Delta p_i^G \right\}$$

$$+ \sum_{i=1}^{n_A} \min\left\{ (s_z^D \cdot s_y^C \cdot s_{p_i}^A \cdot \Delta p_i^A) \right\} + \sum_{i=1}^{n_B} \min\left\{ (s_z^D \cdot s_v^C \cdot s_{p_i}^B \cdot \Delta p_i^B) \right\} + \sum_{i=1}^{n_C} \min\left\{ (s_z^D \cdot s_{p_i}^C \cdot \Delta p_i^C) \right\}$$

$$+\sum_{i=1}^{n_D}\min\left\{(s_{p_i}^{D}\cdot\Delta p_i^{D})\right\}+\min\left\{s_x^{A}\cdot s_y^{C}\cdot s_z^{D}\cdot\Delta x\right\}+\min\left\{s_u^{B}\cdot s_v^{C}\cdot s_z^{D}\cdot\Delta u\right\}\qquad(2.47)$$

Similarly, the maximal deviation can be expressed as:

$$\max\{\Delta w\}=\sum_{i=1}^{n}\max\left\{\left(s_{G,p_i}^{D}+s_z^{D}\cdot s_{G,p_i}^{C}+s_z^{D}\cdot s_y^{C}\cdot s_{G,p_i}^{A}+s_z^{D}\cdot s_v^{C}\cdot s_{G,p_i}^{B}\right)\cdot\Delta p_i^{G}\right\}$$

$$+\sum_{i=1}^{n_A}\max\left\{(s_z^{D}\cdot s_y^{C}\cdot s_{p_i}^{A}\cdot\Delta p_i^{A})\right\}+\sum_{i=1}^{n_B}\max\left\{(s_z^{D}\cdot s_v^{C}\cdot s_{p_i}^{B}\cdot\Delta p_i^{B})\right\}+\sum_{i=1}^{n_C}\max\left\{(s_z^{D}\cdot s_{p_i}^{C}\cdot\Delta p_i^{C})\right\}$$

$$+\sum_{i=1}^{n_D}\max\left\{(s_{p_i}^{D}\cdot\Delta p_i^{D})\right\}+\max\left\{s_x^{A}\cdot s_y^{C}\cdot s_z^{D}\cdot\Delta x\right\}+\max\left\{s_u^{B}\cdot s_v^{C}\cdot s_z^{D}\cdot\Delta u\right\}\qquad(2.48)$$

According to equations (2.27) and (2.28), the final tolerance box for the whole test path containing core A, core B, core C and core D can then be expressed by:

$$\left[w_0+\min\{\Delta w\},w_0+\max\{\Delta w\}\right]\qquad(2.49)$$

where w_0 is the nominal value of the circuit output performance with nominal parameters and $\min\{\Delta w\}$ and $\max\{\Delta w\}$ are calculated by equation (2.47) and (2.48) respectively.

For the case that all the global parameter deviations, local parameter deviations and input deviations are *symmetrical* as illustrated in Figure 2.3, they by using equations (2.37) and (2.38) the tolerance box given in equation (2.49) can be rewritten as:

$$\left[w_0-|\Delta w|,w_0+|\Delta w|\right]\qquad(2.50)$$

where

$$|\Delta w|=\sum_{i=1}^{n}\left|\left(s_{G,p_i}^{D}+s_z^{D}\cdot s_{G,p_i}^{C}+s_z^{D}\cdot s_y^{C}\cdot s_{G,p_i}^{A}+s_z^{D}\cdot s_v^{C}\cdot s_{G,p_i}^{B}\right)\cdot\Delta p_i^{G,\max}\right|$$

$$+\sum_{i=1}^{n_A}\left|s_z^{D}\cdot s_y^{C}\cdot s_{p_i}^{A}\cdot\Delta p_i^{A,\max}\right|+\sum_{i=1}^{n_B}\left|s_z^{D}\cdot s_v^{C}\cdot s_{p_i}^{B}\cdot\Delta p_i^{B,\max}\right|+\sum_{i=1}^{n_C}\left|s_z^{D}\cdot s_{p_i}^{C}\cdot\Delta p_i^{C,\max}\right|$$

$$+\sum_{i=1}^{n_D}\left|s_{p_i}^{D}\cdot\Delta p_i^{D,\max}\right|+\left|s_x^{A}\cdot s_y^{C}\cdot s_z^{D}\cdot\Delta x^{\max}\right|+\left|s_u^{B}\cdot s_v^{C}\cdot s_z^{D}\cdot\Delta u^{\max}\right|\qquad(2.51)$$

In the above equation, $\Delta p_i^{G,\max}(i=1,2,\cdots,n)$, $\Delta p_i^{A,\max}(i=1,2,\cdots,n_A)$, $\Delta p_i^{B,\max}(i=1,2,\cdots,n_B)$, $\Delta p_i^{C,\max}(i=1,2,\cdots,n_C)$, $\Delta p_i^{D,\max}(i=1,2,\cdots,n_D)$, $\Delta x^{\max}$ and $\Delta u^{\max}$ are the maximal deviation of the corresponding parameter deviations and the input deviations.

For the case that the mean values and the variances for the global parameter deviations, the local parameter deviations and the input deviations are known, with equation (2.39) and (2.46) the mean value of Δw can be expressed as:

$$\mu_{\Delta w} = \sum_{i=1}^{n}\left(s^D_{G,p_i} + s^D_z \cdot s^C_{G,p_i} + s^D_z \cdot s^C_y \cdot s^A_{G,p_i} + s^D_z \cdot s^C_v \cdot s^B_{G,p_i}\right) \cdot \mu^G_{p_i}$$

$$+ \sum_{i=1}^{n_A}(s^D_z \cdot s^C_y \cdot s^A_{p_i} \cdot \mu^A_{p_i}) + \sum_{i=1}^{n_B}(s^D_z \cdot s^C_v \cdot s^B_{p_i} \cdot \mu^B_{p_i}) + \sum_{i=1}^{n_C}(s^D_z \cdot s^C_{p_i} \cdot \mu^C_{p_i})$$

$$+ \sum_{i=1}^{n_D}(s^D_{p_i} \cdot \mu^D_{p_i}) + s^A_x \cdot s^C_y \cdot s^D_z \cdot \mu_{\Delta x} + s^B_u \cdot s^C_v \cdot s^D_z \cdot \mu_{\Delta u} \tag{2.52}$$

where $\mu^G_{p_i}$, $\mu^A_{p_i}$, $\mu^B_{p_i}$, $\mu^C_{p_i}$, $\mu^D_{p_i}$, $\mu_{\Delta x}$ and $\mu_{\Delta u}$ are the mean values for $\Delta p^G_i (i = 1,2,\cdots,n)$, $\Delta p^A_i (i = 1,2,\cdots,n_A)$, $\Delta p^B_i (i = 1,2,\cdots,n_B)$, $\Delta p^C_i (i = 1,2,\cdots,n_C)$, $\Delta p^D_i (i = 1,2,\cdots,n_D)$, Δx and Δu. Similarly, by using equation (2.40) and (2.46), the variance of Δw can be expressed as:

$$\sigma^2_{\Delta w} = \sum_{i=1}^{n}\left(s^D_{G,p_i} + s^D_z \cdot s^C_{G,p_i} + s^D_z \cdot s^C_y \cdot s^A_{G,p_i} + s^D_z \cdot s^C_v \cdot s^B_{G,p_i}\right)^2 \cdot (\sigma^G_{p_i})^2$$

$$+ \sum_{i=1}^{n_A}(s^D_z \cdot s^C_y \cdot s^A_{p_i})^2 \cdot (\sigma^A_{p_i})^2 + \sum_{i=1}^{n_B}(s^D_z \cdot s^C_v \cdot s^B_{p_i})^2 \cdot (\sigma^B_{p_i})^2 + \sum_{i=1}^{n_C}(s^D_z \cdot s^C_{p_i})^2 \cdot (\sigma^B_{p_i})^2$$

$$+ \sum_{i=1}^{n_D}(s^D_{p_i})^2 \cdot (\sigma^D_{p_i})^2 + (s^A_x \cdot s^C_y \cdot s^D_z)^2 \cdot \sigma^2_{\Delta x} + (s^B_u \cdot s^C_v \cdot s^D_z)^2 \cdot \sigma^2_{\Delta u} \tag{2.53}$$

where $\sigma^G_{p_i}$, $\sigma^A_{p_i}$, $\sigma^B_{p_i}$, $\sigma^C_{p_i}$, $\sigma^D_{p_i}$, $\sigma_{\Delta x}$ and $\sigma_{\Delta u}$ are the standard deviations for the global parameter deviations, the local parameter deviations and the input deviations of the cores. After the mean value and the standard deviation of Δw are obtained, the range of Δw can also be easily estimated under a certain estimation confidence. For instance, if a 3σ value is used for estimating the maximal deviation, then the corresponding lower and upper bounds of Δw can be expressed as:

$$[\mu_{\Delta w} - 3\sigma_{\Delta w}, \mu_{\Delta w} + 3\sigma_{\Delta w}] \tag{2.54}$$

In the above derivation, only four analogue cores are connected as shown in Figure 2.2. For the case that more than four cores are connected in different structure, the same propagation procedure can be employed to calculate the final tolerance-box and more details can be found in Appendix A.

2.4 Example circuit and simulation results

In order to illustrate and verify the viability of the proposed approach to generate the tolerance box for the whole test path consisting of two or more analogue cores, an experiment has been carried out on part of a locally designed system-on-chip.

2.4.1 Example circuit

The example circuit is a system-on-chip containing a Pre_amplifier, a continuous time filter (FILTER), a voltage-current converter (VIC) and other cores [Sta02b]. The number of transistors in the Pre_amplifier, the FILTER and the VIC are 16, 48 and 25, respectively and the analogue test path considered here is shown in Figure 2.4.

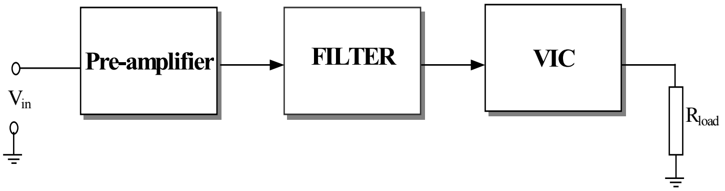


Figure 2.4: The example of the test path.

According to the statement in [Saa00], length, width and the gate oxide thickness are the main three geometrical parameters for MOSFET characterization. In our experiment, the 3σ value of a Gaussian distribution is used for the maximal deviations of the length and width of NMOS and PMOS transistor channels, and the gate oxide thickness. Hence, for all these parameters, the deviation range is $[-3\sigma,+3\sigma]$. Similar to the assumption made in [Shi99], 5% deviations are assumed for all the parameters in our example, i.e. $\pm 3\sigma$ is $\pm 5\%$ of the nominal value. How to apply our fast tolerance-box propagation approach to this example circuit is shown in the flow chart (Figure 2.5). More details about the application and the final results are presented in the following subsections.

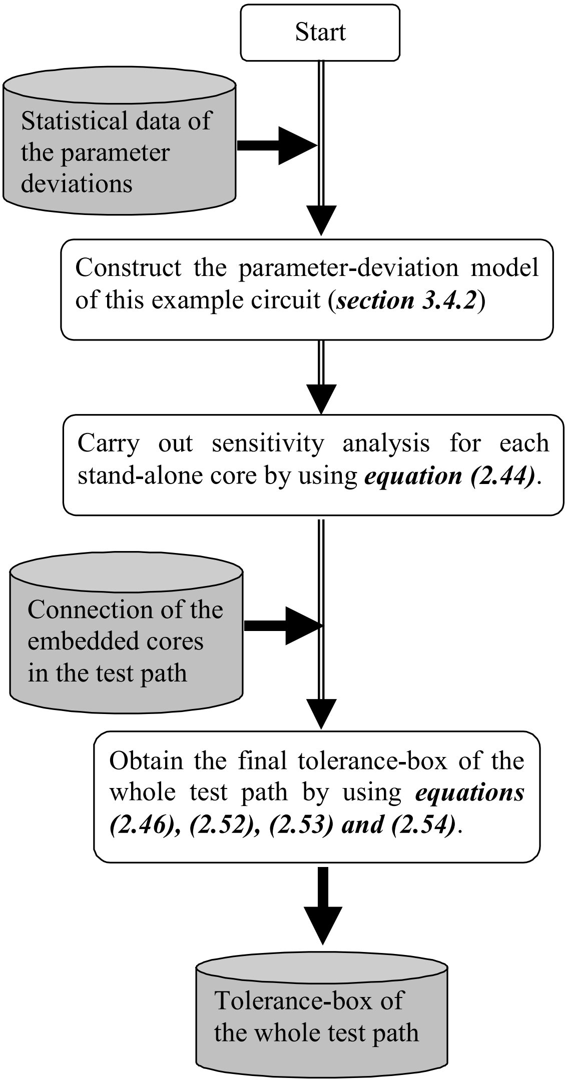

Start
Statistical data of the parameter deviations
Construct the parameter-deviation model of this example circuit (***section 3.4.2***)
Carry out sensitivity analysis for each stand-alone core by using ***equation (2.44)***.
Connection of the embedded cores in the test path
Obtain the final tolerance-box of the whole test path by using ***equations (2.46), (2.52), (2.53) and (2.54)***.
Tolerance-box of the whole test path

Figure 2.5: The flow diagram of the application of our tolerance-box propagation approach to the example circuit.

2.4.2 Parameter-deviation model of the example circuit

Based on the proposed parameter-deviation model shown in Figure 2.2, the parameter-deviation model for the example circuit can be constructed as shown in Figure 2.6 with global parameter deviations (ΔTox^G, ΔL^G and ΔW^G) and local parameter deviations ($\Delta Tox^L{}_{amp}$, $\Delta L^L{}_{amp}$, $\Delta W^L{}_{amp}$ etc.). The global parameter deviations and local parameter deviations in this model are made to be mutually independent. Their mean values and standard deviations are calculated with the procedure described below.

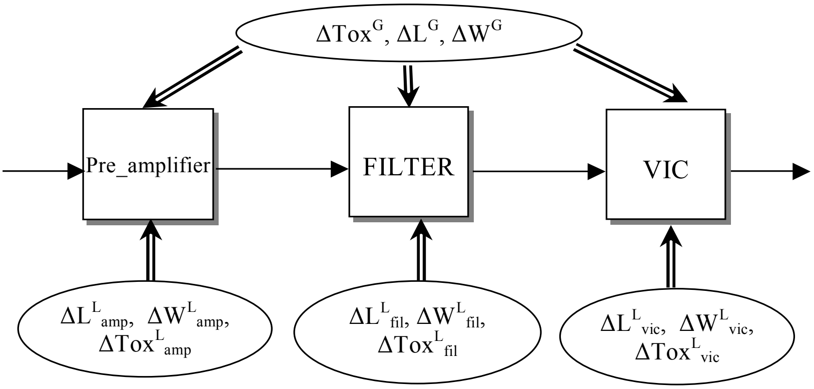

ΔTox^G, ΔL^G, ΔW^G
Pre_amplifier
FILTER
VIC
ΔL^L_amp, ΔW^L_amp, ΔTox^L_amp
ΔL^L_fil, ΔW^L_fil, ΔTox^L_fil
ΔL^L_vic, ΔW^L_vic, ΔTox^L_vic

Figure 2.6: The parameter-deviation model with global parameter deviations and local parameter deviations.

According to the available statistical data of the process and the layout of this example circuit, the mean value and the variance of the real channel length deviations (ΔL_{amp}, ΔL_{fil} and ΔL_{vic}) and width deviations (ΔW_{amp}, ΔW_{fil} and ΔW_{vic}) in these three cores are given as:

$$\mu_{\Delta L_{amp}} = \mu_{\Delta L_{fil}} = \mu_{\Delta L_{vic}} = \mu_{\Delta W_{amp}} = \mu_{\Delta W_{fil}} = \mu_{\Delta W_{vic}} = 0 \tag{2.55}$$

$$(\sigma_{\Delta L_{amp}})^2 = (\sigma_{\Delta L_{fil}})^2 = (\sigma_{\Delta L_{vic}})^2 = (\sigma_{\Delta W_{amp}})^2 = (\sigma_{\Delta W_{fil}})^2 = (\sigma_{\Delta W_{vic}})^2 = 2.8\times10^{-16} \tag{2.56}$$

The correlation coefficients among them are given as following:

$$\rho_{\Delta L_{amp},\Delta L_{fil}} = \rho_{\Delta W_{amp},\Delta W_{fil}} = 0.93 \tag{2.57}$$

$$\rho_{\Delta L_{fil},\Delta L_{vic}} = \rho_{\Delta W_{fil},\Delta W_{vic}} = 0.93 \tag{2.58}$$

$$\rho_{\Delta L_{amp},\Delta L_{vic}} = \rho_{\Delta W_{amp},\Delta W_{vic}} = 0.9 \tag{2.59}$$

For the deviation of the gate-oxide thickness (ΔTox_{amp}, ΔTox_{fil} and ΔTox_{vic}) in these three cores, the given mean values, variances and the correlation coefficients are:

$$\mu_{\Delta Tox_{amp}} = \mu_{\Delta Tox_{fil}} = \mu_{\Delta Tox_{vic}} = 0 \tag{2.60}$$

$$(\sigma_{\Delta Tox_{amp}})^2 = (\sigma_{\Delta Tox_{fil}})^2 = (\sigma_{\Delta Tox_{vic}})^2 = 8\times10^{-20} \tag{2.61}$$

$$\rho_{\Delta Tox_{amp},\Delta Tox_{fil}} = 0.9 \tag{2.62}$$

$$\rho_{\Delta Tox_{fil},\Delta Tox_{vic}} = 0.9 \tag{2.63}$$

$$\rho_{\Delta Tox_{amp},\Delta Tox_{vic}} = 0.85 \tag{2.64}$$

Based on these above-mentioned statistical data and the parameter-deviation model presented in Figure 2.6 and using the similar calculation procedure used for the example in subsection 2.3.1, the coefficients in equation (A.1) can be determined. Therefore, the parameter-deviation model with the global parameter deviations and local parameter deviations can be expressed as:

$$\Delta Tox_{amp} = 0.387 \cdot \Delta Tox_{amp}^{L} + 0.922 \cdot \Delta Tox^{G} \tag{2.65}$$

$$\Delta Tox_{fil} = 0.218 \cdot \Delta Tox_{fil}^{L} + 0.976 \cdot \Delta Tox^{G} \tag{2.66}$$

$$\Delta Tox_{vic} = 0.387 \cdot \Delta Tox_{vic}^{L} + 0.922 \cdot \Delta Tox^{G} \tag{2.67}$$

$$\Delta L_{amp} = 0.315 \cdot \Delta L_{amp}^{L} + 0.949 \cdot \Delta L^{G} \tag{2.68}$$

$$\Delta L_{fil} = 0.199 \cdot \Delta L_{fil}^{L} + 0.980 \cdot \Delta L^{G} \tag{2.69}$$

$$\Delta L_{vic} = 0.315 \cdot \Delta L_{vic}^{L} + 0.949 \cdot \Delta L^{G} \tag{2.70}$$

$$\Delta W_{amp} = 0.315 \cdot \Delta W_{amp}^{L} + 0.949 \cdot \Delta W^{G} \tag{2.71}$$

$$\Delta W_{fil} = 0.199 \cdot \Delta W_{fil}^{L} + 0.980 \cdot \Delta W^{G} \tag{2.72}$$

$$\Delta W_{vic} = 0.315 \cdot \Delta W_{vic}^{L} + 0.949 \cdot \Delta W^{G} \tag{2.73}$$

The mean values for all the global parameter deviations and local parameter deviations are zero. By using the calculation procedure described in subsection 2.3.1, their variance are calculated to be:

$$(\sigma_{\Delta L^{G}})^2 = (\sigma_{\Delta L_{amp}^{L}})^2 = (\sigma_{\Delta L_{fil}^{L}})^2 = (\sigma_{\Delta L_{vic}^{L}})^2 = 2.79 \times 10^{-16} \tag{2.74}$$

$$(\sigma_{\Delta W^{G}})^2 = (\sigma_{\Delta W_{amp}^{L}})^2 = (\sigma_{\Delta W_{fil}^{L}})^2 = (\sigma_{\Delta W_{vic}^{L}})^2 = 2.79 \times 10^{-16} \tag{2.75}$$

$$(\sigma_{\Delta Tox^{G}})^2 = (\sigma_{\Delta Tox_{amp}^{L}})^2 = (\sigma_{\Delta Tox_{fil}^{L}})^2 = (\sigma_{\Delta Tox_{vic}^{L}})^2 = 8 \times 10^{-20} \tag{2.76}$$

2.4.3 Simulation results

Figure 2.7 shows the final tolerance-box of the whole test path (Pre_amplifier, FILTER and VIC) based on the normal Monte Carlo simulation. Since in [Shi99] 10000 iterations are used to obtain the tolerance-box by using Monte Carlo simulations and 2000 iterations are used in [Yoo98], 2000 Monte Carlo simulations were performed at transistor level of the whole example circuit shown in Figure 2.4. Figure 2.8 shows the final tolerance box of the whole test path obtained with our new approach proposed in this chapter. In this approach, the sensitivity analysis is carried out for the stand-alone Pre_amplifier, FILTER and VIC respectively at first. Then, equations (2.52) and (2.53) are used to calculate the mean value and the standard variance of the output deviation under the parameter deviations. Afterwards, equation (2.54) is used to calculate the final tolerance box. In Figure

2.7 and Figure 2.8, the x-axis shows the input voltage and the y-axis shows the output current. For the whole input range (0~1.5V), 150 data points (i.e. every 0.01V) have been employed in both approaches. In general, the tolerance box shown in Figure 2.8 is very close to the tolerance box shown in Figure 2.7.

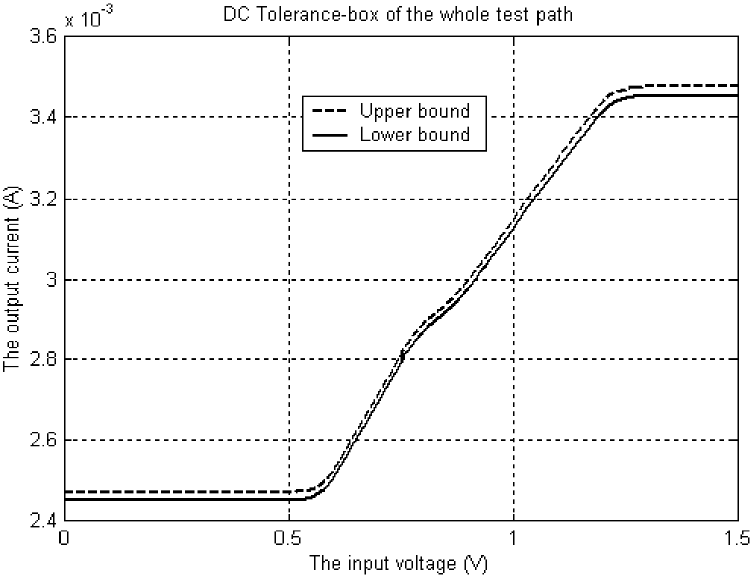


Figure 2.7: The tolerance box of the complete test path based on Monte Carlo simulation (iterations = 2000).

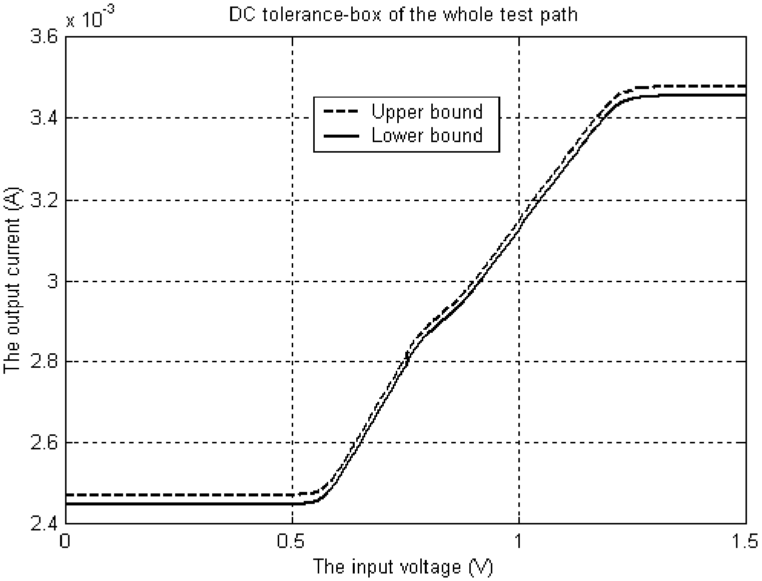


Figure 2.8: The final tolerance box of the complete test path obtained by using the new approach.

In order to observe the difference of the tolerance boxes obtained with these two approaches, the difference is shown in percentage in Figure 2.9. In the figure, the x-axis shows the input voltage and y-axis shows the difference in percentage. Both the difference of the lower bounds and upper bounds of the tolerance boxes obtained with different approaches are shown in the figure. The curves in Figure 2.9 show that the difference of the upper bounds and lower bounds are smaller than 1.36% and 1.34% respectively. In other words, the loss of accuracy is quite small for using our approach.

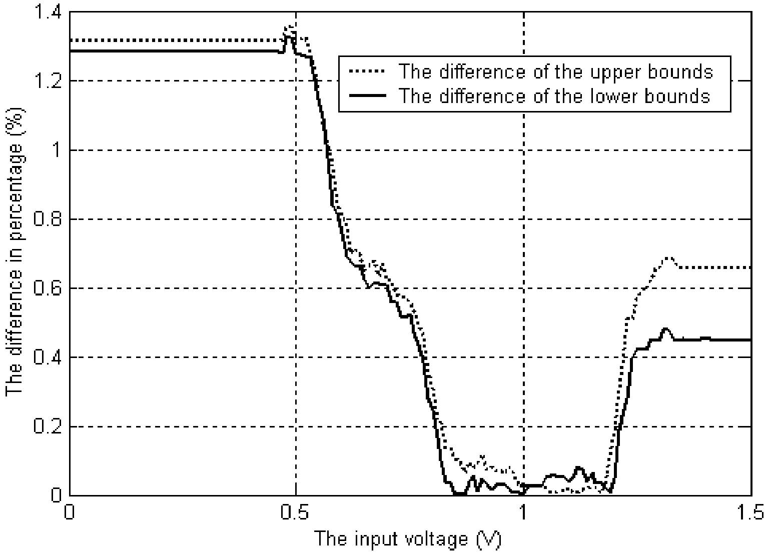


Figure 2.9: Comparison of the tolerance-boxes obtained with Monte Carlo simulation and our new approach.

In order to see the effectiveness of our approach, the CPU time used to obtain the tolerance box with different approaches are compared. At this moment, our proposed approach is implemented with HSPICE, C and MATLAB. Monte Carlo simulations are carried out with HSPICE. For both approaches, the computations have been carried out with the same HP workstation. The different CPU times used to get the results shown in Figure 2.7 and Figure 2.8 are listed in Table 2.1. When compared to the traditional Monte Carlo simulation approach, the proposed new approach is significantly faster (more than 140 times). This is the big advantage of our approach.

Table 2.1: CPU-times used for different approaches.

Our approach	Monte Carlo approach
Sensitivity analysis (Pre_amplifier): 2.6s Sensitivity analysis (FILTER): 10.4s Sensitivity analysis (VIC): 2.3s Nominal simulation: 1.1s Propagation: 1.2s (on PC) Total: **17.6s**	2000 iterations: **2520s**

However, our approach also has one limitation: it cannot provide an extremely accurate tolerance-box. Therefore, for the case that an extremely accurate tolerance box is required, more time-consuming methods, such as Monte Carlo simulation with a lot of iterations has to be used. The presented example shows that our approach can gain 140 times on the CPU time with the loss of less than 1.36% accuracy. Therefore our approach is very attractive for the case where the tolerance box is required to be estimated very quickly. For analogue fault simulation in mixed-signal SoC testing, the number of possible faults and the number of possible test vectors are very large because the number of transistors in the chip is very large. Since the tolerance boxes of the different faulty circuits are different, during analogue fault simulation a lot of tolerance boxes should be generated and it will take extensive CPU time. By using our approach, the tolerance box can be obtained very fast and a lot of CPU time can be reduced.

2.5 Conclusions

In this chapter, a new fast core-based tolerance-box generation and propagation in mixed-signal SoC consisting of embedded analogue cores is proposed based on sensitivity analysis. In our approach, sensitivity analysis for each stand-alone analogue core is carried out first. Then, by using the proposed propagation algorithm, the tolerance box for the complete test path can be obtained. As the original points in our approach, the concept of the sensitivity is extended to core sensitivity and a new model for different categories of parameter deviations is proposed. Finally, this approach has been applied to an example circuit and the simulation results show that our approach is about 140 times faster than the conventional Monte Carlo simulation approach while the loss in accuracy is smaller than 1.36%. Therefore, the proposed approach is very useful in the analogue fault simulation of mixed-signal SoC testing.

2.6 References

[Dol02] Dolphin Integration, "Jazz Line of analog and mixed-signal virtual components," http://www.dolphin.fr/flip/analog/analog_overview.html

[Fan01a] L. Fang and H. G. Kerkhoff, "Tolerance-box propagation in analogue testing," *Proc. of 7th IEEE International Mixed-Signal Testing Workshop*, Atlanta, U.S.A, pp. 156-160, June 2001.

[Fan01b] L. Fang and H. G. Kerkhoff, "Tolerance-box generation in mixed-signal SoC testing," *Proc. of ProRISC2001*, ISBN 9073461294, Veldhoven, the Netherlands, pp. 342-347, November 2001.

[Fan02] L. Fang and H. G. Kerkhoff, "Sensitivity analysis based tolerance-box generation and propagation in mixed-signal SoC testing," *Proc. of 7th IEEE European Test Workshop*, Corfu, Greece, pp. 255-260, May 2002.

[Fem99] N. Femia and G. Spagnuolo, “Genetic optimization of interval arithmetic-based worst case circuit tolerance analysis,” *IEEE Trans. Circuits and Systems—I: Fundamental Theory and Applications,* vol. 46, no. 12, pp. 1441-1456, December 1999.

[Ham93] N. B. Hamida and B. Kaminska, “Analog circuit testing based on sensitivity computation and new circuit modeling,” *Proc. of International Test Conference*, pp. 652-661, 1993.

[Kaa98] V. Kaal, "Structural test generation for analog macros", Ph.D. thesis, University of Twente, the Netherlands, 1998.

[Nag99] N. Nagi, “System-on-a-chip mixed-signal test: issues, current industry practices and future trends,” *Proc. of 5th International Mixed Signal Testing Workshop,* Whistler, BC, Canada, pp. 201-211, 1999.

[Ogr94] J. Ogrodzki, "Circuit simulation methods and algorithms," CRC Press, ISBN:0-8493-7894-X, 1994.

[Oze00] S. Ozev, I. Bayraktaroglu and A. Orailoglu, "Test synthesis for mixed-signal SOC paths," *Proc. of Design, Automation and Test in Europe (DATE) conference*, pp. 128-133, 2000.

[Pee93] P. Z. Peebles, “Probability random variables, and random signal principles,” 3rd edition, McGraw-Hill, ISBN:0-07-112782-8, 1993.

[Saa00] K. Saab, N. Ben-Hamida and B. Kaminska, “Parametric fault simulation and test vector generation,” *Proc. of Design, Automation and Test in Europe (DATE) conference*, pp. 650-656, March 2000.

[Shi99] C. J. R. Shi and M. W. Tian, "Simulation and sensitivity of linear analog circuits under parameter variation by robust interval analysis," *ACM Transactions on Design Automation of Electronic Systems*, vol. 4, no. 3, pp. 280-312, July 1999.

[Sta02b] M. Stancic, L. Fang, M. Weusthof, R. Tijink and H. Kerkhoff, “ A new test generation for embedded analogue cores in SoC,” *Proc. of International Test Conference (ITC)*, Baltimore, USA, pp. 861-869, 2002.

[Tia96] W. Tian, X. Ling and R. Liu, "Novel methods for circuit worst case tolerance analysis," *IEEE Trans. Circuits and Systems—I: Fundamental theory and applications*, vol. 43, no. 4, pp. 272-278, April 1996.

[Tia97] M. W. Tian and C. J. R. Shi, "Rapid frequency domain analog fault simulation under parameter tolerances*," Proc. of Design Automation Conference (DAC97)*, pp. 275-280, 1997.

[Tia00] M. W. Tian and C. J. Richard Shi, "Worst case tolerance analysis of linear analog circuits using sensitivity bands," *IEEE Trans. Circuits and Systems—I: Fundamental Theory and Applications,* vol. 47, no. 8, pp. 1138-1145, August 2000.

[Vla94] J. Vlach and K. Singhal, "Computer methods for circuit analysis and design," New York: Van Nostrand Reinhold, 1994.

[Yoo98] H. Yoon, P. Variyam, A. Chatterjee and N. Nagi, "Hierarchical statistical inference model for specification based testing of analog circuits," *Proc. of IEEE VLSI Test Symposium (VTS)*, pp. 145-150, 1998.

Chapter 3

Test-signal backtrace

In this chapter, a PID (Proportional Integral Derivative) feedback-loop based test-signal backtrace method is proposed. With this approach, the test input signals of the embedded analogue core under test can be propagated backwards to the primary chip inputs. Motivated by the concepts in control theory, a feedback loop with PID controller is constructed for test-signal backtracing in this technique. The computational effort of the proposed approach is quite low. The presented theoretical analysis, simulation and measurement results show the effectiveness of this new approach for test-signal backtracing in the time and frequency domain.

3.1 Introduction

Due to the complexity of mixed-signal SoC (System-on-a-Chip) nowadays, hierarchical approaches are required for test generation [Oze01]. In the hierarchical test-generation approach discussed in Chapter 1, the signal backtrace procedure is required in order to propagate backwards the test signal at the input of the embedded cores to the primary chip inputs. In general, the test-signal backtrace procedure can be defined in the following manner: test-signal backtrace is the procedure to determine a stimulus at the primary chip input which will produce the desired test signal at the input of the embedded core under test (ECUT), after propagating through intermediate cores or DfT (Design for Test) circuits which can potentially transform the input signal [Gom99, Hua99, Ram98, Voo99b]. The intermediate cores are the cores between the primary SoC inputs and the inputs of the embedded core under test. The DfT circuits are the extra circuits on the chip that provide the additional test access for the embedded core under test. The analogue wrapper structure proposed in [Fan00, Sta00], for instance, is an example of a DfT circuit for the testing of embedded cores in a SoC.

Backtrace procedures have played an important role in the development of digital ATPG (automatic test pattern generation). Such procedures find an assignment of primary chip input values that cause a desired logic value to appear on a specified line in the digital IC. For instance, PODEM [Goe81] uses path sensitisation and digital backtrace for test pattern generation of digital ICs. While digital backtrace procedures are relatively easy to implement with the help of Boolean mathematics, there is no generic approach for backtracing analogue signals in a mixed-signal IC. The main reason is that the analogue test signal is continuous and the respective input-output relationships of the components used in analogue circuits are often expressed by the nonlinear equations. Therefore the final input-output relationship of the analogue circuits can only be obtained by solving these nonlinear equations.

The main previous results on analogue test signal backtrace methods have been presented in [Gom99, Hua99, Oze01, Ram98, Ram99, Voo99a, Voo99b]. The method proposed in [Gom99] requires a lot of computation. In [Hua99], the DC specification is backwards propagated by using the derived I/O transfer table for each block. [Oze01] presented an approach for specification back propagation based on the mathematical equations of the behavioural model of the analogue blocks. In [Ram98], the authors use the MNA (Modified Node Analysis) formulation and swap the variables of the solution to determine an input signal that will produce the desired signal at the output. In this method, the desired waveform is required to be in the feasible signal space so that a practical solution exists. In [Ram99], the authors present an approach based on the usage of signal-flow graphs. However, this has limited applicability to linear circuits. The feedback-driven backtrace method proposed in [Voo99a, Voo99b] is quite simple. However, it has some disadvantages because only a proportional gain is used in the feedback. The steady-state error is large if the gain is small, and the loop becomes unstable if the gain is too high. According to the feedback-control theory [Ast95, Mor01], a PID controller can be used in the feedback to solve this problem. Motivated by this idea, a PID feedback-loop based test-signal backtrace method is proposed for mixed-signal SoC testing in this chapter.

The main steps of the proposed analogue test-signal backtrace approach are shown in Figure 3.1.

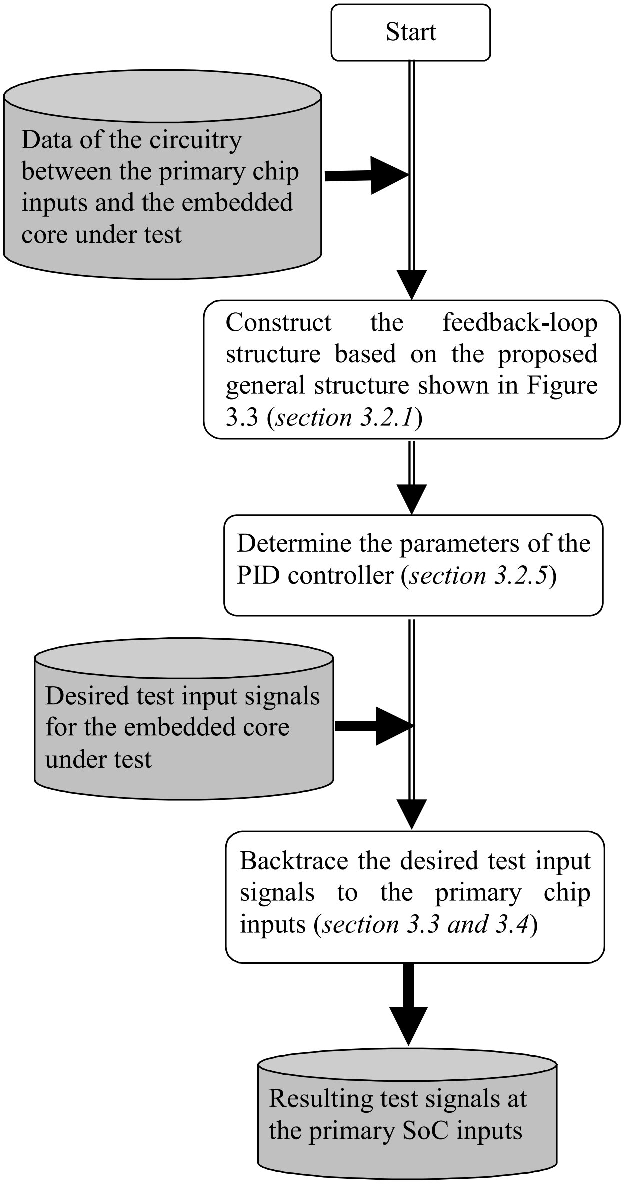

Start
Data of the circuitry between the primary chip inputs and the embedded core under test
Construct the feedback-loop structure based on the proposed general structure shown in Figure 3.3 (*section 3.2.1*)
Determine the parameters of the PID controller (*section 3.2.5*)
Desired test input signals for the embedded core under test
Backtrace the desired test input signals to the primary chip inputs (*section 3.3 and 3.4*)
Resulting test signals at the primary SoC inputs

Figure 3.1: The flow diagram of the proposed test-signal backtrace approach.

As illustrated in the figure, there are three main steps in applying this approach. The first step is to construct the feedback loop containing the PID controller and the circuitry between the primary chip inputs and the core under test. The second step is to determine the parameters of the PID controller used in the feedback loop. The last step is to backtrace the desired test input signals, in our case obtained by an ATPG method, from the inputs of the embedded core under test to the primary chip inputs. The details of the above-mentioned steps will be discussed in section 3.2, 3.3 and 3.4. With this proposed method, the backtraced test signal at the primary chip inputs can be easily obtained by performing normal circuit simulation.

3.2 PID feedback-loop based test-signal backtrace approach

In this section, the details of the PID feedback loop based test-signal backtrace approach are given, including the general structure, the theoretical background, the performance analysis and parameter tuning.

3.2.1 General structure

The general overview of the test path containing embedded analogue cores in mixed-signal SoC testing is illustrated in Figure 3.2. In this structure, ECUT is the embedded analogue core under test. MODULE 1 represents all the circuits between the primary chip input and the input of ECUT, and MODULE 2 represents all the circuits between the output of ECUT and the primary chip output. MODULE 1 and MODULE 2 can be one core or the combination of some cores or DfT circuits. As discussed in the chapter 1 regarding the hierarchical test generation for the mixed-signal SoC testing, the potential test vectors for the ECUT can be generated by ATPG. Then, a translation scheme is employed to translate the core-level test vectors to system-level test vectors. During the test vector translation, the test-signal backtrace procedure should be used since the input node of the embedded core under test is not directly accessible from the primary chip inputs. In other words, the test-signal backtrace procedure determines a stimulus at the primary chip input that will produce the desired test signal at the input of ECUT after propagating through MODULE 1.

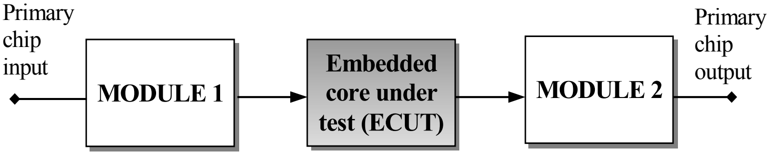


Figure 3.2: Illustration of the test path in the analogue part of mixed-signal SoC.

By using the structure of the analogue test path in mixed-signal SoC testing illustrated in Figure 3.2 , the proposed test-signal backtrace structure is presented in Figure 3.3. In this structure, the MODULE 1 and a PID controller are connected to construct a feedback loop. The ECUT is disconnected from MODULE 1 as shown in Figure 3.3. In order to emulate the effect of loading of the embedded core under test, the impedance Z_{load} is connected to the output of MODULE 1 and its value equals the input impedance of the core under test. Therefore, only MODULE 1, Z_{load} and the feedback loop components are considered in the test-signal backtracing simulation. The remaining part of the SoC is left out. During the backtrace procedure, one first applies the desired test input signals for the ECUT to the input of the feedback loop at point (A). Next, the simulation on the circuits as shown in

Figure 3.3 is carried out. Afterwards, at point (B), the required stimulus to be applied to the primary chip input is available.

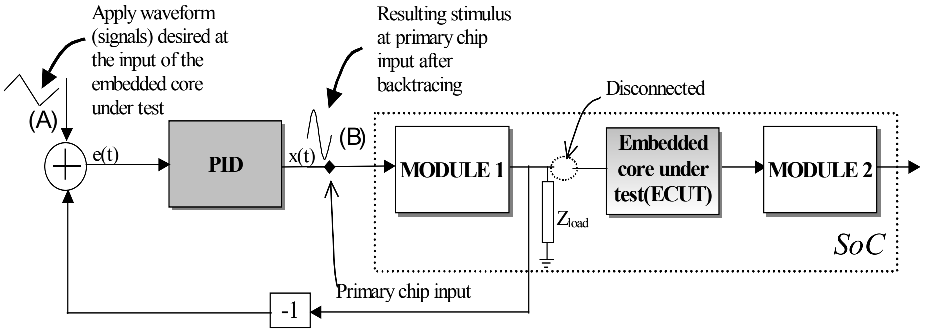

Apply waveform (signals) desired at the input of the embedded core under test
(A)
e(t)
PID
x(t)
(B)
Resulting stimulus at primary chip input after backtracing
Disconnected
MODULE 1
Embedded core under test(ECUT)
MODULE 2
Z_{load}
SoC
Primary chip input
-1

Figure 3.3: The proposed PID feedback-loop based signal backtrace structure.

Theoretically, the desired test input signals should be obtained at the input node of the ECUT if this resulting stimulus is applied to the primary chip input during testing. However, this does not hold in some special cases for some backtrace methods. For instance, for the method proposed in [Voo99b], in some cases, a small feedback gain has to be used in order to keep the feedback loop stable. In this case there will be a significant steady-state error between the desired test-input signal for the ECUT and the MODULE 1 output signal. However, this problem is solved in our proposed approach by using a PID controller in the feedback loop.

3.2.2 Theoretical background

From a mathematical point of view, the PID algorithm can be described as:

$$x(t) = K_P \cdot e(t) + K_I \cdot \int_0^t e(\tau)d\tau + K_D \cdot \frac{de(t)}{dt} \tag{3.1}$$

where $e(t)$ is the error between the desired signal and the output signal of MODULE 1, and $x(t)$ is the output signal of the PID controller. The factors K_P, K_I and K_D are the control parameters of the proportional action, integral action and derivative action respectively. Basically, the output signal of a PID controller is the sum of three terms: the P-term (which is proportional to the error), the I-term (which is proportional to the integral of the error) and the D-term (which is proportional to the derivative of the error). According to the PID feedback control theory [Ast95, Mor01], the integral term results in a control action that is proportional to the time integral of the error. This ensures that the final steady-state error becomes zero. The derivative term is proportional to the time derivative of the control error. This term allows the prediction of the future error and improves the closed-loop stability. Because the PID controller is used in the proposed signal backtrace approach, the new approach has a small state error and good stability.

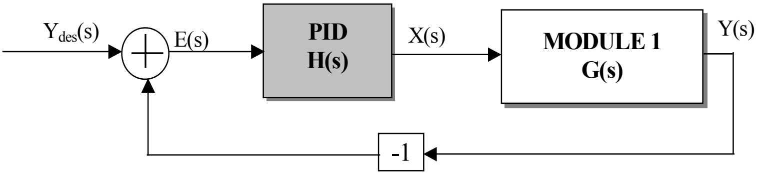


Figure 3.4: Basic structure of the signal backtrace implementation using PID feedback.

Figure 3.4 depicts the basic structure of the feedback loop for signal backtracing in terms of the transfer function. In the figure, $H(s)$ is the transfer function of the PID controller. According to equation (3.1), $H(s)$ can be expressed as:

$$H(s) = K_P + \frac{K_I}{s} + K_D \cdot s \tag{3.2}$$

where K_P, K_I and K_D are proportional gain, integral gain and differential gain respectively. The error $E(s)$ can be expressed as a function of the desired test signal $Y_{des}(s)$ and the MODULE 1 output signal $Y(s)$:

$$E(s) = Y_{des}(s) - Y(s) \tag{3.3}$$

Assume for simplicity that the transfer function of MODULE 1 is $G(s)$. Then the output $Y(s)$ of MODULE 1 can be expressed as:

$$Y(s) = X(s) \cdot G(s) = E(s) \cdot H(s) \cdot G(s) \tag{3.4}$$

By using equation (3.3) and (3.4), one can get:

$$Y(s) = Y_{des}(s) \cdot H(s) \cdot G(s) - Y(s) \cdot H(s) \cdot G(s) \tag{3.5}$$

Therefore,

$$\frac{Y(s)}{Y_{des}(s)} = \frac{H(s)G(s)}{1 + H(s) \cdot G(s)} = \frac{G(s)}{G(s) + \dfrac{1}{H(s)}}. \tag{3.6}$$

Using equation (3.2), the gain of the PID controller can be expressed as:

$$|H(j\omega)| = \left|K_P + \frac{K_I}{j\omega} + K_D \cdot (j\omega)\right| = \sqrt{K_P^2 + \left(K_D \cdot \omega - \frac{K_I}{\omega}\right)^2} \tag{3.7}$$

If the gain of the PID controller has a very large value, i.e.

$$|H(j\omega)| \to \infty \tag{3.8}$$

then one can obtain:

$$\frac{1}{H(s)} \to 0 \tag{3.9}$$

By combining equations (3.6) and (3.9), the following expression can be obtained:

$$\frac{Y(s)}{Y_{des}(s)} = \frac{G(s)}{G(s) + \frac{1}{H(s)}} \to 1 \tag{3.10}$$

According to equation (3.10), the output response $Y(s)$ of the circuit approaches the desired response $Y_{des}(s)$ as the gain of the PID controller defined in equation (3.7) becomes sufficiently large. Meanwhile, the input signal $X(s)$ provides the backtraced waveform. On the other hand, equation (3.8) also dictates the condition for using the PID feedback based backtrace approach. Actually, this condition can always be respected in practice if the parameters of the PID controller are determined with the dedicated parameter tuning technique, for instance Ziegler-Nichols method [Ste02]. The examples to be presented in sections 3.3 and 3.4 also confirm this point.

With regard to the computational effort of our approach, during the backtrace procedure simulation, the PID feedback structure as shown in Figure 3.3 can be represented using a high-level description language. This can be for instance Verilog-A or VHDL-AMS. Hence the computational effort will be low.

3.2.3 Steady-state error

The steady-state error for the feedback loop shown in Figure 3.3 is the error between the output signal of MODULE 1 and the desired signal at the input of the embedded core under test in the final steady state. For the test-signal backtrace procedure, the steady-state error should be as close to zero as possible.

We know from the control theory that there is normally a steady-state error only with proportional term in the controller [Ast95]. However, the main function of the integral term in the PID controller is to make sure that the steady-state error is zero. The reason is that with the integration action, a very small positive error will always lead to an increasing control signal, and a negative error will result in a decreasing control signal no matter how small the error is.

The following simple discussion can show that the steady-state error will always be zero with the integration action in the PID feedback loop. Assume that the system is in steady state with a constant control signal $x_0(t)$ and a constant error signal $e_0(t)$. By using equation (3.1), the control signal $x_0(t)$ can be expressed as:

$$x_0(t) = K_P \cdot e_0(t) + K_I \cdot e_0(t) \cdot t \tag{3.11}$$

As long as $e_0(t) \neq 0$, equation (3.11) clearly contradicts the assumption that the control signal $x_0(t)$ is constant. The PID controller used in Figure 3.3 will therefore always result in a zero steady-state error, i.e. the output signal of MODULE 1 is the same as the desired signal at the input of the embedded core under test in the steady state.

In order to show that this method can work using a normal Spice-like circuit simulator, the proof for the convergence of the proposed test-signal backtrace approach for nonlinear circuits with DC analysis is given in Appendix B.

3.2.4 Sensitivity to a disturbance signal

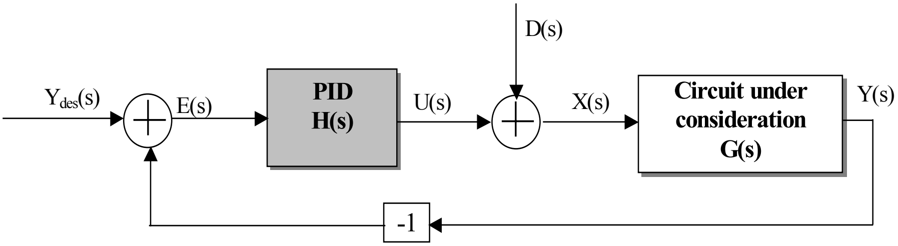


Figure 3.5: Backtrace structure with the disturbance signal D(s).

Let $D(s)$ be the transfer function of a disturbance or noise signal in Figure 3.5. If there is no feedback loop, then the influence of the disturbance signal $D(s)$ to the output of the circuit $Y(s)$ is always $G(s)\cdot D(s)$. However in the case that there is feedback, as shown in Figure 3.5, then:

$$Y(s) = \left[D(s) + U(s)\right]\cdot G(s) \tag{3.12}$$

and

$$U(s) = E(s)\cdot H(s) = H(s)\cdot\left[Y_{des}(s) - Y(s)\right]. \tag{3.13}$$

By using equation (3.12) and equation (3.13), one can obtain:

$$Y(s) = G(s)\cdot D(s) + G(s)\cdot H(s)\cdot Y_{des}(s) - G(s)\cdot H(s)\cdot Y(s) \tag{3.14}$$

With equation (3.14), $Y(s)$ can be expressed as:

$$Y(s) = \frac{\frac{D(s)}{H(s)}}{1+\frac{1}{G(s)\cdot H(s)}} + \frac{Y_{des}(s)}{1+\frac{1}{G(s)\cdot H(s)}}. \tag{3.15}$$

If the gain of the PID controller becomes very large, based on equation (3.9), then one can make the following approximations:

$$\frac{\frac{D(s)}{H(s)}}{1+\frac{1}{G(s)*H(s)}} \to 0 \tag{3.16}$$

$$\frac{Y_{des}(s)}{1+\frac{1}{G(s)*H(s)}} \rightarrow Y_{des}(s) \tag{3.17}$$

Expression (3.17) indicates that $Y(s) \rightarrow Y_{des}(s)$. Therefore, the influence of the disturbance signal to the output of the circuit becomes zero in the proposed signal backtrace structure. Hence, the influence of a disturbance or noise signal can be neglected.

3.2.5 Parameter tuning

Being a common controller, there are quite a few methods available on how to tune the PID controller, i.e. how to select the parameters K_P, K_I and K_D. Among them, the Ziegler-Nichols (Z-N) open-loop tuning method and close-loop tuning method are the most popular traditional methods used for determining the parameters of a PID controller. These methods are well described in [Ell00, Ste02, Una96]. Using this tuning technique, a stable feedback loop can be designed.

The first method is known as the continuous cycling method. It is a close-loop tuning technique. This method starts by assigning the integral and differential gains (K_I and K_D) to zero and then raising the proportional gain until a sustained oscillation takes place in the PID feedback loop. The value of the proportional gain at the point of instability is called K_{MAX}, while the frequency of the oscillation is labelled f_0. If the transfer function of the PID controller is expressed as:

$$H(s) = K_P \cdot \left[1 + s \cdot T_D + \frac{1}{s \cdot T_I}\right] \tag{3.18}$$

where $T_I = \frac{K_P}{K_I}$ and $T_D = \frac{K_D}{K_P}$ are known as the integral and derivative time respectively. Then the parameters of the PID controller can be set using the following rules:

$$K_P = 0.6 \cdot K_{MAX} \tag{3.19}$$

$$T_I = \frac{1}{2 \cdot f_0} \tag{3.20}$$

$$T_D = \frac{1}{8 \cdot f_0} \tag{3.21}$$

Finally, by using equation (3.2), equation (3.18) and the above three equations, the PID controller parameters can be set according to the following equations:

$$K_P = 0.6 \cdot K_{MAX} \tag{3.22}$$

$$K_I = 1.2 \cdot f_0 \cdot K_{MAX} \tag{3.23}$$

$$K_D = \frac{0.075 \cdot K_{MAX}}{f_0} \tag{3.24}$$

The alternative method is known as the process-reaction curve method. It is an open-loop step-response parameter tuning technique. In this method, the open loop (without the feedback) unit step response of the process, which is regard to as MODULE 1 in Figure 3.3 in our case, is measured. Normally, the curve has a form as shown in Figure 3.6. The response is approximated by a straight line (the thick dashed line), with a and L indicated as shown. This line can be constructed by finding the point of the curve with the biggest first-order differential value and subsequently drawing the tangent line at this point. The rules of Ziegler-Nichols tuning method can in this case be expressed by the following equations:

$$K_P = \frac{1.2}{a} \tag{3.25}$$

$$T_I = 2 \cdot L \tag{3.26}$$

$$T_D = 0.5 \cdot L \tag{3.27}$$

Finally, by using equation (3.2), equation (3.18) and the above three equations, the PID controller parameters can be set according to the following equations:

$$K_P = \frac{1.2}{a} \tag{3.28}$$

$$K_I = \frac{0.6}{a \cdot L} \tag{3.29}$$

$$K_D = \frac{0.6 \cdot L}{a} \tag{3.30}$$

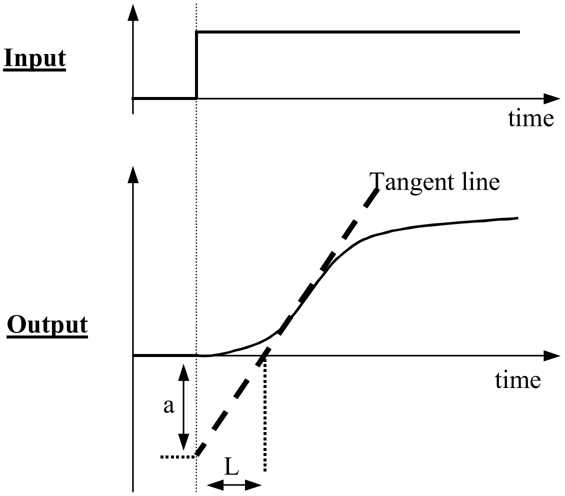


Figure 3.6: Graphical determination of the PID controller parameters from the open-loop step response.

3.3 Time and frequency domain simulation with the LPF

In this section, a simple active low-pass filter (LPF) is used to show how to apply our approach to signal backtracing in the time as well as frequency domain.

3.3.1 Example circuit and parameter tuning

The schematic of this active low-pass filter is shown in Figure 3.7, where $R_1 = 10K\Omega$, $R_2 = 10K\Omega$, $C_1 = 0.016\mu F$ and $C_2 = 0.032\mu F$. The power supply of the opamp is 3V. The AC response of this low-pass filter is shown in Figure 3.8.

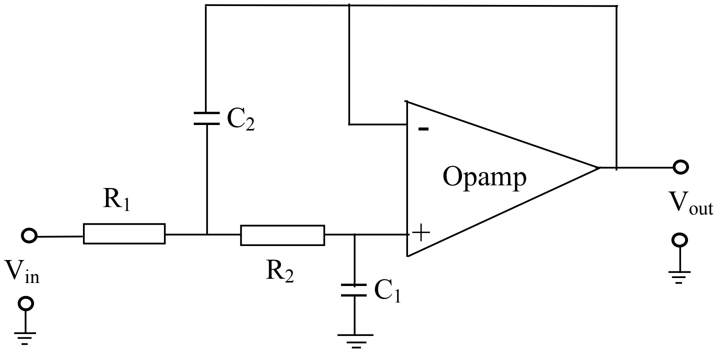


Figure 3.7: Schematic of the active low-pass filter.

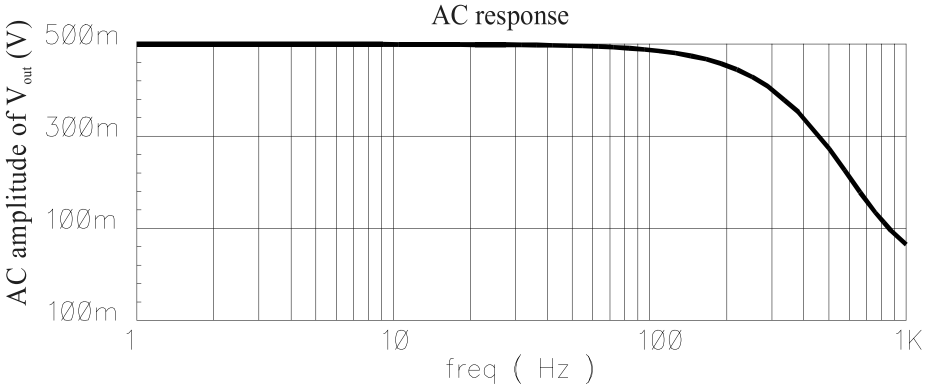


Figure 3.8: AC response of the low-pass filter. The AC amplitude of the sinusoidal input signal is 500mV and its DC offset is 1.3V.

In order to determine the input signal V_{in} of this low-pass filter if the desired output signal V_{out} is given, the signal backtrace method has to be used. Figure 3.9 shows the structure used for signal backtracing, i.e. replace the MODULE1 in Figure 3.3 with the low-pass filter shown in Figure 3.7. Before the signal-backtracing procedure can be applied, the parameters for the PID controller used in the structure shown in Figure 3.9 have to be determined. For this example, the open-loop step response Ziegler-Nichols method (process reaction curve method) is used to calculate the parameters of the PID controller.

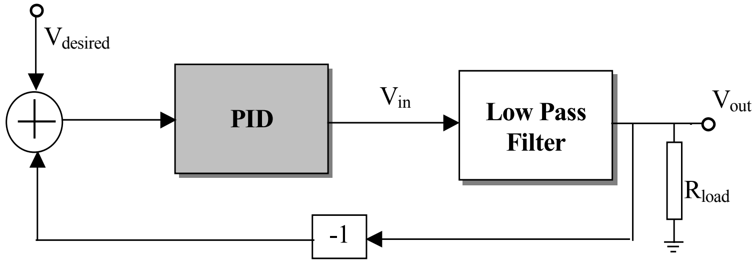


Figure 3.9: The PID feedback based signal backtrace structure for the low-pass filter.

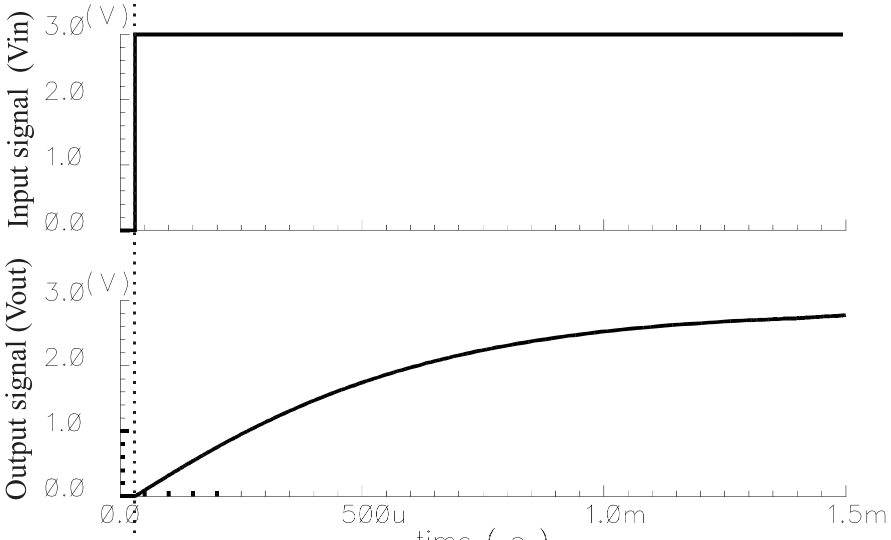


Figure 3.10: Step response of the low-pass filter.

If the step-pulse waveform is applied at the input of this low-pass filter (V$_{\text{in}}$), the output waveform at V$_{\text{out}}$ as shown in Figure 3.10 can be obtained by performing circuit simulation. With the step response shown in Figure 3.10 and by using the method described in the previous section, one can obtain:

$$a = 1.4 \times 10^{-4} \tag{3.31}$$

$$L = 3.02 \times 10^{-8} \tag{3.32}$$

Then, by using equation (3.28), (3.29) and (3.30), the parameters for the PID controller can be determined:

$$K_P = 8.571 \times 10^3 \tag{3.33}$$

$$K_I = 1.418 \times 10^{11} \tag{3.34}$$

$$K_D = 1.295 \times 10^{-4} \tag{3.35}$$

With these parameters of the PID controller and the structure shown in Figure 3.9, the signal backtrace procedure can be carried out. Additionally, the gain of the PID controller defined in equation (3.7) has a large value with the parameters determined in equations (3.33), (3.34) and (3.35). The condition defined in equation (3.8) can therefore be satisfied.

3.3.2 Signal backtrace in the time and frequency domain

After the determination of the parameters of the PID controller used in the signal backtrace structure shown in Figure 3.9, the simulation for the signal backtrace can be carried out. Figure 3.11 shows the transient-analysis simulation results. In Figure 3.11(a), the desired waveform ($V_{desired}$) at the output of the LPF is shown. Backtracing of this signal to the input of this low-pass filter provides the waveform to be applied at the input of the filter (V_{in}). This resulting input waveform is shown in Figure 3.11(b). If this resulting input waveform is applied to the input of this low-pass filter, the corresponding waveform at the output of this low-pass filter is shown in Figure 3.11(c).

In order to observe the difference between the desired signal and the resulting signal at the output of the low-pass filter, the error between them is shown in Figure 3.11(d). The curve in Figure 3.11(d) shows that the error between them is extremely small. The absolute vale of the error is smaller than 3×10^{-10} V. Since the smallest voltage value of the desired signal is 1V, the error is smaller than $(3\times10^{-8})\%$. Hence it can be concluded that the resulting waveform at the output of this low-pass filter closely matches the desired output waveform.

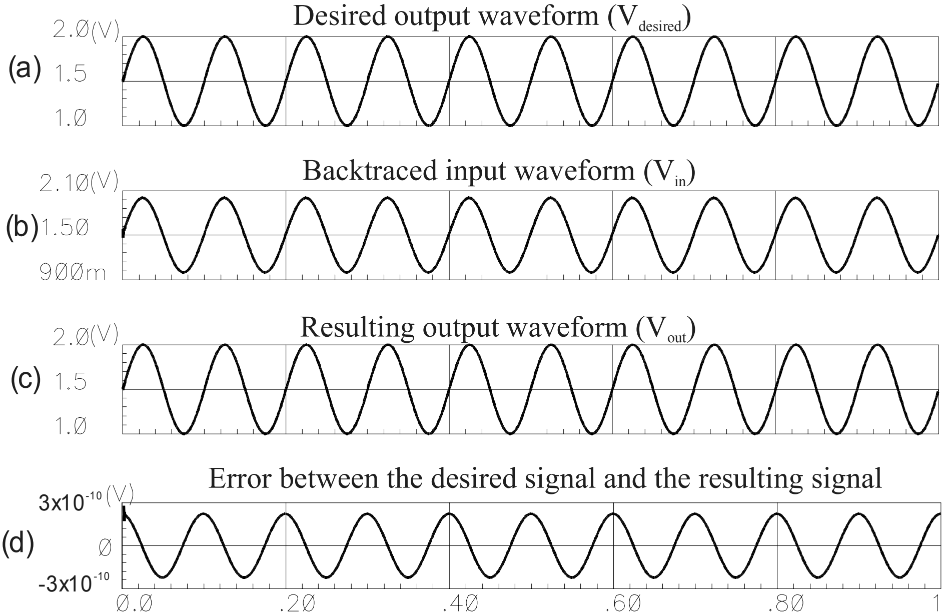


Figure 3.11: Transient analysis of the signal-backtrace procedure of the LPF.

Figure 3.12 shows the AC analysis simulation results for signal backtracing. Similar to Figure 3.11, the AC amplitude of the desired output waveform at V_{out} is shown in Figure 3.12(a). The AC amplitude of the backtraced input signal is shown in Figure 3.12(b) and the AC amplitude of the resulting waveform at the output of this low-pass filter is shown in Figure 3.12(c). Figure 3.12(d) shows the difference between the AC amplitude of the desired signal and the resulting signal.

The difference presented in Figure 3.12(d) shows that the error between them is extremely small. The absolute vale of the error is smaller than 2.5×10^{-9} V. Since the AC amplitude of the desired signal is 0.5V, the error is smaller than $(5\times10^{-7})\%$. This means that the resulting waveform at the output of the low-pass filter is very close to the desired waveform. Actually, as shown in Figure 3.12(b), the AC amplitude of the backtraced input waveform becomes larger if the frequency is increasing. The reason is that the gain of the low-pass filter becomes smaller as the frequency increases, as shown in Figure 3.8. The results in Figure 3.11 and Figure 3.12 show that the proposed PID feedback-loop based signal backtrace method works effectively in the time as well as frequency domain.

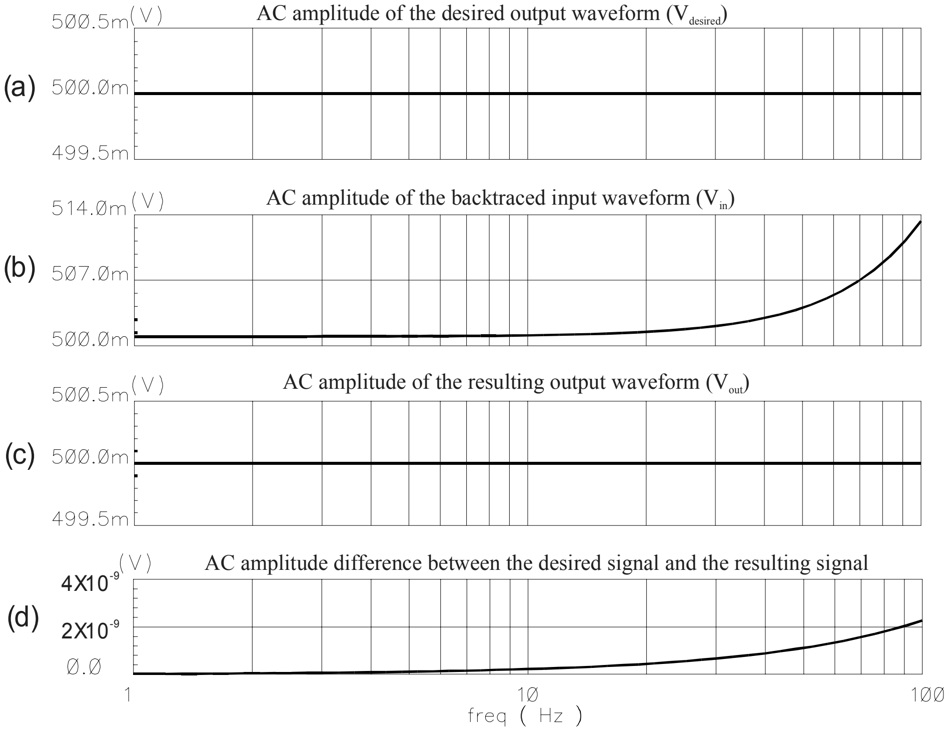


Figure 3.12: AC analysis of the signal-backtrace procedure of the low-pass filter.

3.4 Test-signal backtracing

In this section, the proposed signal-backtrace technique is used for real analogue test-signal backtracing in the testing of a mixed-signal SoC.

3.4.1 Structure of the test path

The example circuit in this section is a system-on-chip containing a pre-amplifier, a continuous-time filter (FILTER), a voltage-current converter (VIC) and other cores [Sta02b]. The analogue test path considered here is shown in Figure 3.13. In this example circuit, the FILTER and VIC are the cores under test. The FILTER has two output paths: the band-pass filter path and the low-pass filter path. The centre frequency of the band-pass output is 13.4kHz and its bandwidth is 12.43kHz (-3dB). The cutoff frequency of the low-pass output is 18kHz (-3dB). An operational amplifier with resistive feedback is used for the non-inverting Pre-amplifier (gain=2).

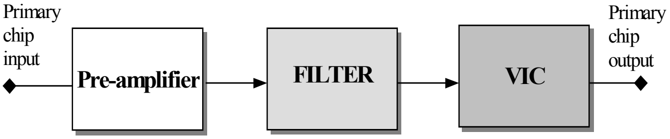


Figure 3.13: The general overview of the analogue test path of the example circuit.

3.4.2 Test-signal backtracing for the test signals of the FILTER

It is first assumed that the FILTER is the analogue embedded core under test. For the stand-alone FILTER, the testability-analysis based test-pattern generation approach is used to generate the potential test vectors [Sta02a, Sta02b, Sta03]. Among these potential test vectors, the test input signal for one particular fault is the combination of two sine waves, one 13.3 kHz signal (AC amplitude is 50mV and DC offset is 1.3V) and one 17.7 kHz signal (AC amplitude is 50mV and DC offset is 1.3V). Since the input of the FILTER cannot be directly accessed at the primary chip input, the backtrace procedure should be employed to find the stimulus at the primary chip input that can provide the required test input signal at the input node of the FILTER.

The structure of the backtrace approach used for our example circuit is shown in Figure 3.14.

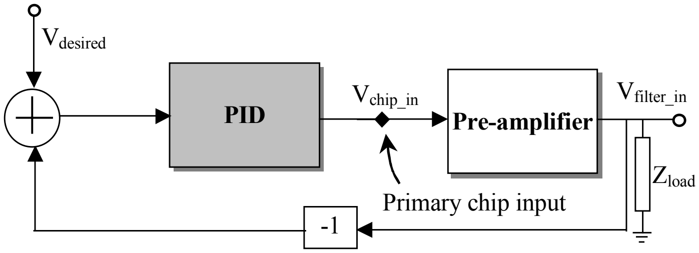


Figure 3.14: The test-signal backtrace structure for the test signal of the FILTER.

The load impedance Z_{load} in this structure, based on the specification of the FILTER, is the combination of one resistor (1MΩ) and one capacitor ($1pF$). In this structure, the desired test input signal for the FILTER, the combination of two sine signals as described in the previous paragraph, is applied to the input of the feedback loop ($V_{desired}$). In the simulation, the pre-amplifier is described at the transistor level and the rest of the backtrace structure is simulated using the high-level language Verilog-A as used in the Cadence design environment.

Before the test-signal backtrace structure shown in Figure 3.14 can be applied, the parameters of the PID controller used in this structure have to be determined. The open-loop step response Ziegler-Nichols method is used to determine the parameters of the PID controller. By applying the same method that is used for the signal-backtrace structure with the low pass filter, one can obtain:

$$a = 0.239 \tag{3.36}$$

$$L = 6.55 \times 10^{-7} \tag{3.37}$$

Then, by using equations (3.28), (3.29) and (3.30), the parameters of the PID controller can be determined:

$$K_P = 5.02 \tag{3.38}$$

$$K_I = 3.83 \times 10^{6} \tag{3.39}$$

$$K_D = 1.64 \times 10^{-6} \tag{3.40}$$

Next, the simulation for the test-signal backtracing can be carried out. In this step, the desired test signal is applied at point $V_{desired}$. Then, the simulation is carried out with the structure shown in Figure 3.14. Afterwards, the final input signal which should be applied to the primary chip input during the final testing can be obtained at node V_{chip_in}.

For this example, the combination of two sine waves, one 13.3kHz signal and one 17.7kHz signal, is applied to the node $V_{desired}$ as shown in Figure 3.14 and the transient simulation results are presented in Figure 3.15. Figure 3.15(a) shows the desired test input signal at the input node of the FILTER ($V_{desired}$) and Figure 3.15(b) depicts the resulting signal to be applied at the primary chip input (V_{chip_in}). Figure 3.15(c) shows the signal at the output of the pre-amplifier (V_{filter_in}). The goal of the backtrace procedure is to find the proper signal at the primary chip input (V_{chip_in}) in order to get the signal at the input of the FILTER (V_{filter_in}) as close as possible to the desired test input signal. The difference between the desired test signal and the resulting test signal at the input of the FILTER is depicted in Figure 3.15(d). As can been seen from Figure 3.15(d), the difference is very small except the short settling time ($< 10\mu s$). The settling time results from the fact that the PID feedback loop requires a very short time to reach the desired signal. This issue is not a problem in practice because the backtraced signal can be used in the final production testing after a certain time (longer than the settling time).

The results presented in Figure 3.15(d) indicate that the absolute value of the difference after the setting time is less than 0.001V. This means that the error is smaller than 0.1% because the desired test signal is larger than 1V. On the other hand, the specification of an analogue circuit allows some variations in the circuit characteristics due to the allowed deviations of the process parameters. Due to the allowed deviation of the specifications of the Pre_amplifier, the test-signal obtained at the input of the FILTER could differ for different chips during the final production testing. As a results of this, the final test signal obtained at the input of the FILTER could also differ from the simulated signal shown in Figure 3.15(c). Hence, during production testing, if the resulted chip input signal as shown in Figure 3.15(b) is applied to the primary chip input, then the test signal with an allowed slight difference from the signal shown in Figure 3.15(a) can be obtained at the input of the FILTER.

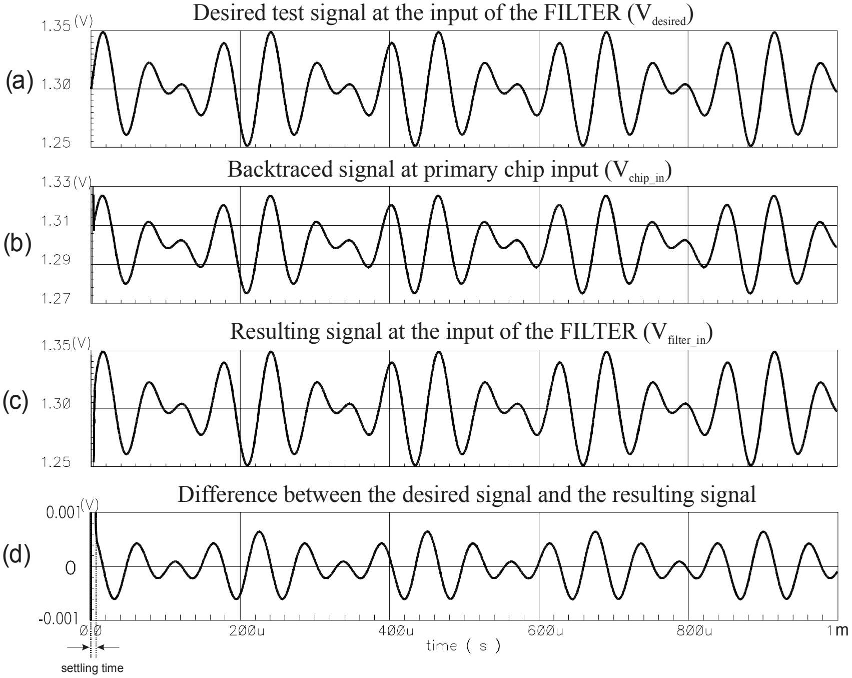


Figure 3.15: The results of the backtrace simulation for the test signal of the FILTER.

In order to evaluate the computational effort of our PID feedback-based backtrace approach, the CPU time for the simulation of the structure for signal backtracing and the standalone pre-amplifier with the same input signal are compared in Table 3.1. Both simulations are carried out with the same HP workstation and simulator. According to the data in Table 3.1, the computational effort for our approach is low and quite acceptable in practice.

Table 3.1: CPU time comparison.

Structure	CPU time for transient analysis
Pre-amplifier	0.84s
PID feedback with pre-amplifier	1.2s

3.4.3 Test-signal backtracing for the test signals of the VIC

Now it will be assumed that the VIC is considered as the analogue core under test. At first, the testability-analysis based test-pattern generation approach is used to generate the potential test vectors [Sta02a, Sta02b] for this stand-alone VIC. Afterwards, these core-level test signals have to be translated into the system-level test signal because the VIC is embedded in the SoC. Since the input of the VIC cannot be directly accessed at the primary chip input, the backtrace procedure should be employed to find the stimulus at the primary chip input that can provide the required test input signal at the input node of this VIC. The structure of the signal-backtrace approach used for backtracing the test signals of the VIC is illustrated in Figure 3.16.

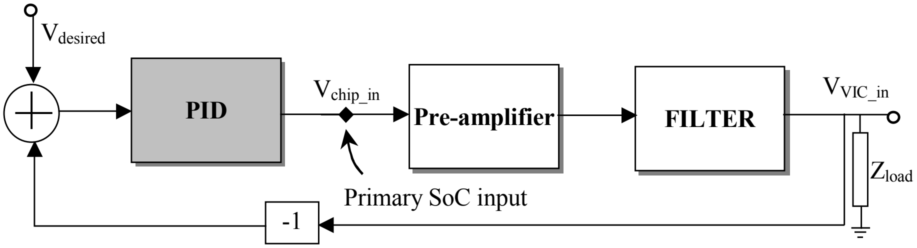


Figure 3.16: The test-signal backtrace structure for generating the test signals for the VIC.

The parameters of the PID controller used in the structure shown in Figure 3.16 can be determined in the same way as that used for the structure shown in Figure 3.14. Based on the step response of the combination of the Pre-amplifier and FILTER, the values of the parameters can be calculated:

$$a = 0.181 \tag{3.41}$$

$$L = 3.62 \times 10^{-5} \tag{3.42}$$

$$K_P = 16.6298 \tag{3.43}$$

$$K_I = 9.12 \times 10^{4} \tag{3.44}$$

$$K_D = 1.2 \times 10^{-4} \tag{3.45}$$

Without loss of any generality, an example test vector consisting of the combination of two sine waves, one 8.891kHz signal (AC amplitude is 25mV and DC offset is 1.3V) and one 11.857kHz signal (AC amplitude is 25mV and DC offset is 1.3V), is considered here. The backtracing simulation results are shown in Figure 3.17. The desired test input signal ($V_{desired}$) at the input node of the VIC is shown in Figure 3.17(a), the resulting signal to be applied at the primary chip input (V_{chip_in}) is depicted in Figure 3.17(b), while the signal at the output of the FILTER (V_{VIC_in}) is shown in Figure 3.17(c). The difference between the desired test signal and the resulting test signal at the input of the VIC is shown in Figure 3.17(d) in order to observe the accuracy of the proposed approach clearly. The results in Figure 3.17(d) show that the difference is very small (less than 0.004V). Moreover, the final test signal applied to the input of the SoC as shown in Figure 3.17(b) is a normal signal that can be generated with a conventional waveform generator. Therefore, our approach is effective for practical use.

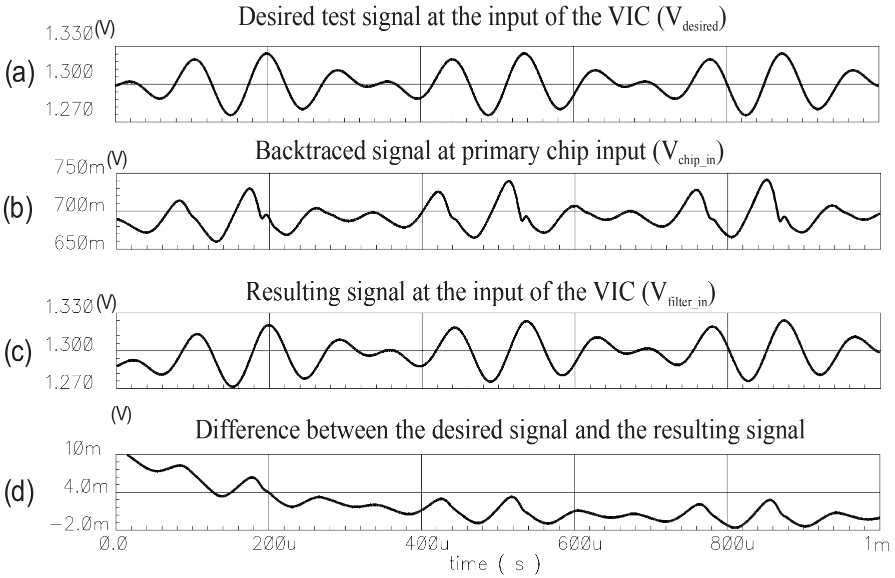


Figure 3.17: Backtrace simulation results for the test signal of the VIC.

3.4.4 Experimental results

In order to verify the basic principles of our new test-signal backtracing approach, a hardware experiment with a PCB (Printed Circuit Board) has been carried out. The vehicle used in the experiment has the same structure as the analogue test path shown in Figure 3.13, i.e. a Pre_amplifier, a FILTER and a VIC are connected in series. The Pre_amplifier is constructed by using an opamp with resistive feedback. It has the same AC gain and operational frequency range as the Pre_amplifier used in the SoC shown in Figure 3.13. The FILTER is built-up from four opamps and capacitors. It also has the same AC characteristic of the FILTER used in the SoC, except that it has a different DC offset. With regard to the VIC used in our experiment, it is built based on the negative feedback amplifier structure with the same AC characteristics as the VIC used in the SoC. In summarizing, the vehicle used in our hardware experiment has the same AC characteristics of the test path shown in Figure 3.13.

After the hardware is set up, the backtraced test signal described in the section 3.4.3 is applied at the input of the Pre_amplifier. Then the corresponding signal at the input of the VIC is measured. In more details, the test signal shown in Figure 3.17(b) is applied to the input of our emulation hardware set up (i.e. t he input of the Pre_amplifier) and the signal at the input of the VIC on the PCB is measured. The measurement results are shown in Figure 3.18.

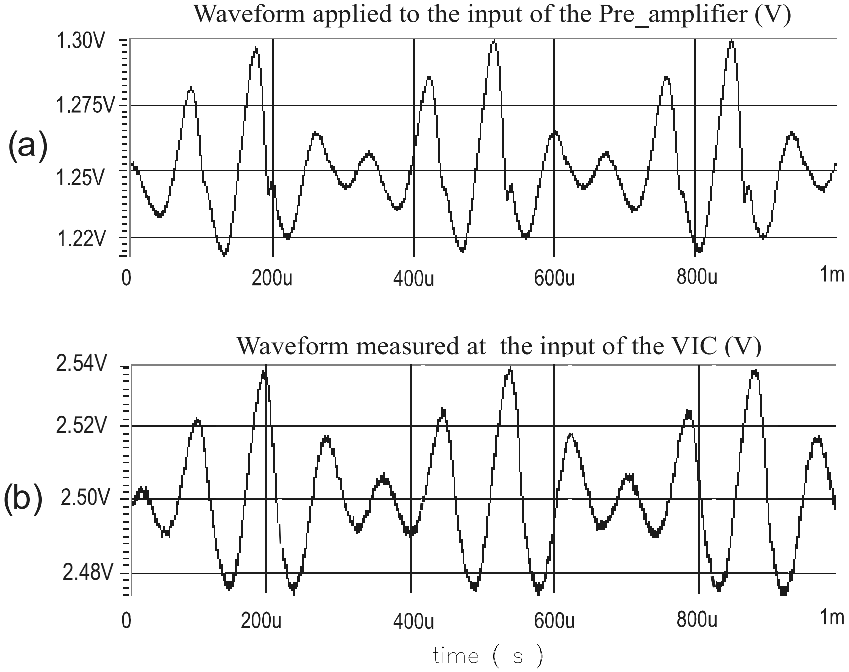


Figure 3.18: Measurement results: (a) the waveform applied to the input of our emulation hardware, (b) the waveform measured at the input of the VIC.

Figure 3.18(a) shows the waveform applied to the input of our emulation hardware, i.e. the input of the Pre_amplifier. This waveform is generated by an AWG (Arbitrary Waveform Generator) based on the backtraced signal shown in Figure 3.17(b). Due to the fact that the DC offset voltage of our emulation hardware is different from the DC offset of the Pre_amplifier used in the SoC, the waveform shown in Figure 3.18(a) has a different DC offset as compared to the signal shown in Figure 3.17(b). However, both waveforms have the same shape. The input signal at the input of the VIC is measured and shown in Figure 3.18(b). It is the real test signal applied to the VIC during final production testing.

On the other hand, Figure 3.17(a) shows the waveform that is expected to appear at the input of the VIC during the final production testing. As can be seen from Figure 3.18(b) and Figure 3.17(a), both waveforms have the same shape except that they have different DC offset voltage and the waveform shown in Figure 3.18(b) is more noisy due to the real physical noise of the hardware. Therefore, from the AC characteristics point of view, they match quite well. That means that our approach can successfully backtrace the analogue test signals from the input of embedded cores to the primary SoC inputs.

3.5 Conclusions

This chapter proposes a PID feedback based backtrace method for analogue test signals of embedded cores in mixed-signal SoCs. With this proposed method, the test input signal for the embedded core could be back traced to the primary chip inputs in the time as well as frequency domain. Moreover, the computational effort for the backtrace procedure is quite low since it is quite easy to implement the PID controller in a high-level description language during simulation. The presented simulation results show that the test-signals of the embedded cores can be successfully backtraced to the primary chip inputs.

3.6 References

[Ast95] K. Astrom and T. Hagglund, "PID controllers: theory, design and tuning," Instrument of Society America, ISBN:1-55617-516-7, USA, 1995.

[Ell00] G. Ellis, "Control system design guide," 2nd edition, ISBN:0-12-237465-7, Academic Press, San Diego, USA, 2000.

[Fan00] L. Fang, M. Stancic and H. G. Kerkhoff, "Mixed-signal core-based testing," *Inf. Proc. of European Test Workshop (ETW) 2000*, pp. 279-280, May 2000.

[Fan02] L. Fang and H. G. Kerkhoff, "PID feedback based backtrace method for mixed-signal SoC testing," *Proc. of 8th IEEE International Mixed-Signal Testing Workshop*, Switzerland, pp. 147-152, 2002.

[Goe81] P. Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits," *IEEE Trans. Computers*, C-30, pp. 215-222, 1981.

[Gom99] A. V. Gomes and A. Chatterjee, "Robust optimization based backtrace method for analog circuits," *Proc. of International Conference on Computer Aided Design (ICCAD)*, pp. 304-307, 1999.

[Hua99] J. L. Huang, C. Y. Pan and K. T. Cheng, "Specification back-propagation and its application to DC fault simulation for analog/mixed-signal circuits", *Proc. of VLSI Test Symposium (VTS),* pp. 220-225, 1999.

[Mor01] K. Morris, "Introduction to feedback control," Academic press, ISBN:0-12-507660-6, 2001.

[Oze01] S. Ozev and A. Orailoglu, "System-level test synthesis for mixed-signal designs," *IEEE Trans. Circuits and Systems—II: Analog and Digital Signal Processing*, vol. 48, no. 6, pp. 588-599, June 2001.

[Ram98] R. Ramadoss and M. L. Bushnell, "Test generation for analog circuits using partitioning and inverted system simulation," *Proc. of International Mixed-Signal Testing Workshop (IMSTW),* pp. 68-73, June 1998.

[Ram99] R. Ramadoss and M. L. Bushnell, "Test generation for mixed-signal devices using signal flow graphs," *Journal of Electronic Testing: Theory and Applications*, no. 14, pp. 189-205, 1999.

[Sta00] M. Stancic, L. Fang and H. G. Kerkhoff, "Detecting parametric faults in embedded cores of mixed-signal ICs" *Proc. of 6th IEEE International Mixed-Signal Testing Workshop (IMSTW),* pp. 20-21, June 2000.

[Sta02a] M. Stancic and H. G. Kerkhoff, "Test generation for mixed-signal circuits using testability analysis," *Inf. proceedings of the 7th European Test Workshop (ETW02)*, pp. 249-254, 2002.

[Sta02b] M. Stancic, L. Fang, M. Weusthof, R. Tijink and H. Kerkhoff, " A new test generation for embedded analogue cores in SoC," *Proc. of International Test Conference (ITC)*, pp. 861-869, 2002.

[Sta03] M. Stancic, “Test selection for analogue cores based on testability transfer function,” Ph.D thesis, University of Twente, Enschede, the Netherlands, 2003.

[Ste02] R. T. Stefani, B. Shahian, C.J.Savant and G. H. Hostetter, "Design of feedback control system," 4th edition, ISBN 0-19-514249-7, Oxford University Press, 2002.

[Una96] M. A. Unar, D. J. Murray-Smith and S. F. A. Shah, “Design and tuning of fixed structure PID controllers: a survey,” *technical report CSC-96016*, Centre for Systems and Control, University of Glasgow, 1996.

[Voo99a] R. Voorakaranam and A. Chatterjee, "Hierarchical Test Generation for analog Circuits Using Incremental Test Development," *Proc. of IEEE VLSI Test Symposium (VTS)*, pp. 296–301, April 1999.

[Voo99b] R. Voorakaranam and A. Chatterjee, "Feedback driven backtrace of analog signals and its application to circuit verification and test," *Proc. of 20th Anniversary Conference on Advanced Research in VLSI*, pp. 342-355, 1999.

Chapter 4

Mixed-level modelling for analogue fault simulation

As discussed in Chapter 1, one possible solution to reduce the massive analogue fault-simulation time is to use high-level modeling technique. In this approach, the fault-free blocks/cores are simulated with high-level models to reduce the CPU time during fault simulation. At the same time, the faulty block/core is simulated at transistor level for easy and accurate injection of the fault. A crucial issue for this method is the generation of suitable high-level models of the fault-free blocks/cores. Because the behaviour of the faulty block/core is unknown and it is possible that it functions totally different from the fault-free one, the high-level model used in fault simulations has different requirements as compared to high-level models used in the IC design field. A new mixed-level model structure with three stages is proposed in this chapter. The approach has been applied to an actual industrial chip and the fault simulations with the mixed-level model have been carried out. The results show that this kind of mixed-level model can work properly in a fault-simulation environment and effectively reduce the fault-simulation time.

4.1 Introduction

In the production of mixed-signal circuits, test can be a limiting factor contributing significantly to manufacturing cost [Mil98, Gar01]. Compared to digital testing, the lack of a systematic approach to mixed-signal testing development in the IC industry has resulted in a relatively poor fault coverage with respect to the bridging-fault model, high test costs, and long test-development time [Xin98].

In order to solve these problems, a structured test method called Defect Oriented Test (DOT) has been proposed for analogue circuits and studied in the recent years [Beu99, Sac98, Xin98]. In this approach, the manufacturing defects are modelled with certain fault models, which can be inserted into the circuit schematic to construct potential faulty circuits. For those faulty circuits, circuit simulations are carried out with test signals as inputs and the outputs are compared to the output of the fault-free circuit. If the difference between faulty circuit output and fault-free circuit output exceeds the pre-defined threshold, then the fault can be detected by the corresponding test pattern. In this way, the test patterns for final production testing can be determined. The procedure to perform the circuit simulation for all the faulty circuits with the entire potential test signals is called analogue fault simulation.

The presented results in [Beu99, Xin98, Sac98] show that DOT is a promising structural testing method. Case studies on two automotive mixed-signal ICs in [Xin98] show that, in addition to the conventional function-oriented tests, the fault coverage with respect to the bridging fault model can be significantly improved by some simple tests derived by using the DOT approach. However, the analogue fault simulation always requires extremely long simulation times because the number of possible faults in the fault list is large and analogue simulation at the transistor level is time-consuming. Therefore, one of the key issues hampering the widespread application of DOT in industry is how to reduce the fault simulation time.

Due to the complexity of the today's analogue and mixed-signal ICs, the whole chip is divided into several functional blocks/cores during design. One solution to reduce the fault-simulation time is to use high-level models for the faulty-free blocks in the fault simulation. This means that the fault-free blocks are simulated with high-level models to reduce the simulation time. Meanwhile, the faulty block is simulated at transistor level for easy and accurate injection of the fault into the fault-free netlist [Kaa98]. Crucial for this method is the generation of suitable high-level models of the fault-free analogue functional blocks.

There are already several methods available to generate high-level models in the IC design field [Cas91, Kru95, Roc96, Mos94]. However, the faulty block can have unexpected behaviour in the analogue fault simulation. It is possible that the conventional high-level models, which are generated based on fault-free surrounding blocks, cannot work properly in the fault-simulation environment. It is also possible that they can mask the fault, i.e. the originally detectable fault can sometimes not be detected due to the usage of the high-level model. As a result, the high-level model used in the fault-simulation environment has different requirements as compared to the high-level model normally used in the IC design field.

In this chapter, a new idea with regard to the mixed-level model used in fault simulation is proposed. An input stage, a functional stage and an output stage are used in the proposed model. The functional stage is an equation-based part to represent the function of the original fault-free block. For the input and output stage, the original transistor-level circuit for the input stage and output stage are reused. Another option is to use some electronic components, which can provide an isolation function and guarantee the proper operation of the complete model when preceded or succeeded by a faulty block.

In this chapter, our idea for the new mixed-level modelling has been applied to an analogue block in a real-life industrial mixed-signal chip. The mixed-level model of that block has been generated and the fault simulation with the mixed-level model of the block has been carried out. It uses an industrial mixed-signal fault simulation and test optimisation tool based on DOT. The fault simulation results are compared with the results from the simulations at transistor level. The presented results show that this kind of models work properly in the fault-simulation environment and speed up fault simulation significantly.

4.2 Defect Oriented Test (DOT)

The main reason for the necessity of production testing is the existence of unpredictable and uncontrollable phenomena in different steps of the manufacturing process. As the results of these imperfections in the manufacturing process, several types of defects can exist in manufactured ICs. The most common causes of defects in manufacturing are human errors, equipment failures, process instabilities, material instabilities, substrate in-homogeneity and lithographical spot defects [Sac98]. The effects of those defects can cause various types of faults, depending on the IC topology and the nature of the defect.

In order to evaluate the influence of the defects on the circuit performance, the fault caused by those defects should be electrically modelled. The most commonly used fault model for analogue circuits is the bridging-fault model [Eng00, Xin98, Beu99]. Depending on its value, it can degrade into a short or open fault. The modelling of manufacturing defects makes it possible to insert a fault model into the circuit netlist, run a simulation with a test signal as input and check whether the output varies from the output of the fault-free circuit, i.e. whether the fault will be detected by the test. In this way, the test patterns for the final production testing can be generated. Of course, in order to optimise the test patterns for low-test cost, knowledge on the probability of the occurrence of the faults is also required. The procedure to obtain a weighted fault list based on the process statistics and the IC layout is called fault extraction.

The probability of occurrence of the fault depends on the layout geometry and the characteristics of the process step related to the specific layer(s) where the fault occurs. To give an example, Figure 4.1 shows a two dimensional sketch of a conducting particle between two metal lines. For the sake of simplicity, the defect is assumed to be circular, which of course does not have to be the case in reality. This defect can be modelled as a bridging fault of resistance R_b, as shown in the figure. As can be seen from the figure, the probability of the occurrence of this bridging fault depends on the size of this particle and the distance between line_1 and line_2. For this example, the fault-free case can be modelled with infinite large R_b, meanwhile the short fault can be modelled by using a very small value of R_b.

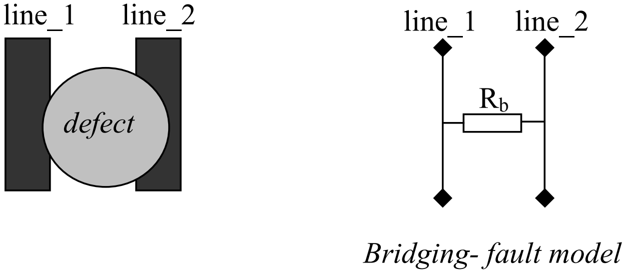


Figure 4.1: Example of a defect and its bridging fault model.

In general, the test-pattern generation based on defect-oriented testing consists of the following steps:

- *Fault extraction*. Based on the circuit layout, defect statistics, technology data and extraction rules, the weighted fault list is generated.

- *Fault simulation*. By inserting the faults into the fault-free circuit one by one (single fault model), all the faulty circuits can be constructed. Using test stimuli (DC, AC, transient etc.) as input, the simulation is carried out with the fault-free circuit and faulty circuits. Their outputs are then compared to see whether the fault can be detected. The test stimuli that can detect the faults in the fault list are subsequently listed.

- *Test-pattern optimisation*. Since the number of the faults that can be detected by each test vector is different and the cost of each test vector is also different (for instance the cost of a DC test vector is always cheaper than a AC test vector), the order of the test vectors should be optimised to obtain the highest fault coverage with the lowest test cost.

In our work, an industrial proprietary tool called DOTSS (Defect Oriented Testing Simulation System) is used for fault simulation. DOTSS is a fault-simulation tool that supports the DOT methodology for medium-sized analogue chips. Besides mixed-signal fault simulation it also includes test optimisation [Dot00, Xin98]. Starting from the layout of a design, DOTSS generates a list of weighted faults that may occur during IC production and inserts them one by one in the original netlist. DOTSS then simulates these faulty netlists together with a *user-defined* set of tests. It subsequently quantifies the fault coverage and optimises the test sequences with the user-defined test cost of individual tests. Thus it allows designers and test engineers to make a quick and accurate assessment of the test coverage of realistic process defects before any silicon is produced.

Because the number of faults in the fault list is usually very large, experience from the practitioners of fault simulation shows that the fault simulation of analogue circuits is a computationally intensive task taking very long CPU times for medium and large analogue blocks [Xin98]. One method to reduce fault-simulation time is to use a fault-sampling technique [Beu99, Eng00], which means that only some faults are selected from the complete fault list for fault simulation. There are several different criteria to apply fault sampling. As the fault extraction method discussed in this chapter grants weights (i.e. the possibility of the occurrence) to all faults, a fault selection approach based on the total weight of the selected faults is used. In this approach, the number of the selected faults should be minimized for a required total weight of the selected faults. Thus, the faults having the highest probability of occurrence (the biggest weight) should be selected in this approach.

Another promising method to reduce fault-simulation time is to use a high-level model during the fault simulation. By using this method, the fault-free blocks are simulated with a high-level model to reduce simulation time, while the faulty block is simulated at transistor level for easy and accurate injection of faults. In this chapter, this method has been studied with a new mixed-level model structure.

4.3 Mixed-level modelling for analogue fault simulation

In IC design and testing, there are already some methods to generate high-level models, for instance the models suggested in [Cas91, Ger96, Kru95, Roc96, Ros98]. Two typical high-level models are the so-called macro-model and behavioural model [Get93]. In the macro model, the existing components in the simulator, e.g. controlled sources, are used to represent the function of the block. For instance, in [Cas91], a macromodel for an operational amplifier is presented. On the other hand, the behavioural model is an equation-based model. It uses some mathematical functions to represent the function of the original block and has more flexibility. However, it requires a special simulation language that can handle the equation-based model, for instance VHDL-AMS [Chr99] and Verilog-A [Fit97]. Some examples of behavioural models can be found in [Ant95, Cur96, Ger96, Mos94, Ros98].

However, the faulty block can have unexpected functions [Miu00] in the analogue fault simulation. The output signals of the faulty block can be out of the range of the input signal for the succeeding block. The input impedance and output impedance of the faulty block are totally unknown and it is possible there is a large difference from those of the fault-free circuit. For instance, the output could be shorted to ground. Therefore, it is possible that the conventional macro-model or behavioural model, which is generated with fault-free surrounding blocks and within normal signal range, will not work properly in the fault-simulation environment. It is also possible that fault masking occurs, i.e. the detectable fault turns out to be undetectable during the fault simulation because of the usage of the high-level model. The high-level model used in fault simulation has therefore different requirements compared to the high-level model used in conventional IC design. One important requirement is that the model should work properly if there is a fault in the preceding or succeeding blocks. Unfortunately, this point is not considered in the published high-level modelling techniques used in analogue fault simulation, such as the high-level models proposed in [Dev96] and [Nai93].

On the other hand, if one looks at the structure of analogue functional blocks in an IC, there are always some special circuits for inputs and outputs to solve the input and output load matching problem, provide the input protection function and an output buffer function to enable a low output resistance for driving resistive and capacitive loads. Meanwhile, the internal central circuits in the block construct the actual analogue function of the block. For instance, the block diagram of a normal opamp shown in Figure 4.2 clearly shows that the opamp has three stages: input stage, functional stage (the high-gain stage) and output stage and the supporting circuitry (the compensation circuitry and the bias circuitry). Based on this analysis, a general high-level model structure for analogue blocks used in analogue fault simulation is proposed in Figure 4.3. It consists of three stages: input stage, functional stage and output stage [Fan01].

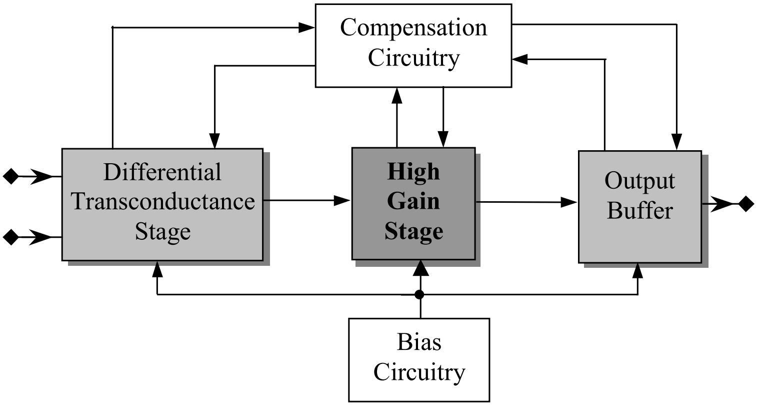


Figure 4.2: Block diagram of a general opamp [All02].

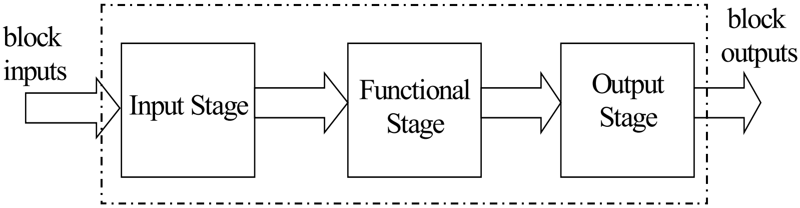


Figure 4.3: The proposed general structure of the mixed-level model of an analogue block.

In this structure, the input stage acts as an interface between electrical signals outside the block and the internal representation of the function by mathematical equations in the functional stage. The input electrical signals can be in the current or voltage domain. The input stage can be implemented with some existing electronic components, such as transistors, resistors and capacitors. The most attractive approach to implement the input stage is to reuse some of the originally designed circuit, as it will represent the original properties very well. In principle, it should represent the following properties of the circuit: input resistance, input capacitance, input bias current, input offset voltage, the bound of the input current and the limitations of the input voltage. In many analogue circuits, the input part of a circuit can be easily recognized and hence extracted and automatically reused.

The functional stage is the central part of the model and can be implemented using any kind of mathematical function. It is very flexible and can represent the functions of any kind of analogue circuits, for instance operational amplifier, ADC, DAC, mixer etc. For different analogue circuits, different mathematical approaches can be used. As examples we mention the Piece-Wise Linear (PWL) approximation [Dab99], multiple adaptive regressive splines [Dev96, Cha95] and neural networks [Mei96] to represent the functional operation of the block.

The output stage maps the results provided by the functional stage into the actual electrical environment of the analogue block. Like the input stage, it can also be implemented with some existing electronic components, such as transistors, resistors and capacitors. It can also reuse parts of the originally designed circuit. In principle, the following properties should be represented by the output stage: output resistance, output capacitance, output-voltage clipping and output-current clipping.

In most cases, the input stage and output stage have the function that some influences of the behaviour of the faulty circuit can be isolated from the functional stage. For instance, the input stage shown in Figure 4.4 has the voltage clamp function. With this voltage protection circuitry, the highest voltage that can be appeared at V_{in} is clamped to $V_{DD}+V_{th}$. Here, V_{th} is certain voltage value that depends on the design of the diode and the PMOS. During fault simulation, if a voltage that is higher than $V_{DD}+V_{th}$ appears at V_{in} due to the unexpected behaviour of the faulty circuit, then it will be clamped to $V_{DD}+V_{th}$ with this input stage. Hence, the function stage, which can only deal with the voltage not higher than $V_{DD}+V_{th}$, can work properly during the fault simulation.

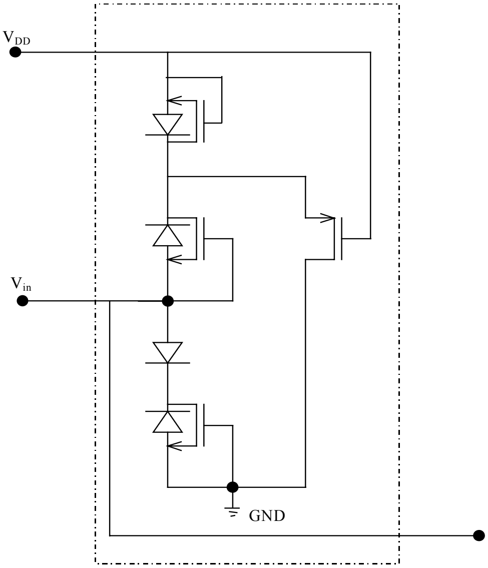


Figure 4.4: An example of input stage with voltage clamp function.

During the fault simulation, the functional stage can always work properly in our mixed-level model because the input stage and output stage are fault-free circuits and the functional stage only enables the communication with the input stage and output stage. In this way, the mixed-level model can be guaranteed to work properly during the fault simulation. Of course, the limits of the isolation depend on the input/output stage circuitry. For the example circuit shown in Figure 4.4, if the voltage is extremely high (e.g. 10kV), then the protection voltage clamp circuitry might be not work any more.

An important issue in order to derive the mixed-level model is how much time is required to construct it. If the design data are available during the model generation, then a model can be generated very fast because the input/output stages and the functional operation of the analogue block are always described very well by the design data. In general, a model can be generated using the following steps:

(a): With the help of the design data, separate the input stage and output stage for all inputs/outputs of the analogue block in the design.

(b): Copy the input stage and output stage into the mixed-level model.

(c): Based on the functional description of the block, an equation-based representation of the functional stage can be generated by using the existing behavioural modelling technique, for instance Piece-Wise Linear (PWL) approximation [Dab99], multiple adaptive regressive splines [Dev96, Cha95] etc.

(d): Combine the input stage, functional stage and output stage and subsequently verify the function of the complete mixed-level model.

4.4 Example circuit and the mixed-level model construction

In this section, based on the new mixed-level model structure shown in Figure 4.3, the mixed-level model for an analogue functional block in a real industrial mixed-signal chip is constructed. The functional operation of this mixed-level model is also verified by simulation.

4.4.1 An actual industrial design: the data transceiver

An actual industrial design, a data transceiver, has been used to evaluate our approach. This chip is used for low-speed data communication between the local devices connected to a bus as shown in Figure 4.5. The basic function of the chip is to receive data from the local bus via INOUT and send it to a local device (e.g. a microprocessor) via RXD, and receive the data from a local device via TXD and transmit it to the local bus via INOUT.

The general structure of the data transceiver is depicted in Figure 4.6. Besides the RXD and TXD pins, the chip has the additional pins to get/send the control signals from/to the local device. In order to reduce its power consumption, it can operate in different modes. If there is no request for receiving/transmitting data, then it operates in the sleeping mode, in which the chip consumes very little power. Due to its application and function, some parts of the chip, such as the RECEIVER part and the TRANSMITTER part, are analogue circuits. Some other parts of the chip, such as the CONTROL block, consist of digital circuits [Fan01].

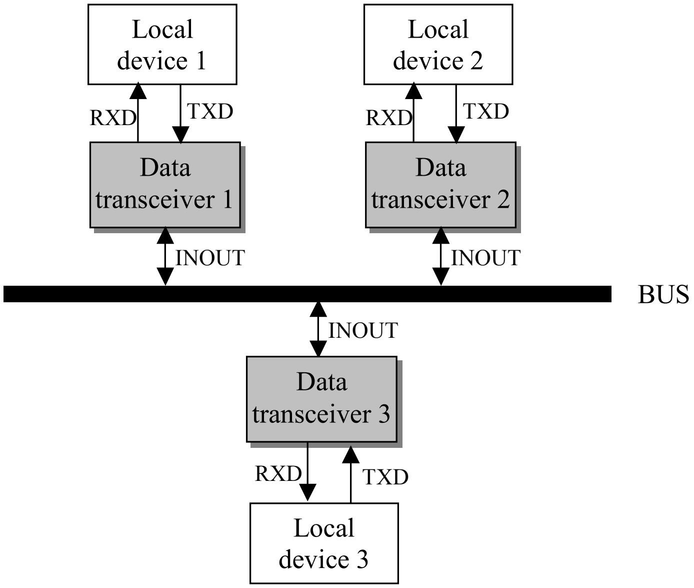


Figure 4.5: General structure of the typical application of the data transceiver.

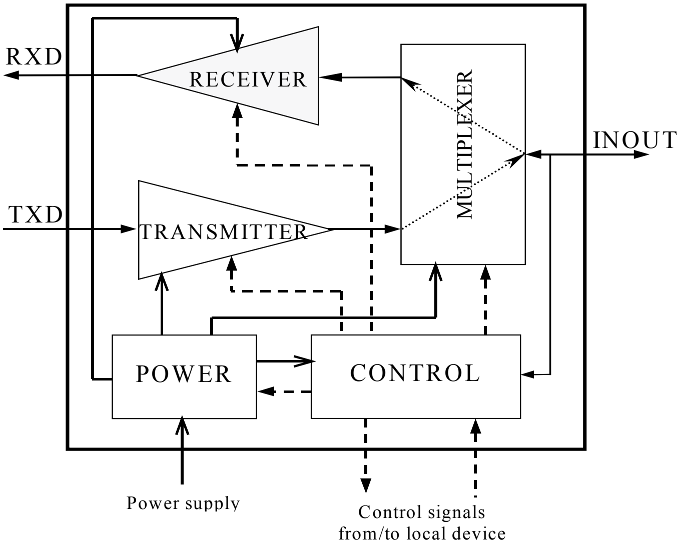


Figure 4.6: The architecture of the data transceiver.

For this chip, a fault list including 1233 bridging faults has been generated based on layout information, defect statistics data and the specified extraction rules. The primary test plan based on the chip specification includes 37 test vectors. The goal of our work is, based on DOT and the bridging fault model, to evaluate the fault coverage of the primary test plan, optimise the test order, delete some redundant test vectors and add some necessary test vectors. For one test vector in the test plan, the average simulation time is about 500s if the simulation is carried out at transistor level. Hence, the simulation time for all faults and all test vectors is around 6300 hours (about 260 days), which is unacceptable in practice.

In order to reduce this unacceptable fault simulation time, the fault-sampling method is applied to the fault list. Based on the weight of the faults, a reduced-length fault list with 87 faults is used in the final fault simulation, and the total weight of these 87 faults is 60% of the total weight of all the faults (1233 faults). The "60%" total weight is used here because, according to the experience with this kind of chip and manufacturing process, it is a good threshold for guaranteeing the confidence of the fault-simulation results.

4.4.2 The functional description of the RECEIVER block

The previously described mixed-level modelling approach has been applied to the RECEIVER block in the chip as indicated in Figure 4.6. The inputs/outputs of the RECEIVER block are shown in Figure 4.7. The descriptions of the input/outputs of the RECEIVER block are listed inTable 4.1.

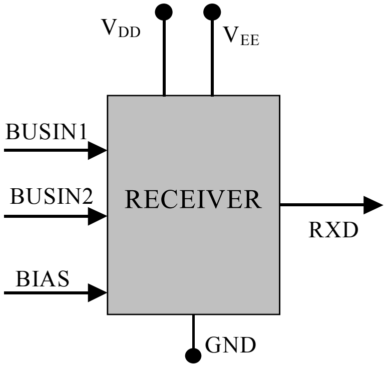


Figure 4.7: The inputs/outputs of the RECEIVER block.

Table 4.1: Description of inputs/outputs of the RECEIVER block

Name	Function
BUSIN1	BUS input signal after being filtered
BUSIN2	BUS input signal without being filtered
BIAS	Bias current supply
RXD	Output
VDD, VEE	Power supply
GND	Ground

The basic functionality of the RECEIVER block is a comparator, as shown in Figure 4.8, i.e. depending on the input signal voltage level, a different voltage-level signal is provided at the output. The function of the signal at BUSIN2 is to provide the extra information for the protection circuit in the RECEIVER block. Therefore, it is not listed in the normal functional description below.

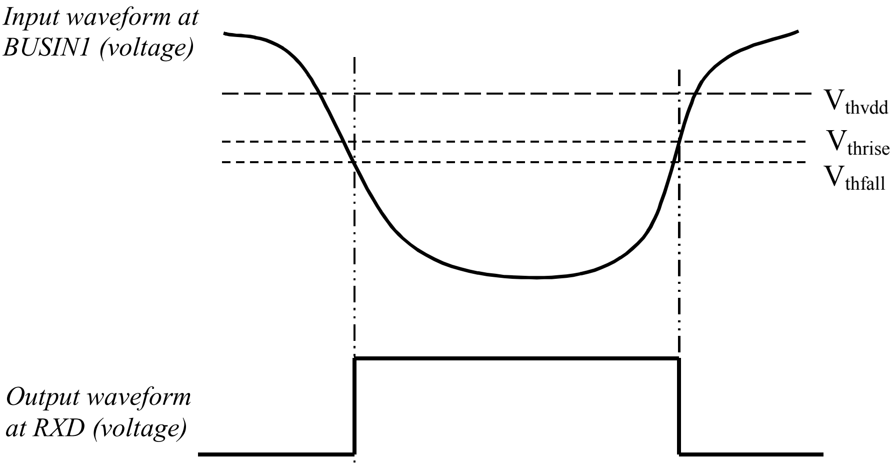


Figure 4.8: The basic relation between input and output of the RECEIVER block.

Table 4.2: The function of the RECEIVER block

State	**Input at BUSIN1 (V_{in})**	**Output at RXD (V_{out})**
Sleeping	V_{in} >= V_{thvdd}	V_{out} = logic low
Work State 1	V_{thvdd}> V_{in} >=V_{thfall} AND Vin is falling	V_{out} = logic low
Work State 2	V_{in}<V_{thfall} (V_{in} is falling) OR V_{in}<V_{thrise} (V_{in} is rising)	V_{out} = logic high
Work State 3	V_{thvdd} > V_{in} >= V_{thrise} AND V_{in} is rising.	V_{out} = logic low

More details with regard to the description of the function are given in Table 4.2. Actually, the basic function of the RECEIVER, shown in Figure 4.8 and Table 4.2, is a comparator with hysteresis because the threshold (V_{thfall}) for V_{in} falling is smaller than the threshold (V_{thrise}) for V_{in} rising. These three threshold voltages are specified in the design specifications of the RECEIVER as following:

$$V_{thvdd} = V_{DD} - 1.7$$

$$V_{thfall} = 0.41 * V_{DD}$$

$$V_{thrise} = 0.58 * V_{DD}$$

where V_{DD} is the voltage at VDD.

4.4.3 The mixed-level model of the RECEIVER block

Applying the idea of the general mixed-level model with three stages shown in Figure 4.3 to the RECEIVER block, the mixed-level model shown in Figure 4.9 can be generated with the following steps:

(a): Based on the design data including the schematic, the input stage and output stage of the RECEIVER block were identified.

(b): The originally designed input stage and output stage are reused in the mixed-level model shown in Figure 4.9.

(c): Based on the design specification and the functional operation of the RECEIVER block described in section 4.4.2, the equation-based functional stage was obtained.

(d): The input stage, functional stage and output stage were combined to obtain the complete mixed-level model.

With regard to the manpower used to generate this mixed-level model, it is three hours in total (one hour work of design engineer and two hours work of test engineer).

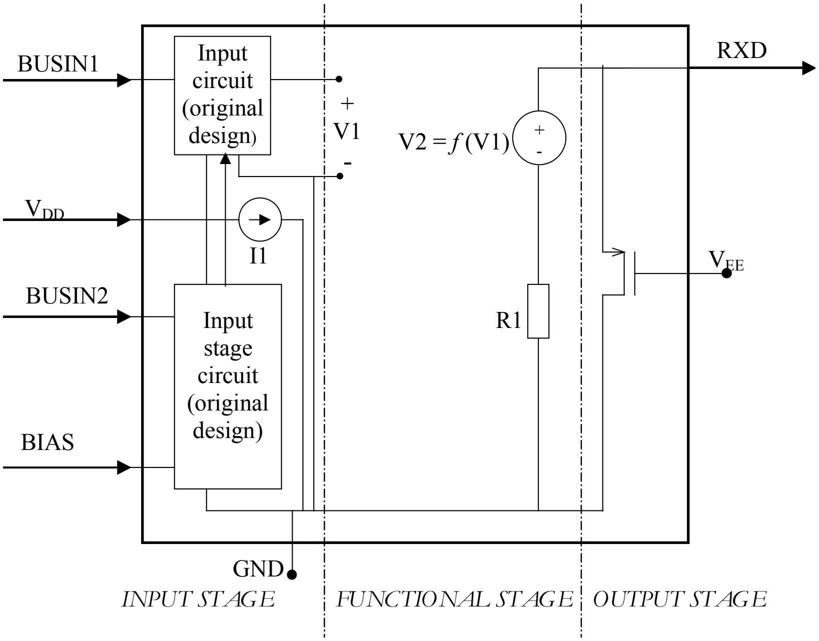


Figure 4.9: The scheme of the mixed-level model of the RECEIVER block.

The originally designed circuitry is reused in the input stage and output stage in the mixed-level model depicted in Figure 4.9. As a result of this reusing, the influence of the unexpected behaviour of the preceding and succeeding blocks can be minimized. For instance, if the BIAS input is changed to some strange value, the mixed-level model can still work properly because this influence can be isolated by the input stage.

Now more details with regard to the elements used in the model of Figure 4.9 will be given as following. V1 denotes the output voltage from the input stage of BUSIN1, which is the voltage propagated into the functional stage in the mixed-level model. I1 is a controlled current source to model the current through V_{DD} and its value is determined by the operational state of the RECEIVER block and the voltage at V_{DD}. According to the design specifications, the values for I1 are given in Table 4.3. The final equation used in the mixed-level model is: $I1 = \frac{I_{VDD} * V_{DD}}{12}$ because the current is also determined by the voltage at V_{DD}.

Table 4.3: The current I_{vdd} during different states when $V_{VDD} = 12V$.

State	**Current (I_{vdd})**
SLEEP	880nA
Work State 1 & 3	62uA
Work State 2	78uA

V2 is the output voltage source, which is controlled by the input voltage V1. The specifications to set its value are listed in Table 4.2 and Table 4.4. $Min(V_{EE} + 2, V_{DD} - 0.15)$ in Table 4.4 means that the minimal value of two is selected. The relation between V1 and V2 has been extracted based on the design specifications and V_{VEE} is the voltage level provided by the POWER block.

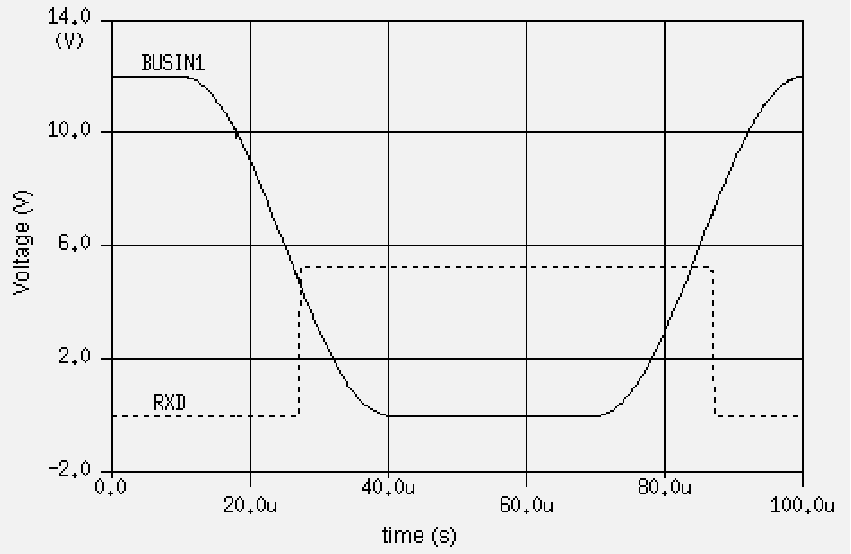
Table 4.4: The output voltage source at NRXDREC.

State	**V2 (V)**
Sleep	0
Work State 1 & 3	0
Work State 2	Min (V_{EE}+2, V_{DD}-0.15)

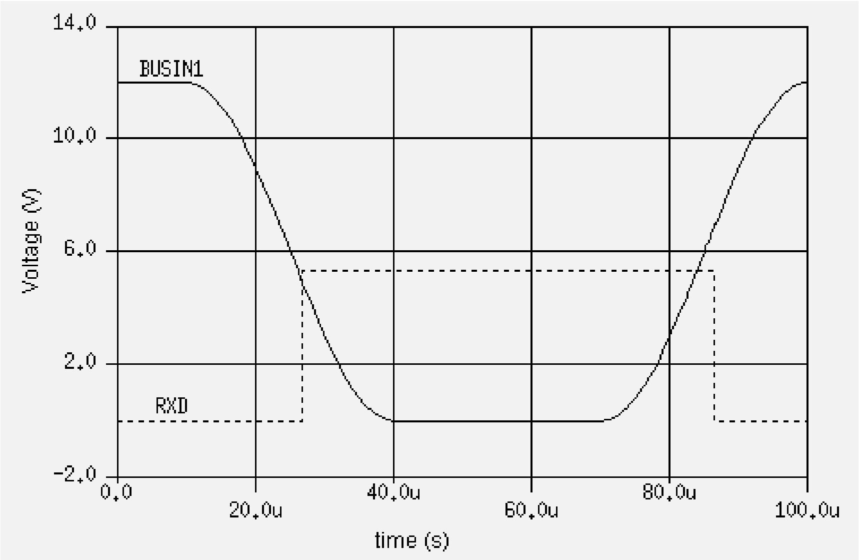
R1 models one part of the output resistance of the output RXD. Because the function of the RECEIVER block is to receive low-speed data, less than 20 kBaud, only a resistor is used here.

4.4.4 Verification of the functional behaviour of the mixed-level model

In order to verify the function of the mixed-level model of the RECEIVER block and to investigate how much CPU time can be reduced, several simulations have been carried out. First, the standalone fault-free RECEIVER block is connected with voltage/current sources and load resistors and the simulation results are shown in Figure 4.10. The simulation CPU-times are also shown in the figures, which indicates that mixed-level modelling can speed up the simulation time significantly (about 15 times).



(a) Simulation results for the function of the RECEIVER at *transistor level*; The CPU time is 10.41s.

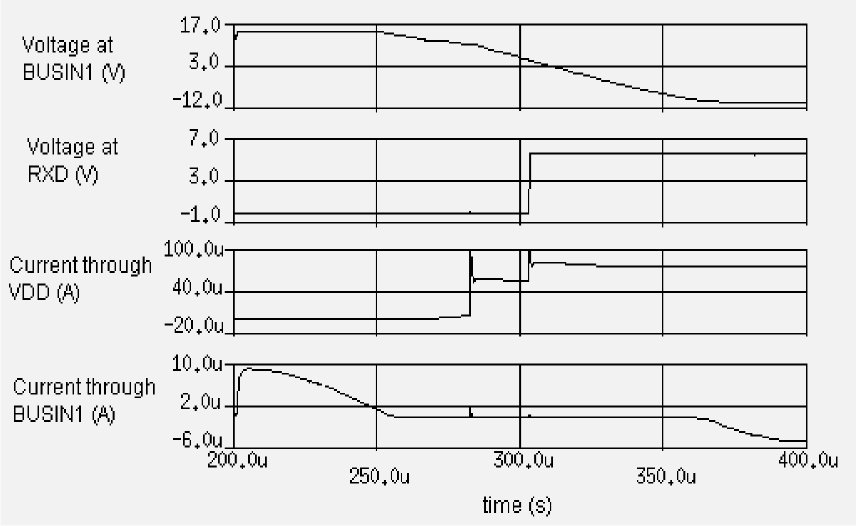


(b) Simulation results for the function of the RECEIVER with the *mixed-level model*; The CPU time is 0.69s.

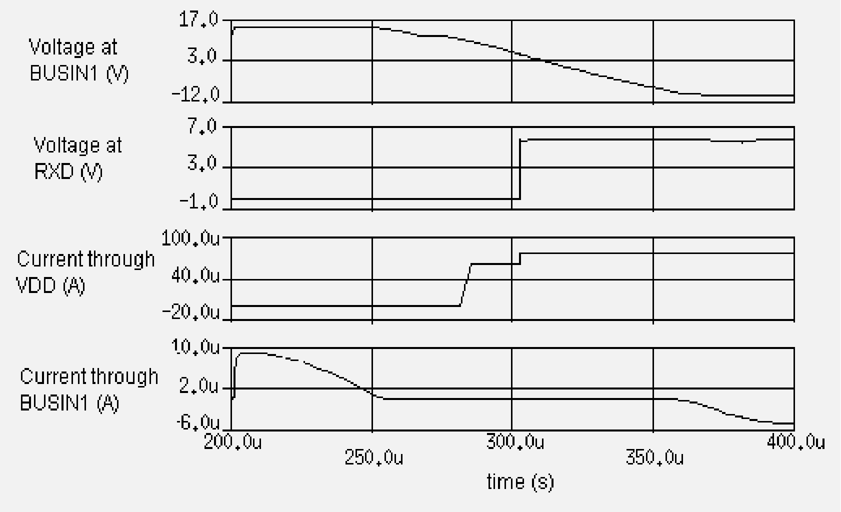
Figure 4.10: The comparison of the functional behaviour of the RECEIVER.

In the next step, the function of the mixed-level model has been verified in the case the RECEIVER block is embedded in the complete chip as shown in Figure 4.6. In order to verify the input-voltage protection function of the mixed-level model, a high voltage (15V) is applied at BUSIN1 for the beginning period as shown in Figure 4.11. The V_{DD} voltage is

12V during the simulation. As can be seen from the figure, the mixed-level model works properly with this high input voltage.



(a) Simulation at *transistor level* in the case of an extended voltage; The CPU time of the complete circuit is 933s.



(b) Simulation with the *mixed-level model* of the RECEIVER block in the case of an extended voltage; The CPU times of the whole circuit is 717s.

Figure 4.11: The comparison of the functional operation of the RECEIVER block if the RECEIVER block is embedded in the complete chip.

Because the chip is designed for low-speed data communication, the dynamic characteristics, which are shown as some glitches during the transitions in Figure 4.11(a), have no influence on the required function. It can therefore be concluded that the results in

Figure 4.10 and Figure 4.11 show that the mixed-level model of the RECEIVER block has the same function as the one at transistor level and the mixed-level modelling can significantly reduce simulation time. Because there are several blocks in the whole circuit and only the mixed-level of the RECEIVER block is used, only about 20% CPU time is reduced for the whole circuit simulation, as shown in Figure 4.11. However, theoretically, if all the blocks in the chip are simulated with mixed-level models, an increasing speed of around 15 times can be achieved, which is obtained for the standalone RECEIVER block.

4.5 Fault-simulation results

The purpose of introducing mixed-level models is to speed up fault simulation. Now, the proposed mixed-level model of the RECEIVER block is applied to the fault simulation of the real design in Dotss. In the following experiment, only the mixed-level model of the RECEIVER block has been used. The fault simulation results are shown in Table 4.5 and Table 4.6. In the fault simulation, a 200Ω resistor is used to model a bridging fault because it is the most likely value in the real process [Beu99] for this mixed-signal chip and DOTSS can only deal with bridging faults at this moment [Dot00]. The fault-simulation results using the transistor level of the RECEIVER block are also presented for the purpose of comparison.

Among the 87 faults in the reduced length fault list, there are 4 faults that only occur in the POWER block and there are 2 faults that only occur the TRANSMITTER block. For the data in Table 4.5, the test vector to measure the current through RXD for A different input voltage at INOUT is used. The measured current is I1 and I2 if the input voltage at INOUT is 0V and 12V respectively. Based on the specifications of the chip, the test limits for this test vector are $[-5\mu A, 5\mu A]$ for I1 and $[5mA, 15mA]$ for I2. This means that for the fault-free circuit, $-5\mu A \le I1 \le 5\mu A$ and $5mA \le I2 \le 15mA$ should hold. If for the faulty circuits, $I1 < -5\mu A$ or $I1 > 5\mu A$ or $I2 < 5mA$ or $I2 > 15mA$, then the fault can be detected by this test vector.

In the experiment of Table 4.6, another test vector is used in which the voltage at INOUT is measured if the input voltage at TXD is 12V and 0V respectively. The test limits of this test vector can be obtained from the design specification as [9.6V, 12V] for V1 and [0, 2.4V] for V2. These two test vectors have been used in the experiment because INOUT and RXD have both some connections with the RECEIVER block. Therefore, the results of the experiment can show more clearly with regard to the usefulness of the mixed-level model in analogue fault simulation.

In these two tables, “YES” means that the fault is detected by the test vector and “NO” means that the fault is not detected by the test vector. Because the simulation time of the same test vector for different faulty circuits is not same, the CPU time reduction rates in Table 4.5 and Table 4.6 are different for different faults.

The results in Table 4.5 and Table 4.6 show that fault simulation using mixed-level models can save simulation time, while providing the same fault-simulation results. The proposed mixed-level model of the RECEIVER block works correctly during fault simulation and turns out to be effective in reducing CPU time.

Table 4.5: Fault simulation results for 4 bridging faults in the POWER block.

Fault list	With/Without the mixed-level model	Simulation results	Simulation time	Detection

Bridging fault 1	The whole chip at transistor level	I1 = 10.28pA
		I2 = 11.703mA
			695s	NO

	With the RECEIVER mixed-level model	I1 = 10.279pA
		I2 = 11.703mA
			535s	NO

Bridging fault 2	The whole chip at transistor level	I1 = 10.274pA
		I2 = 115.132nA
			282s	YES

	With the RECEIVER mixed-level model	I1 = 10.274pA
		I2 = 127.681nA
			216s	YES

Bridging fault 3	The whole chip at transistor level	I1 = 10.279pA
		I2 = 11.426mA
			569s	NO

	With the RECEIVER mixed-level model	I1 = 10.279pA
		I2 = 11.426mA
			534s	NO

Bridging fault 4	The whole chip at transistor level	I1 = 10.280pA
		I2 = 6.517mA
			490s	NO

	With the RECEIVER mixed-level model	I1 =10.278pA
		I2 = 6.517mA
			457s	NO

Table 4.6: Fault simulation results for 2 bridging faults in the TRANSMITTER block.

Fault list	With/without the mixed-level model	Simulation results	Simulation time	Detection

Bridging fault 1	The whole chip at transistor level	V1 = 11.988V
		V2 = 7.984V
			217s	YES

	With the RECEIVER mixed-level model	V1 = 11.811V
		V2 = 7.984V
			174s	YES

Bridging fault 2	The whole chip at transistor level	V1 = 11.988V
		V2 = 1.039V
			292s	NO

	With the RECEIVER mixed-level model	V1 = 11.988V
		V2 = 1.039V
			257s	NO

Because there are several blocks in the circuit and only the mixed-level model of the RECEIVER block is used during the fault simulation, the CPU time is only reduced to around 10% to 20%, as shown in Table 4.5 and Table 4.6. However, it is expected that more CPU time can be reduced if more blocks in the circuit are simulated with mixed-level models.

4.6 Conclusions

In this chapter, a new general structure for mixed-level modelling with three stages is proposed to speed up analogue fault simulation. The approach has been applied to the RECEIVER block of an actual industrial chip. The fault simulations have been carried out using this mixed-level model in DOTSS, an industrial analogue fault simulation and test optimisation tool based on DOT. The results are compared with those at transistor level. The comparison shows that this kind of mixed-level modelling can effectively reduce the fault-simulation time, while providing the same results for fault simulation.

4.7 References:

[Ant95] B. A. A. Antao and A. J. Brodersen, “Behavioral simulation for analogue system design verification,” *IEEE Trans. VLSI Systems*, vol. 3, no. 3, pp. 417-429, Sept. 1995.

[All02] P. E. Allen and D. R. Holberg, “CMOS analogue circuit design,” 2nd edition, Oxford University Press, ISBN: 0195116445, 2002.

[Beu99] R. H. Beurze, “Practical implementation of Defect-Oriented Testing for a mixed-signal class-D amplifier,” M.Sc. thesis, University of Twente, the Netherlands, 1999.

[Cas91] G. Casinovi and A. S. Vincentelli, “A macromodeling algorithm for analogue circuits,” *IEEE Trans. Computer-Aided Design*, vol. 10, no. 2, pp. 150-160, Feb. 1991.

[Cha95] C. Y. Chao and L. S. Milor, “Performance modeling using additive regression splines,” *IEEE Trans. Semiconductor Manufacturing*, vol. 8, no. 3, pp. 239-251, Aug. 1995.

[Chr99] E. Christen and K. Bakalar, "VHDL-AMS --- a hardware description language for analogue and mixed-signal applications," *IEEE Trans. Circuits and Systems—II: Analogue and Digital Signal Processing*, vol. 46, no. 10, pp. 1263-1272, October 1999.

[Cur96] W. Current, J. Parker and W. Hardaker, “Behavioral modeling of analogue and mixed-signal circuits for efficient computer aided design,” *International Journal of Electronics*, vol. 80, no. 1, pp. 63-76, 1996.

[Dab99] J. Dabrowski, “Functional–level analogue macromodeling with piecewise linear signals,” *IEE Proceedings of Circuits, Devices and Systems*, vol. 146, no. 2, pp. 77-82, April 1999.

[Dev96] G. Devarayanadurg, P. Goteti and M. Soma, “Hierarchy based statistical fault simulation of mixed-signal ICs,” *Proc. of International Test Conference*, pp. 521-527, 1996.

[Dot00] “Dotss user manual for Dotss 3.0,” Philips ED&T, Eindhoven, August 2000.

[Eng00] N. Engin, “Linking mixed-signal design and test: generation and evaluation of specification-based tests,” Ph.D. thesis, University of Twente, the Netherlands, ISBN: 9036514940, 2000.

[Fan01] L. Fang, G. Gronthoud and H. G. Kerkhoff, “Reducing analogue fault-simulation time by using high-level modeling in Dotss for an industrial design”, *Proc. of IEEE European Test Workshop*, ISBN 0-7695-1877, Stockholm, Sweden, pp. 61-67, 2001.

[Fit97] D. Fitzpatrick and I. Miller, “Analogue behavioral modeling with the VERILOG-A language,” Kluwer Academic Publishers, ISBN: 0792380444, 1997.

[Gar01] R. Garcia, “Distributed test strategy cuts cost,” *Test & Measurement World (on line)*, November 2001.

[Ger96] A. Gertners, V. Zagursky and D. Saldava, “Behavior model of mixed ADC systems,” *Journal of Electronic Testing: Theory and Applications*, vol. 9, pp. 19-27, 1996.

[Get93] I. Getreu and D. Teegarden, "An introduction to behavioral modeling," *Microelectronics Journal*, vol. 24, no. 7, pp. 708-716, 1993.

[Kaa98] V. Kaal, “Structural test generation for embedded analogue macros,” Ph.D. thesis, University of Twente, the Netherlands, 1998.

[Kru95] W. Kruiskamp and D. Leenaerts, “Behavioral and macro modeling using piecewise linear techniques,” *Analogue Integrated Circuits and Signal Processing*, vol. 10, pp. 67-76, 1995.

[Mei96] P. B. L. Meijer, “Neural network applications in device and subcircuit modeling for circuit simulation,” Ph.D. thesis, Eindhoven University of Technology, Eindhoven, the Netherlands, 1996.

[Mil98] L. S. Milor, “A tutorial introduction to research on analogue and mixed-signal circuit testing,” *IEEE Trans. Circuits Systems—II: Analogue and Digital Signal Processing*, vol. 45, no. 10, pp. 1389-1407, 1998.

[Miu00] Y. Miura, “Fault behavior and change in internal condition of mixed-signal circuits,” *IEICE Trans. Information and Systems*, vol. E83-D, no. 4, pp. 943-945, 2000.

[Mos94] V. Moser, P. Nussbaum, H. P. Amann, L. Astier and F. Pellandini, “A graphical approach to analogue behavioral modeling,” *Proc. of European Design & Test Conference*, pp. 535-539, 1994.

[Nai93] R. Naiknaware, G. N. Nandakumar and S. R. Kasa, “Automatic test plan generation for analogue and mixed-signal integrated circuits using partial activation and high level simulation,” *Proc. of International Test Conference*, pp. 139-148, 1993.

[Roc96] S. Rochel, “Behavioral modeling and simulation of mixed-signal circuits,” *Analogue Circuit Design*, edited by J. H. Huijsing, R. J. V. D. Plassche and W. M. C. Sansen, Kluwer Academic Publishers, pp. 205-215, 1996.

[Ros98] R. Rosenberger and S. A. Huss, “A systems theoretic approach to behavioral modeling and simulation of analogue functional blocks,” *Proc. of Design, Automation and Test in Europe conference (DATE)*, pp. 721-728, 1998.

[Sac98] M. Sachdev, “Defect Oriented Testing for CMOS analog and digital circuits”, Kluwer Academic Publishers, ISBN 0-7923-8083-5, 1998.

[Xin98] Y. Xing, “Defect-Oriented Testing of mixed-signal ICs: some industrial experience,” *Proc. of International Test Conference*, pp. 678-687, 1998.

Chapter 5

Mixed-signal P1500-compatible core-based testing architecture

During test translation, some core-level test patterns might not be able to translate to the system-level due to the limited test access of the embedded cores. In order to solve this problem, a mixed-signal P1500-compatible core-based testing architecture is proposed in this chapter. Analogue wrapper cells and test buses are introduced to provide the extra test access for the embedded mixed-signal cores in this architecture. The embedded digital cores with an IEEE P1500 test wrapper can be directly used in this architecture to get extra test access. As an example, one analogue test path of a SoC including the new core-based architecture was evaluated by means of simulation. The simulation results show that the embedded core testing and the core-interconnection testing can be easily performed by using the proposed architecture. The simulation results also show that the performance degradation and silicon area overhead of the extra DfT circuits introduced by the proposed architecture are acceptable for analogue testing of embedded cores.

5.1 Introduction

The controllability of the test input signal and the observability of the test response for embedded cores are limited by the behaviour of the other cores in the normal functional path in mixed-signal SoC testing. It is possible that the test pattern generated for stand-alone cores cannot be realised for the embedded situation due to the limited test access. By using the testability-analysis method proposed in [Sta03], the embedded cores without sufficient test access, i.e. with low testability, can be found. In order to test those embedded cores with low testability, extra controllable paths should be added to the SoC to allow access to inputs and outputs of each embedded core from the primary SoC pins. Using DfT techniques is one solution to provide those extra controllable paths [Agr98]. In general, the mixed-signal boundary-scan technique and to less extend the analogue test bus is used as possible DfT solutions for digital testing and analogue testing respectively [Agr98, Jar91, Wag88].

The digital boundary-scan standard IEEE 1149.1 is the most widespread DfT solution for digital board-level testing [Ble93, Par99]. It has been defined in 1990 as an IEEE standard to test digital ICs on a board by controlling and observing the pins and internal nodes via a serial boundary-scan path and shift registers. The IEEE 1149.1 standard does not address analogue aspects of board-level testing, for instance detecting opens and shorts between the analogue ICs on the board. In order to solve this problem, the mixed-signal test bus standard IEEE 1149.4 described in [11494, Cro97, Sun98] has been recently defined.

Digital core-based testing, which is in the process of being described by the proposed IEEE standard P1500 [P1500], is an effective test method for the SoC containing embedded digital cores. The boundary-scan concept proposed in IEEE 1149.1 is employed in IEEE P1500 for chip-level testing [Adh99]. This test method is usable for testing embedded digital cores in a SoC. At this moment, the IEEE P1500 is restricted to digital cores only. However, nowadays many SoCs also include analogue and mixed-signal parts [Dol02, Nag99, Oze00]. Until the year 2000, there has been no effective solution available to test embedded analogue /mixed-signal cores.

A new mixed-signal P1500-compatible core-based testing architecture is proposed in this chapter. It can provide effective test access for both embedded digital cores and embedded analogue cores. The proposed architecture is compatible with IEEE P1500. Hence, the digital cores with IEEE P1500 test wrappers can be used in the proposed architecture without any problem. With regards to the analogue testing of the mixed-signal cores, the analogue wrapper cell structure and analogue test buses have been designed to provide the test access. This means that, by using our proposed architecture, the analogue test stimuli can be transported from the external SoC inputs to the inputs of the embedded core under test, and subsequently the test response can be propagated from the embedded core to the external SoC outputs [Fan00, Sta00].

The general concepts developed in IEEE 1149.1, IEEE 1149.4 and IEEE P1500 are introduced in section 5.2. Then, the new mixed-signal P1500-compatible core-based testing architecture is proposed. Afterwards, the details of the main parts for the proposed architecture are discussed in section 5.3, including the analogue input/output wrapper cell, the wrapper instruction register and the TAP controller. Finally, the proposed architecture has been applied to the analogue part of a locally designed mixed-signal SoC and the simulation results for the embedded core testing and the core-interconnection testing are presented in section 5.4. The performance degradation and the silicon area overhead of the proposed architecture are also presented in section 5.4.

5.2 Core-based testing

In this section, the concepts for the digital boundary-scan standard IEEE 1149.1 and the mixed-signal test bus standard IEEE 1149.4 are briefly discussed first. Then the digital core-based testing standard IEEE P1500 is introduced. Finally, the new mixed-signal core-based testing architecture based on P1500 is proposed.

5.2.1 Digital boundary-scan standard IEEE 1149.1

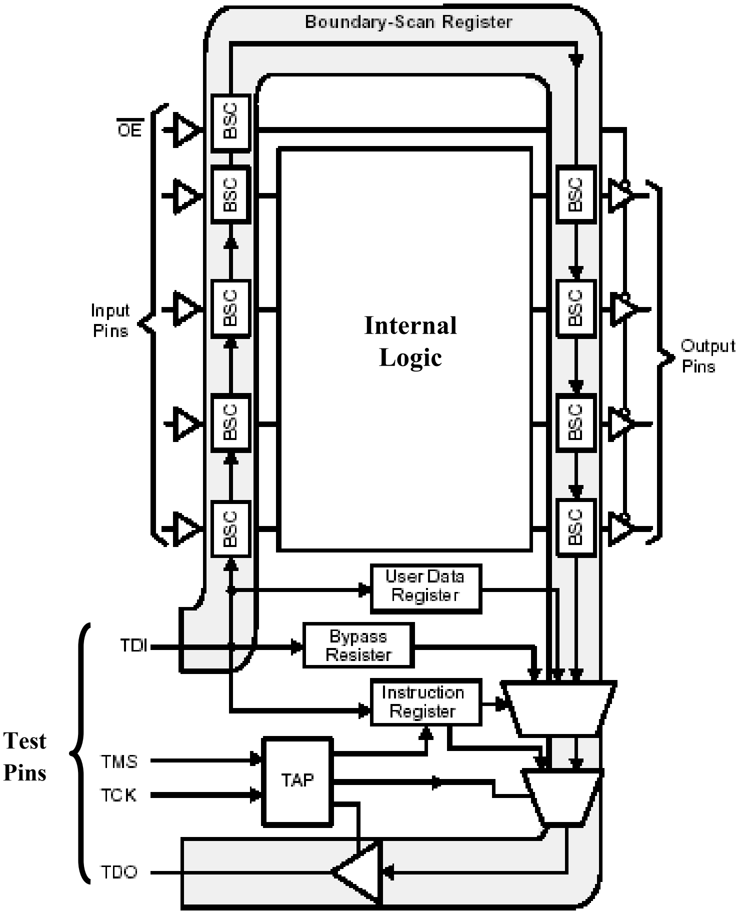


Figure 5.1: Overview of an 1149.1 compliant IC.

The digital boundary-scan standard, which is well described in [Ble93] and [Par99], has been defined in 1990 as the IEEE standard 1149.1. It is able to test digital ICs on a board by controlling and observing the pins and internal nodes via a serial boundary-scan path and shift registers. The architecture of an IEEE 1149.1 compliant IC is depicted in Figure 5.1. The main parts in this architecture are the instruction register, the bypass register, the boundary-scan register cells and the test access port (TAP) controller. In the figure, BSCs are the boundary-scan cells connected to chip input and output pins, TDI is the test-data input, TDO is the test-data output and TCK is the test clock.

As shown in Figure 5.1, the boundary-scan cells are located between the bonding pads and the internal circuitry. The cell can connect or disconnect the input/output pin from the system circuitry and can apply or measure a digital signal at the pin of the IC. The boundary-scan cells can be connected into a scan chain, through which the digital signal can be shifted sequentially. By using this shifting mechanism, the test-input signal can be applied to the specified pin and the test-output signal at a certain specified pin can be observed. The TAP controller determines in which of the registers the serial data is shifted. The instruction register determines in which mode all boundary-scan cells in the IC will operate. This depends on the instruction shifted into the instruction register.

5.2.2 Mixed-signal test bus standard IEEE 1149.4

The IEEE 1149.1 standard does not address analogue aspects of board-level testing, such as measuring passive component values, detecting opens and shorts between off-chip analogue components, or carry out diagnostics on differential signal paths. The mixed-signal test bus standard IEEE 1149.4 described in [11494, Cro97, Rob97, Sun98] provides the solutions to these challenges.

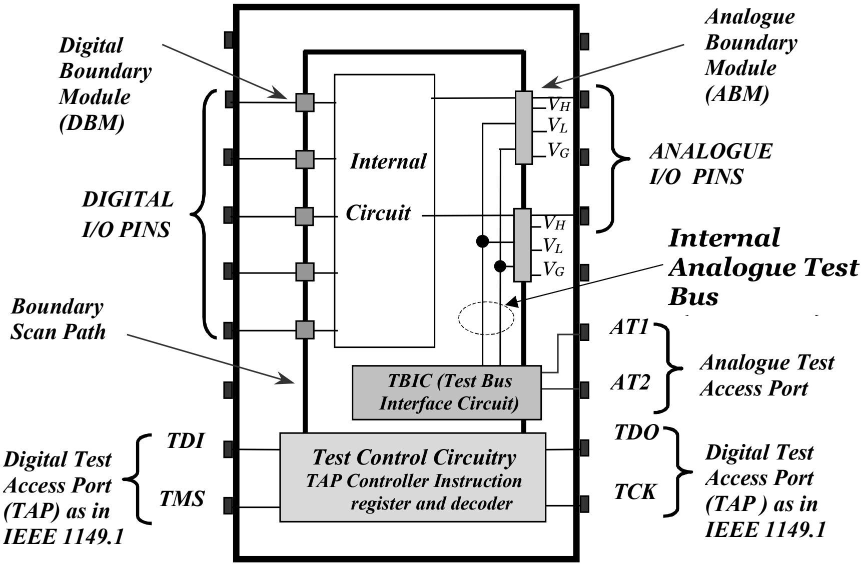

Digital Boundary Module (DBM)
DIGITAL I/O PINS
Boundary Scan Path
Digital Test Access Port (TAP) as in IEEE 1149.1
TDI
TMS
Internal
Circuit
TBIC (Test Bus Interface Circuit)
Test Control Circuitry
TAP Controller Instruction register and decoder
VH
VL
VG
VH
VL
VG
Analogue Boundary Module (ABM)
ANALOGUE I/O PINS
Internal Analogue Test Bus
AT1
AT2
Analogue Test Access Port
TDO
TCK
Digital Test Access Port (TAP) as in IEEE 1149.1

Figure 5.2: Overview of an IEEE 1149.4 compliant IC.

Figure 5.2 shows the general structure of an IEEE 1149.4 compliant IC. Compared to the structure shown in Figure 5.1, the hardware features added by 1149.4 are:

— an analogue TAP comprising a minimum of two external pins (AT1, AT2)

— on-chip analogue bus wires (AB1, AB2)

— a test bus interface circuit (TBIC) connecting the on-chip and off-chip analogue buses

— an analogue boundary module (ABM) at each analogue pin.

With regard to the analogue test buses, a single wire bus is used for delivering a stimulus signal from an input pin ("AT1") to selected circuit nodes, and a signal wire bus is required for routing selected analogue output signals to an output pin ("AT2"). The TBIC can be digitally controlled in order to connect or disconnect the internal test bus from the board-level test bus.

5.2.3 Digital core-based testing using the proposed standard IEEE P1500

The work recently carried out within the P1500 working group [P1500, Adh99] aims at defining a uniform, yet flexible hardware interface between an embedded digital core within a SoC and its environment. With this digital core-based architecture, the pre-defined test patterns of a core can be delivered to and from the inputs and outputs of embedded digital cores. It also defines the additional standard hardware for the embedded cores to enable the core internal test and the core-level interconnection tests. A scheme of a SoC with a P1500 core-based testing architecture is shown in Figure 5.3.

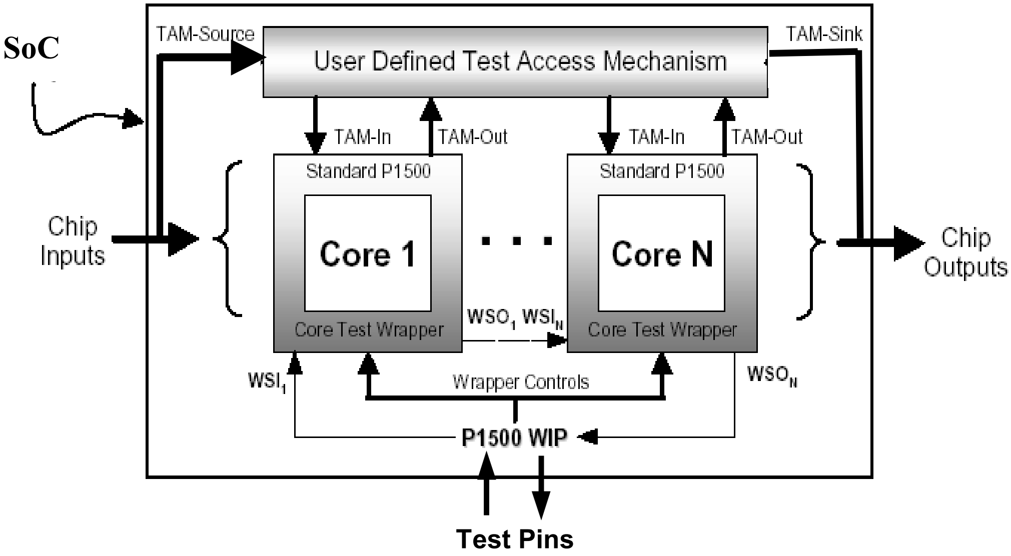


Figure 5.3: SoC with P1500 wrapped cores.

WSI and WSO are the wrapper serial input and output, and WIP is the Wrapper Interface Port. By controlling the user-defined Test Access Mechanism (TAM) and the wrapper cells around the embedded cores, the following test modes can be achieved:

Normal mode. This mode allows the system chip and cores to function in their normal system operation mode.

Internal core test mode. This mode enables the pre-generated core-level test patterns to be applied to the embedded core.

Interconnect test mode. This test mode enables the core test wrapper to be used for testing the system chip interconnect and logic between cores.

Test isolation mode. This mode allows the core to be isolated in order to facilitate testing of other cores in the SoC.

As shown in Figure 5.3, the three key components used in the P1500 architecture are the core test wrapper, the test control mechanism and the test access mechanism. The core test wrapper provides mechanisms for core test-data access and core-test isolation. The wrapper allows for controlling core inputs and observing core outputs via the TAM, enabling internal tests of the cores to be re-applied at the system-chip level. The core test wrapper is a standard component of P1500 and its behaviour is defined by the standard [P1500]. The general structure of the core test wrapper is illustrated in Figure 5.4. As can be seen from the figure, the concept of the boundary-scan methodology proposed in IEEE 1149.1 is employed in this digital core-based architecture. Although the behaviour of the core test wrapper is defined in the P1500 standard, its *implementation* is user defined. This means that different users can have different implementations.

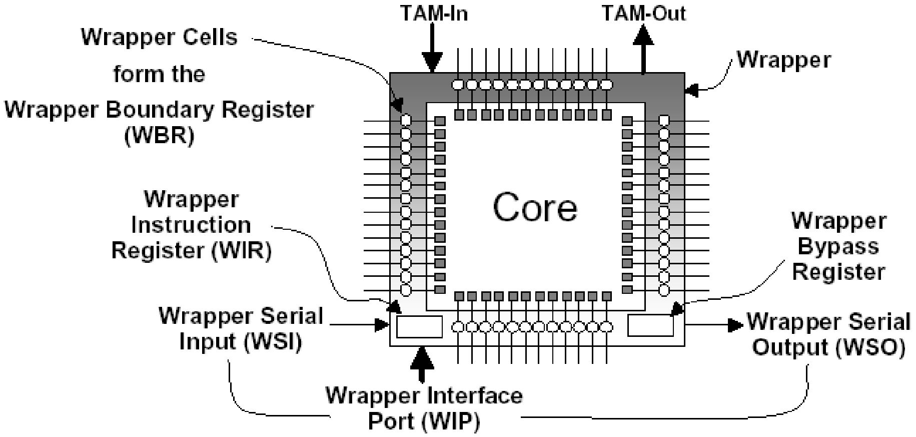

TAM-In
TAM-Out
Wrapper Cells form the Wrapper Boundary Register (WBR)
Wrapper
Core
Wrapper Instruction Register (WIR)
Wrapper Bypass Register
Wrapper Serial Input (WSI)
Wrapper Serial Output (WSO)
Wrapper Interface Port (WIP)

Figure 5.4: The components of the P1500 core test wrapper.

The purpose of the P1500 test control mechanism is to enable and control test modes and test signals within the system chip. The main component for the test control mechanism is the wrapper interface port (WIP). The user defined TAM is employed to transfer test data between the primary I/O of the system chip and the core test wrapper.

5.2.4 The proposed mixed-signal core-based testing architecture

As basis for our architecture, the one described for digital cores in [P1500] was used. However, now a mixed-signal core has been assumed as shown in Figure 5.5. In order to provide the analogue test signal access for the embedded mixed-signal cores, the user-defined test access mechanism on top is extended with analogue test buses. The input analogue wrapper cell and output analogue wrapper cell are introduced for the core test wrapper and the standard P1500 WIP is enhanced with an analogue test-bus controller part. Therefore, digital core with P1500 test wrapper can be directly used in the proposed architecture because it fits within the general architecture as presented in IEEE P1500 [Adh99] and the TAP controller can also provide the standard P1500 control signals for the test wrappers of the embedded digital cores.

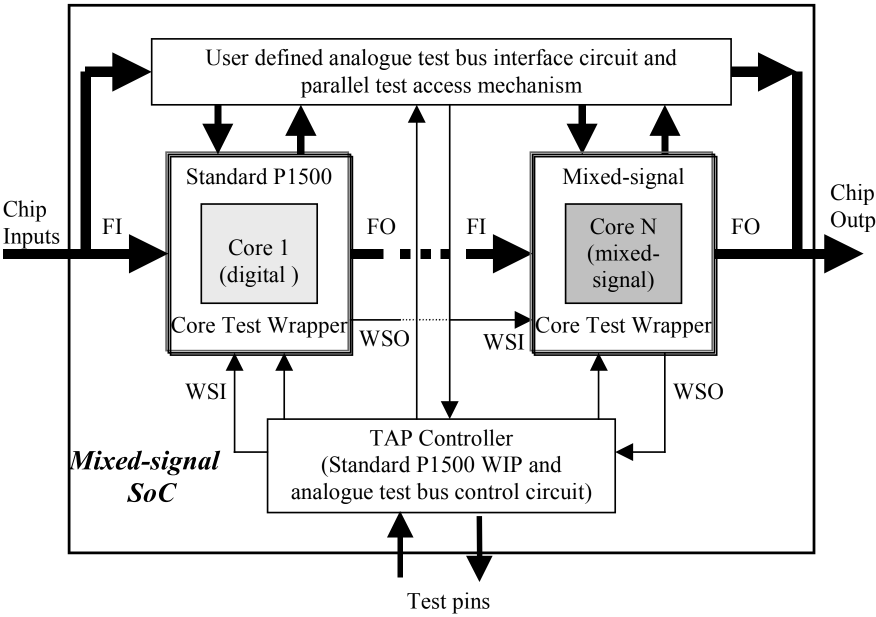


Figure 5.5: Proposed mixed-signal core-based testing architecture.

More details of the mixed-signal core and its surrounding wrapper cells are depicted in Figure 5.6. In this structure, FI/FO means functional input/output. The mixed-signal core is divided into analogue inputs and outputs and their digital counter parts. Each core input and output has a wrapper cell connected to it. Depending on the type of core input/output, the proper wrapper cell can be chosen from the available wrapper cell library by the EDA (Electronic Design Automation) tools automatically or by the SoC designer manually. For instance, the digital input wrapper cell is connected to the digital input of the core and the analogue voltage driven output wrapper cell is connected to the analogue voltage driven output of the embedded core.

Due to the fact that there are different types of core inputs and outputs in the SoC, different wrapper cells should be designed and stored in the wrapper cell library. However, all wrapper cells should have the same control interface and all the control signals have to be provided by the Wrapper Instruction Register (WIR).

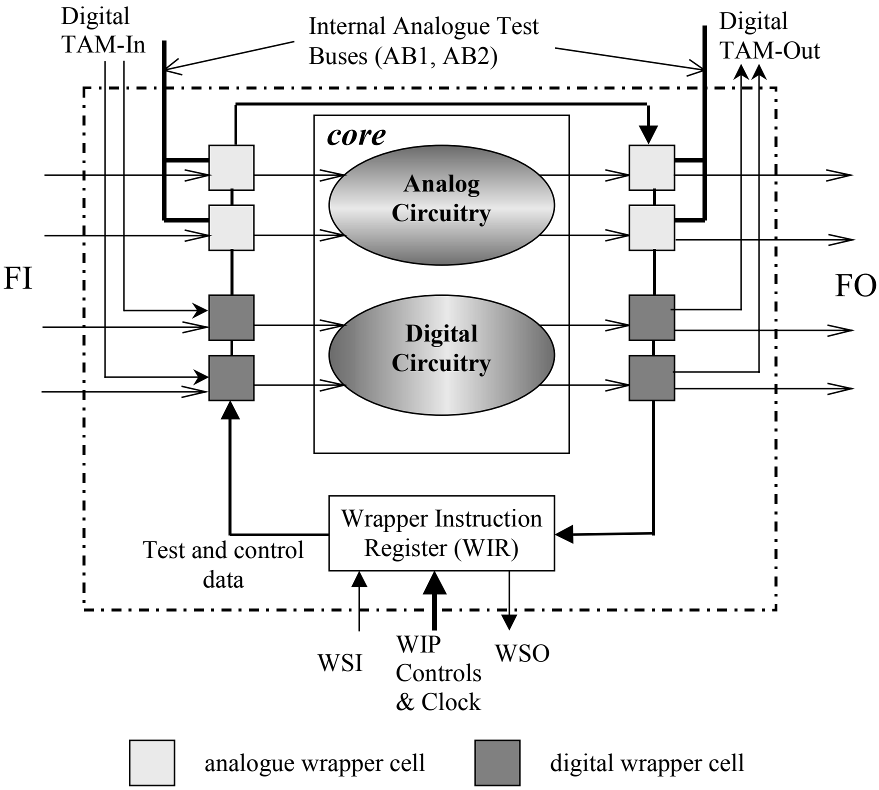


Figure 5.6: A mixed-signal core with the corresponding core test wrapper, including analogue and digital wrapper environment and analogue test busses.

The digital input and output wrapper cells are identical to the ones as suggested within [P1500]. These digital core wrapper cells can operate in a certain mode based on the received standard P1500 control signals via the TAP controller and WIR. The analogue input and output wrapper cells are connected to the analogue input/outputs of the core. They are also connected to the user-defined TAM via the internal analogue test buses (AB1 and AB2). The WIR is connected to the on-chip TAP controller. It controls all the states of the wrapper cells within the corresponding core test wrapper.

5.3 The hardware implementation of the proposed core-based testing architecture

In the previous section, a new mixed-signal P1500-compatible core-based testing architecture has been proposed combining the digital core-based testing techniques and an analogue test-bus technique. In this section, the main components used in this structure will be presented. The digital input and output wrapper cell will not be discussed here because they are identical to the wrapper cells defined in IEEE P1500 [P1500]. However, the structure and implementation of the analogue input and output wrapper cells will be explained in this section. The implementation of the enhanced WIR, the TAP controller and the user-defined TAM will be discussed as well.

5.3.1 Analogue input wrapper cell

The basic scheme of the general analogue input wrapper cell is depicted in Figure 5.7. The wrapper cell consists of three analogue switches SB1, SB2 and SC. The control signals for these three switches are coming from the WIR of the core. In this structure, an important switch is SC, which can connect or disconnect the core from its surroundings. As can be seen from Figure 5.7, this structure has some similarities with the ABM defined in IEEE 1149.4 [11494]. However, it has fewer components than the ABM structure. The main reason is that lower measurement capability is required for the core-interconnection testing than the IC-interconnection testing.

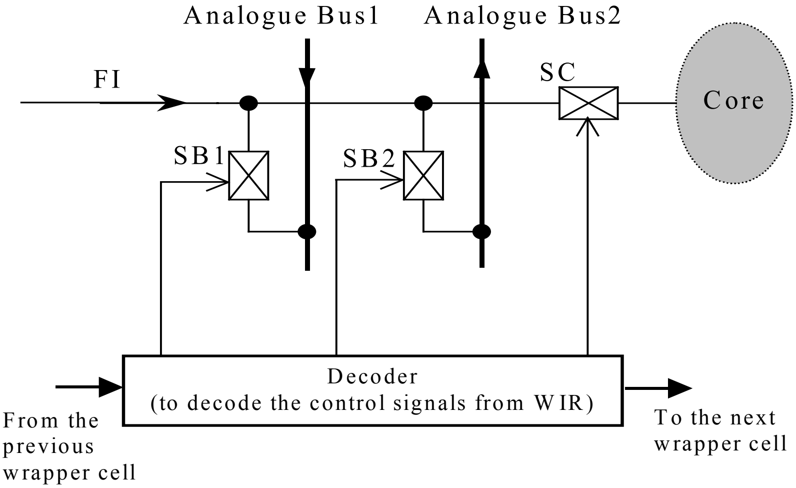


Figure 5.7: The proposed basic structure of the general analogue input wrapper cell.

The required test modes of the presented analogue input wrapper cell are given in Table 5.1. It provides the most basic operations required.

Table 5.1: Analogue input wrapper cell operational modes.

SC	SB1	SB2	Test Mode
On	*Off*	*Off*	Normal function
On	*On*	*Off*	Internal core test
Off	*Off*	*On*	Interconnection test
Off	*Off*	*Off*	Test isolation

The *normal function* mode allows the system chip and cores to function in their normal system operation mode. The *core test* mode enables the pre-generated analogue test signals [Sta03] for the embedded mixed-signal core to be applied to the core via the user-defined

TAM and the core test wrapper. The *interconnection test* mode enables the core test wrapper to be used for testing the interconnection lines/wires between cores. The *test isolation mode* allows the core to be isolated in order to facilitate testing of other cores in the SoC. These four test modes are the basic required ones. Besides the test modes listed in Table 5.1, more user-defined test modes can be defined if they are required by the user.

Since the structure shown in Figure 5.7 is the basic structure, more complex structures can be used if more testing functions are required. [Hui02] proposed an analogue input wrapper cell (Figure 5.8) similar to the analogue boundary module (ABM) as suggested in IEEE 1149.4. The switches H, L and R can connect the core input to a high, low or reference voltage. These voltages may be provided to the wrapper cell separately, or by making a connection to the power-supply voltage V_{dd} or GROUND. The comparator can be used to digitalis the test response for interconnection testing.

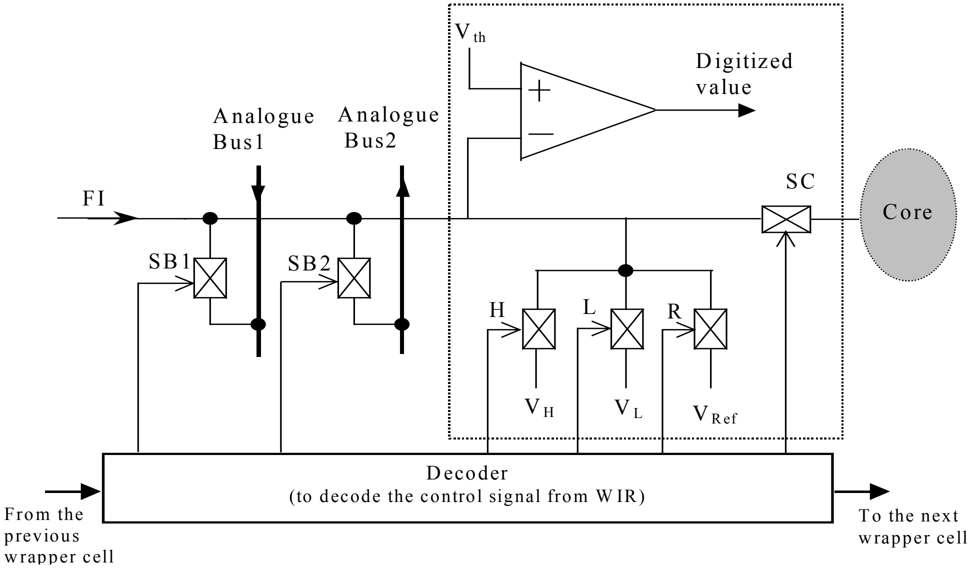


Figure 5.8: An input wrapper cell structure similar to the analogue boundary module (ABM) as used in IEEE 1149.4.

With regard to the implementation of the analogue input wrapper cell, an analogue voltage-driven input wrapper cell has been designed in 0.8μm AMS CMOS technology by us [Ber02]. The switches used in the wrapper cell are made of transmission gates, a matched pair of a NMOS and a PMOS transistor. The DC switch-on resistance of the transmission gate used in [Ber02] is smaller than 800Ω and its corresponding switch-off resistance is larger than 1GΩ. The AC simulation results of the transmission gate show that the AC amplitude decreases less than 0.3% and the phase change is smaller with 1.4^o if the frequency is lower than 100MHz. The simulation results of the proposed complete input wrapper cell (Figure 5.8) in [Ber02] also show that the AC amplitude and phase degradation are 0.5% and -5^0 respectively if it is operating at 100MHz in the “Normal function” mode. This frequency range is sufficient for audio and video application circuits.

5.3.2 Analogue output wrapper cell

The basic structure of the analogue output wrapper cell is shown in Figure 5.9. It resembles the previous analogue input wrapper cell. However, four switches are used in this structure. The switch SZ is added, which is used to connect a certain load to the output of the core. In this way, the core can still be tested under the same loading conditions if the core is disconnected from its surroundings. Of course, for the rare case that the functioning of the core does not depend on its load, this switch can be left out of the design.

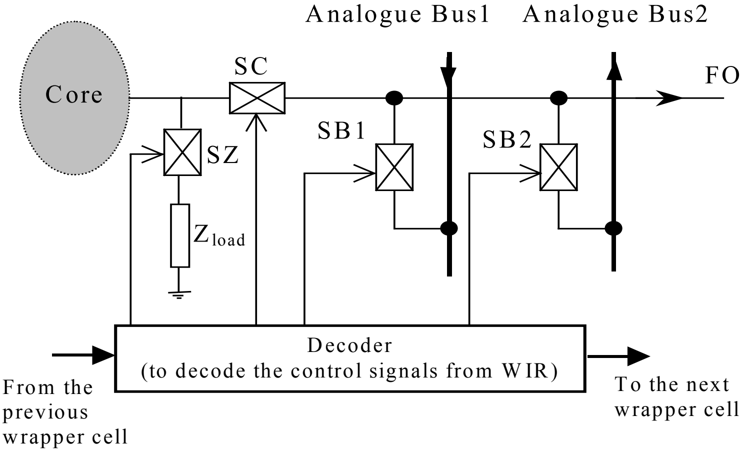


Figure 5.9: The proposed general structure of basic the analogue output wrapper cell.

The different conditions of the switches for the required test modes of the output wrapper cell are listed in Table 5.2.

Table 5.2: Analogue output wrapper cell operational modes.

SC	SB1	SB2	SZ	Test Mode
Off	*Off*	*Off*	*Off*	Normal function
On	*Off*	*On*	*On*	Core test
Off	*On*	*Off*	*Off*	Interconnection test
Off	*Off*	*Off*	*Off*	Test isolation

Based on this proposed general structure, an analogue output wrapper cell has been designed by us in [Ber02], employing CMOS transmission gates as switches. The simulation results of this analogue output wrapper cell can be used in audio and video applications because the AC amplitude and phase degradation are 0.6% and -5^0 respectively if it is operating at 100MHz in the “Normal function” mode.

5.3.3 Wrapper instruction register

All wrapper cells are controlled by the wrapper instruction register (WIR) in the proposed architecture. The function of the WIR is to receive instruction data or test data from the TAP controller and decode the instruction data. Then, it controls the operating status of every wrapper cell in the embedded core based on the instruction received. The WIRs for different cores and the TAP controller are connected together via a serial scan chain as shown in Figure 5.10. As shown in the figure, the WIRs receive the test data via this scan chain. With regard to the data shifted into the WIR, depending on the control signals from the TAP controller, the WIR can treat this serial data as an instruction or as data for the wrapper cells.

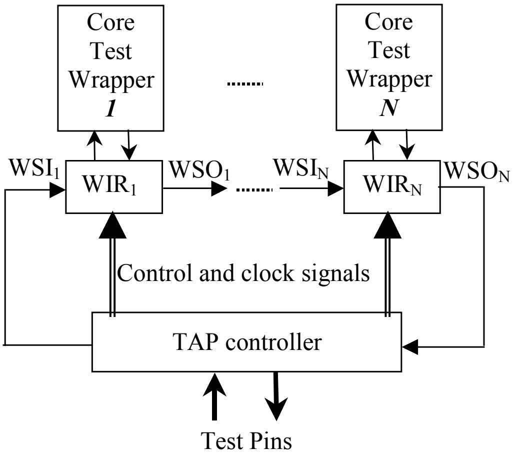


Figure 5.10: The illustration of the connection between the WIRs and TAP controller.

The simplified scheme of a WIR is depicted in Figure 5.11. It consists of the Bypass Register, the Instruction Register and the serial data interface connected with the test wrapper cells.

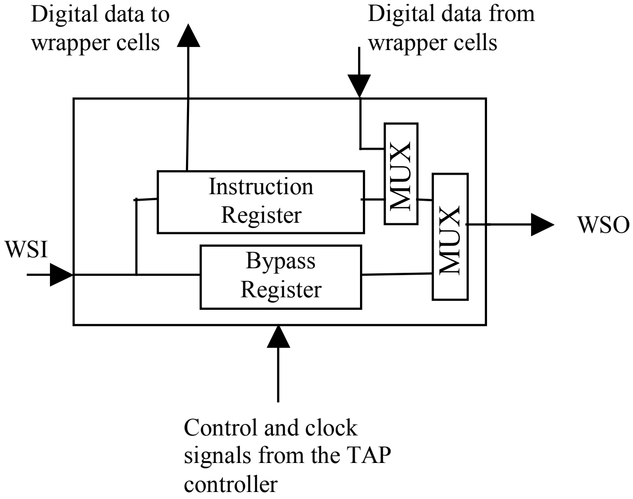

Digital data to wrapper cells
Digital data from wrapper cells
MUX
Instruction Register
MUX
WSO
WSI
Bypass Register
Control and clock signals from the TAP controller

Figure 5.11: The simplified scheme of a WIR.

The WIR receives the data from the TAP controller via the WSI. It then decodes the data by the Instruction Register if the data is meant for this core, or bypasses the data to the next WIR by using the Bypass Register. On the other hand, by using the internal scan chain inside the core test wrapper (Figure 5.6), it sends the digital test data and control signals to the wrapper cells and receives the digital test response data from the wrapper cells if necessary. It can also send the digital test response to the primary chip output via the succeeding WIRs and the TAP controller. Based on the required functionality of the WIR, a WIR has been implemented in CMOS by us in [Hui02].

5.3.4 TAP controller

The TAP controller is the test control interface with the outside world. It mainly consists of two parts: the standard P1500 WIP for providing the standard P1500 control signals for the digital cores with P1500 test wrapper and the analogue test-bus control circuit to provide the control signals for analogue wrapper cells and the analogue test buses. All core test wrapper cells inside the SoC can be controlled at the chip input pins through the TAP controller. The complete design of our mixed-signal TAP controller in CMOS technology can be found in [Hui02], which can provide the control signals for the standard P1500 wrapper cells, analogue wrapper cells and the analogue test buses.

5.3.5 User defined test access mechanism

Our user-defined test access mechanism consists of the parallel test access mechanism for digital testing and the analogue test buses for analogue testing. All input wrapper cells and output wrapper cells should be connected to the TAM. As a user-defined part, the design of this hardware can be entirely left to the SoC designer. However, similar to the situation that the IC designer can use a lot of pre-designed blocks from the library of EDA tools, the small components of the TAM can also be pre-designed and stored in a library. This simplifies the design of the user-defined TAM by the SoC designer.

As a general consideration, different test buses should be available for each kind of test signals. In practise this means there can be voltage-driven test buses or current-driven test buses. It is also possible to have more than one test bus of a certain kind. In [Ber02], the structure of one type of TBIC and a voltage-driven test bus is proposed.

If the embedded analogue cores have differential inputs/outputs, then differential analogue test buses have to be employed. Meanwhile, the input/output analogue wrapper cells with differential inputs/outputs have to be used as well.

5.4 Example circuit and simulation results

In order to verify the proposed architecture in mixed-signal SoC testing, an experiment was carried out on a part of a locally designed SoC in this section.

5.4.1 Example circuit

The example circuit shown in Figure 5.12 is a SoC containing a Pre-amplifier, a continuous time filter (FILTER), a voltage-current converter (VIC) and other digital cores [Sta02b]. The input signals are connected to the Pre_amplifier and the load resistors are connected to the outputs of the VIC. The considered analogue test path is the signal path containing the serial interconnection of the Pre_amplifier, the FILTER and the VIC. The FILTER core has one low-pass filter (LPF) path and one band-pass filter (BPF) path [Yuf99]. The characteristics of the LPF and BPF are determined by the bias current, which is provided by other circuits in the SoC as shown in Figure 5.12. If the bias current is 200nA, then the cut-off frequency of the LPF is 18kHz (-3dB point); the centre frequency of the BPF is 13.4kHz and the bandwidth 12.43kHz (-3dB point). Because the DfT architecture for the digital embedded cores is already provided by IEEE P1500 [P1500], only the DfT architecture for the analogue testing of the mixed-signal cores is discussed here. Therefore, the digital cores and the other circuitry in the SoC have been clustered in the figure for the sake of simplicity.

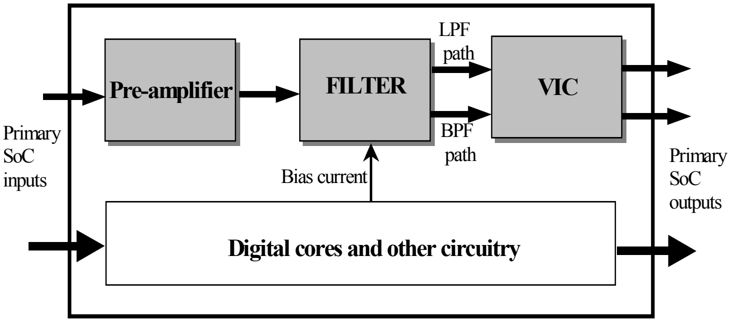


Figure 5.12: Illustration of a SoC containing embedded Pre-amplifier, FILTER, VIC, and other digital cores and circuitry.

In order to provide the test access to the embedded digital and analogue cores, the core-based testing architecture as proposed in Figure 5.5 has been applied to this SoC. The SoC with the proposed architecture is shown in Figure 5.13. For simplicity, only the analogue wrapper cells and the corresponding wrapper instruction register (WIR), analogue test buses and the TAP controller are depicted in Figure 5.13.

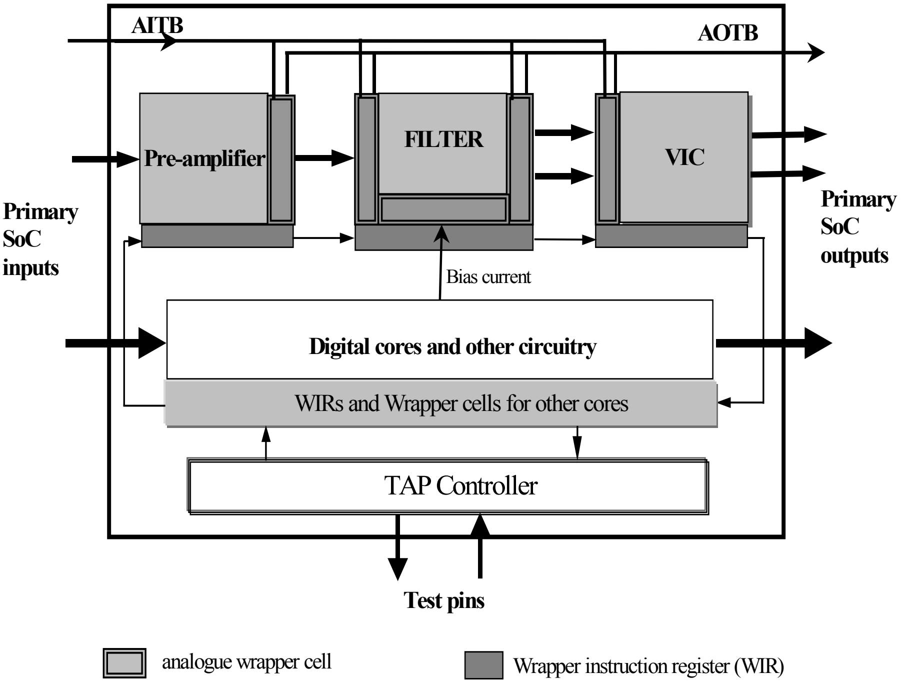


Figure 5.13: The illustration of the SoC with the proposed mixed-signal core-based testing architecture.

In this architecture, AITB is the analogue input test bus and AOTB is the analogue output test bus. The analogue input wrapper cells and output wrapper cells are implemented based on the general structure presented in Figure 5.7 and Figure 5.9. CMOS transmission gates are used as the switches in the wrapper cells. The circuit details of the analogue wrapper cells can be found in [Ber02]. The implementations of the WIR and the TAP controller have been presented in [Hui02].

5.4.2 Simulation results for normal operation

In order to observe the influence of the extra circuits introduced by the proposed architecture with respect to the normal circuit operation, the DC and AC simulation results of the analogue test path with and without wrapper cells have been compared. The analogue test path considered here contains the Pre-amplifier, the FILTER and the VIC, as shown in Figure 5.12 and Figure 5.13. The bias current for the FILTER is 200nA during the DC and AC simulations. For the circuit show in Figure 5.13, all the input and output wrapper cells operate in the "*normal function*" mode during the simulation.

The DC simulation results for the test path without wrapper cells are shown in Figure 5.14. In this figure, the x-axis shows the DC voltage at the input of the Pre-amplifier and the y-axis represents the DC current in the load resistors (R=500Ω) connected to the outputs of the VIC. Since the FILTER has one LPF path and one BPF path, there are two paths in the

VIC: one for the LPF path and one for the BPF path. Figure 5.14(a) and Figure 5.14(b) depict the DC current in the load resistor connected to the VIC output of the LPF path and BPF path respectively. The DC simulation results for the normal function mode with the core-based testing architecture are shown in Figure 5.15. Similar to as Figure 5.14, the DC currents in the load resistor connected to the VIC output of the LPF path and BPF path are shown in Figure 5.15(a) and Figure 5.15(b) respectively.

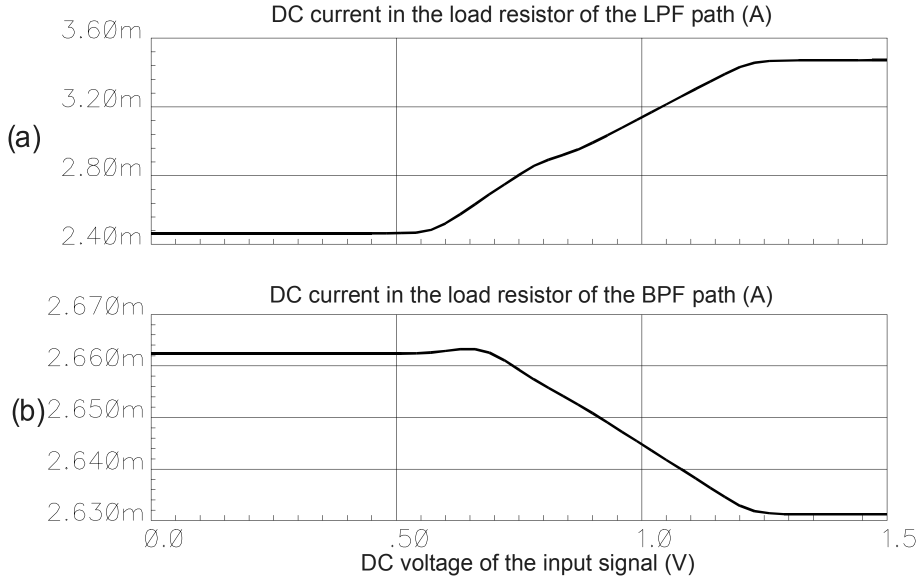


Figure 5.14: DC analysis results of the test path without wrapper cells.

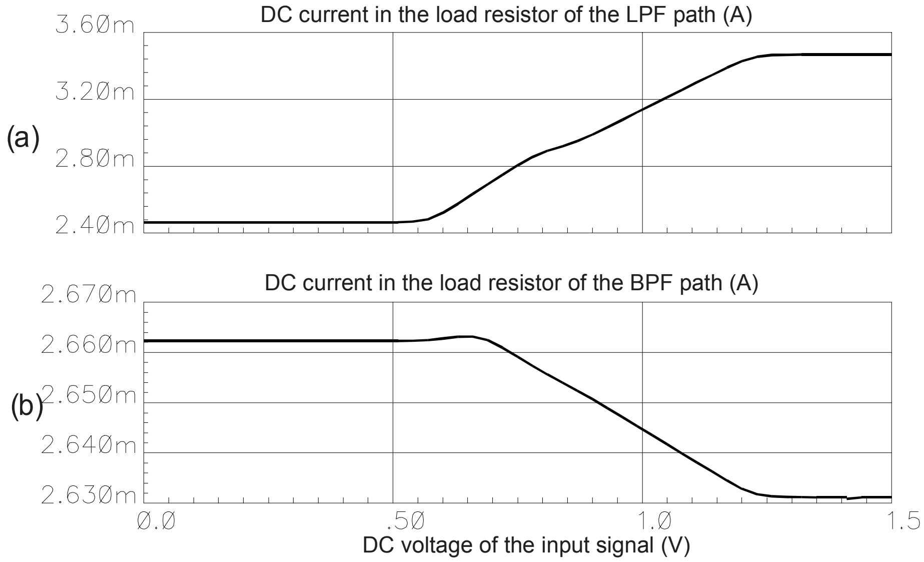


Figure 5.15: DC analysis results of the test path with analogue wrapper cells.

In order to observe the influence of the proposed architecture to the DC performance of the test path, the difference between the DC simulation results with and without wrapper cells are shown in Figure 5.16. The data for this figure has been obtained by subtracting the simulation results without the wrapper cells of the corresponding simulation results with the wrapper cells. For the DC current in the load resistor of the LPF path, the absolute value of the difference is smaller than 300nA, as shown in Figure 5.16(a). Since the DC current during the normal operation is larger than 2.4mA, as shown in Figure 5.14(a), the difference is smaller than 0.0125%. Similarly, the difference for the DC current in the load resistor of the BPF path is shown in Figure 5.16(b). It shows that the absolute value of the difference is smaller than 110nA. The difference is less than 0.005% as the DC current during the normal operation is larger than 2.63mA as shown in Figure 5.14(b). Therefore, it can be concluded that the proposed architecture has only an extremely small influence on the DC performance of the example circuit shown in Figure 5.12.

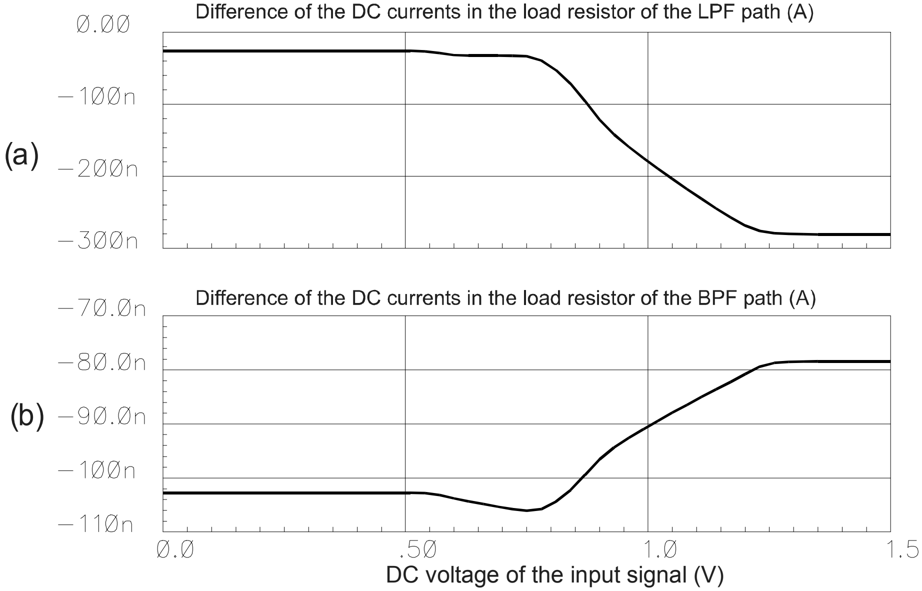


Figure 5.16: The difference of the DC analysis of the test path with/without wrapper cells in the LPF path (a) and BPF path (b).

Similar to the DC simulation, the AC simulation results for the test path with and without wrapper cells are shown in Figure 5.17 and Figure 5.18 respectively. In these two figures, the x-axis represents the frequency of the AC analysis and the y-axis represents the AC amplitude of the current in the load resistor connected to the output of the VIC. During the AC simulation, the signal at the input of the pre-amplifier is a sine waveform with DC offset of 0.675V and AC amplitude of 100mV. Figure 5.17(a) and Figure 5.18(a) show the AC amplitude of the current in the load resistor connected to the LPF path, while Figure 5.17(b) and Figure 5.18(b) show the AC amplitude of the current in the load resistor connected to the BPF path. The waveforms shown in Figure 5.17 and Figure 5.18 match quite well.

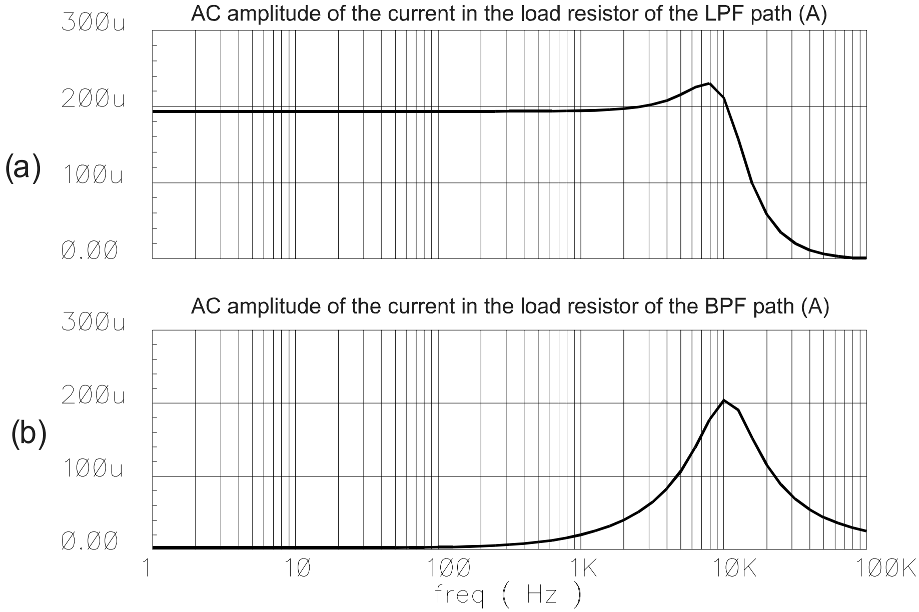


Figure 5.17: AC amplitude of the current in the load resistor of the test path without wrapper cells.

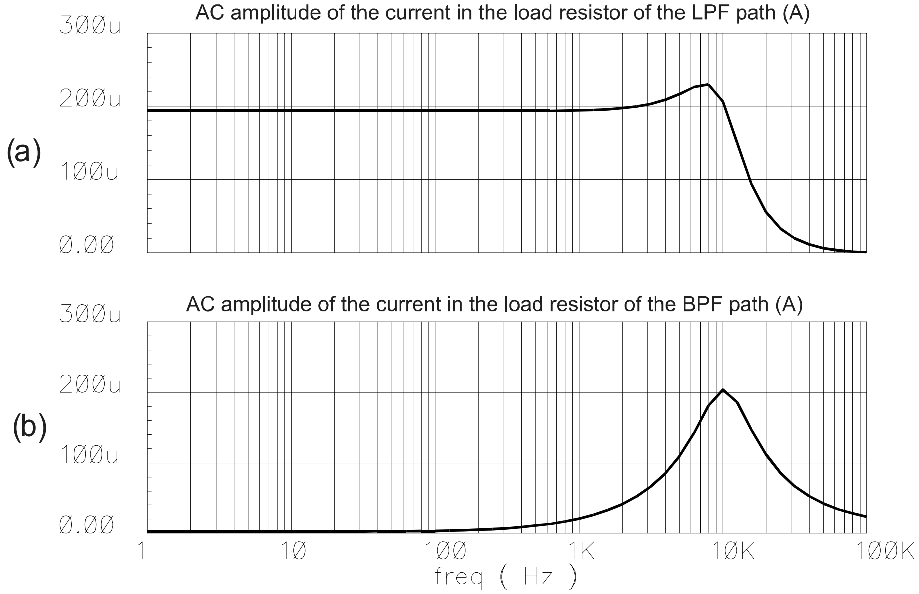


Figure 5.18: AC amplitude of the current in the load resistor of the test path with wrapper cells.

In order to clearly observe the difference between the AC simulation results of the test path with and without wrapper cells, the difference of these two cases is depicted in Figure 5.19 by subtracting the results shown in Figure 5.17 by the corresponding results presented in Figure 5.18. Figure 5.19(a) shows the difference of the AC amplitude of the current in the load resistor connected to the VIC output of the LPF path. The maximum absolute value of the difference shown in Figure 5.19(a) is around 700nA. Since the normal AC amplitude shown in Figure 5.17(a) is around 200uA, the largest difference is around 0.35%. Similarly, by using the data presented in Figure 5.17(b) and Figure 5.19(b), the biggest difference for the AC amplitude of the current in the load resistor connected to the VIC output of the BPF path can be calculated to be around 0.3%.

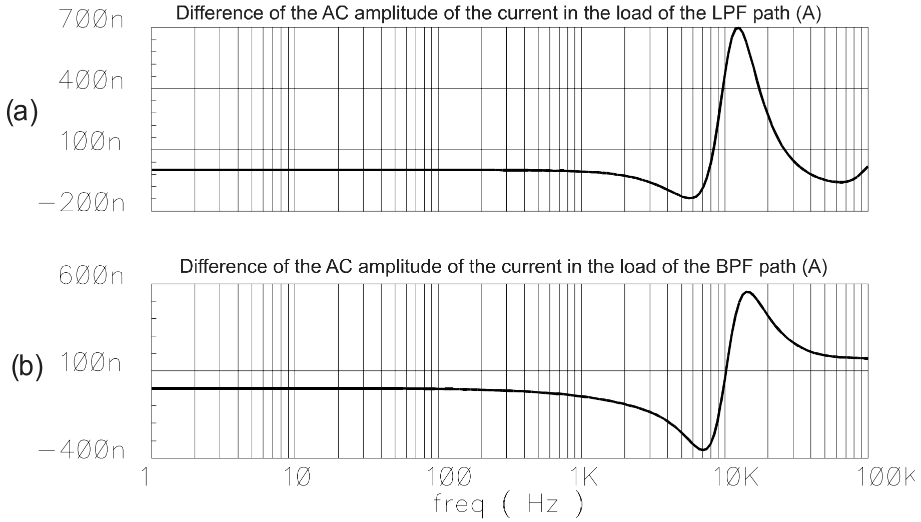


Figure 5.19: The difference between the AC amplitude of the current in the load resistor of the test path with/without wrapper cells in the LPF path (a) and BPF path (b).

The AC phases of the corresponding AC simulations are shown in Figure 5.20, Figure 5.21 and Figure 5.22. Similar to the simulation results of the AC amplitude, the AC phases of the simulation without wrapper cells are shown in Figure 5.20, AC phases of the simulation with wrapper cells are shown in Figure 5.21, and the differences between them are shown in Figure 5.22.

As can be seen from Figure 5.22(a), the difference of the phase of the current in the load resistor of the LPF path is less than 4 degrees for a frequency lower than 50kHz. The cutoff frequency of the LPF is 18kHz. Therefore, the wrapper cells and the analogue test path have a very small influence in terms of AC phase within the band of the LPF. The difference for the BPF path is depicted in Figure 5.22(b). It shows that the difference is less than 4 degrees if the frequency is lower than 100kHz. This means the influence of the proposed architecture on the performance of the BPF path in terms of AC phase is also very small.

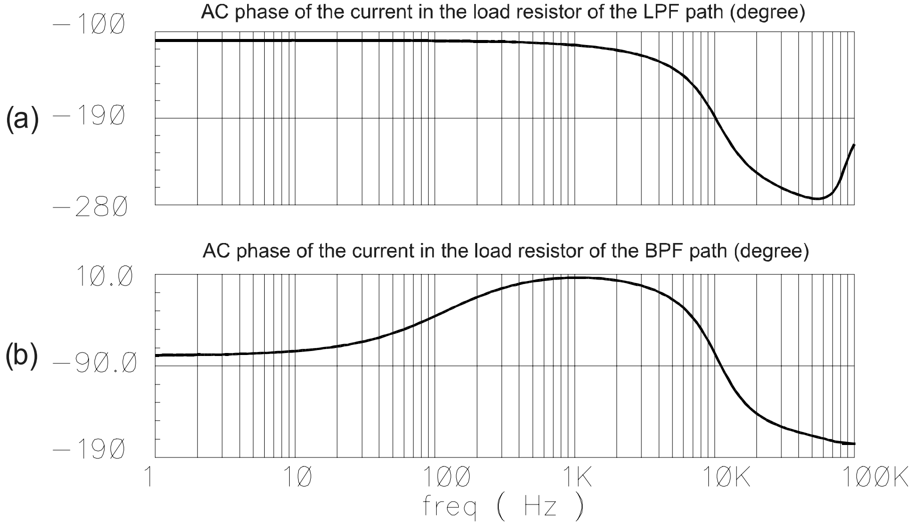


Figure 5.20: AC phase of the current in the load resistor of the test path without wrapper cells.

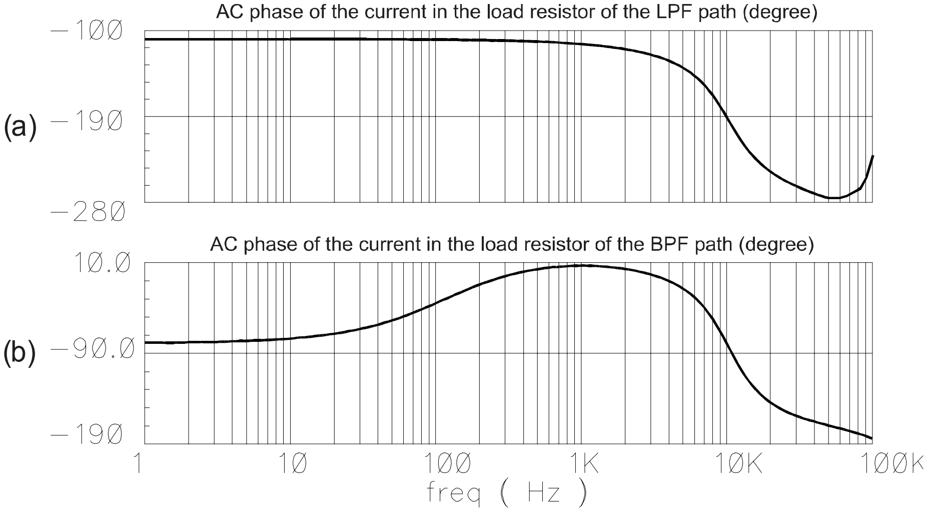


Figure 5.21: AC phase of the current in the load resistor of the test path with wrapper cells.

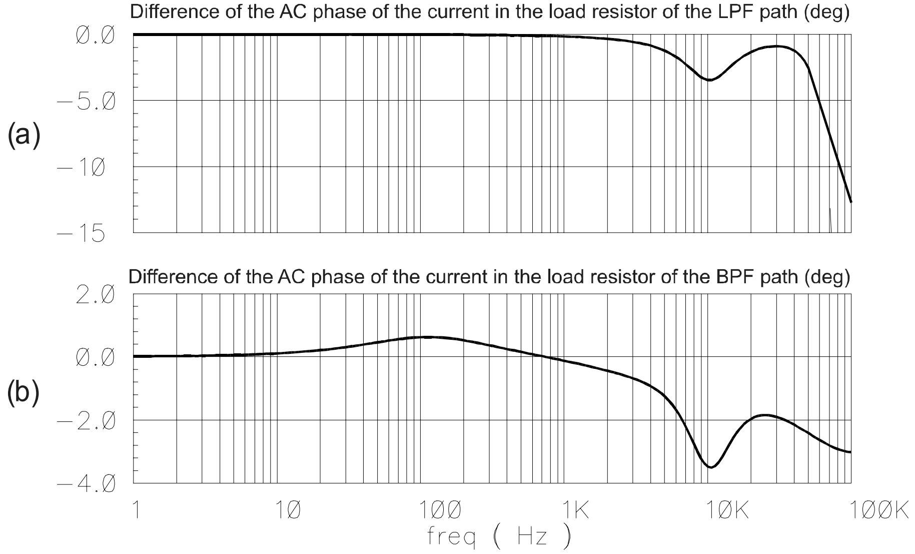


Figure 5.22: The difference between the AC phases of the current in the load resistor of the test path with/without wrapper cells.

The DC and AC simulation results show that the performance degradation of introducing the proposed architecture is quite small. Therefore, the proposed architecture is acceptable for using in the example circuit in terms of the performance degradation.

5.4.3 Simulation results for the embedded core testing case

The results in the previous subsection have shown that the influence of the proposed architecture on the normal operation of our SoC is quite small. As the main purpose of the proposed architecture is embedded core testing, a simulation of this test using the proposed architecture is carried out in this subsection. Without loss of any generality, it is assumed that the FILTER is the analogue embedded core under test. For the stand-alone FILTER, the testability-analysis based test-pattern generation approach is used to generate the potential test signals [Sta02a, Sta02b]. Among these potential test signals, the test input signal for one particular bridging fault is the combination of two sine waves, one 13.3kHz signal (AC amplitude is 50mV and DC offset is 1.3V) and one 17.7kHz signal (AC amplitude is 50mV and DC offset is 1.3V) [Sta03]. Since the input and output of the FILTER cannot be directly accessed at the primary chip input and output, our DfT architecture is employed to provide the extra test access.

The example circuit with the proposed core-based testing architecture is shown in Figure 5.13. In order to test the FILTER, all input wrapper cells and output wrapper cells of the FILTER operate in the "core test" mode. During the simulation, the combined sine signal (13.3kHz and 17.7kHz) described previously is applied to the input of the AITB (Figure 5.13). Then, the signal at the output of the AOTB is observed for both the faulty circuit and fault-free circuit. For the purpose of comparison, the corresponding signals at the output of the FILTER are also shown. The simulation results are presented in Figure 5.23. In this figure, the x-axis represents the time for the transient analysis. Figure 5.23(a) shows the waveforms observed at the output of the FILTER for both the fault-free circuit and faulty circuit. Figure 5.23(b) shows the signals observed at the output of the AOTB during the fault simulation.

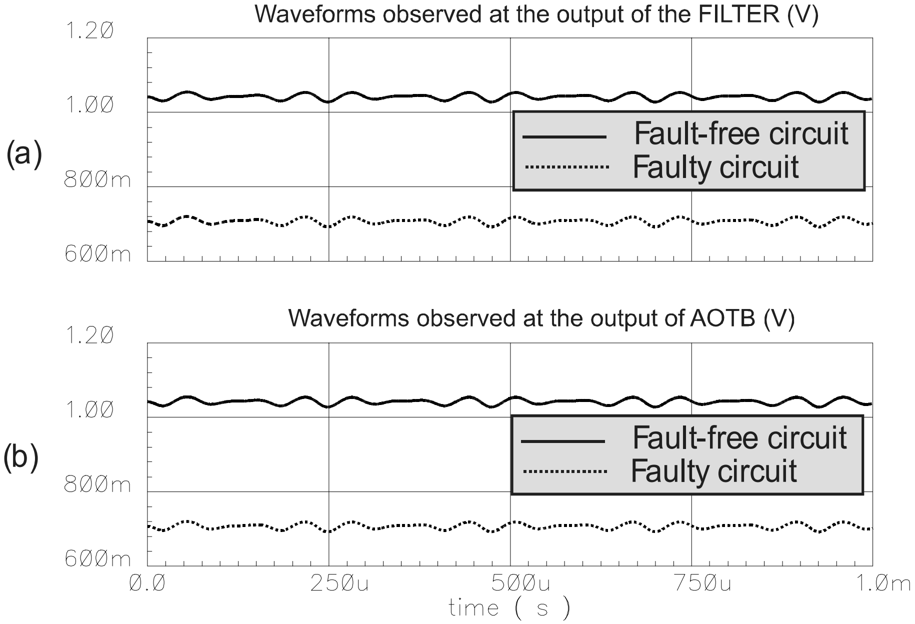


Figure 5.23: Simulation results of embedded core testing for one bridging fault in the FILTER.

First of all, if the test input signal is applied via the proposed architecture (AITB and the analogue input wrapper cells), the fault can still be detected at the output of the FILTER since there is a clear distinction between the response of the faulty circuit and fault-free circuit. This means that the proposed architecture can provide the extra test access from the primary SoC inputs to the input of the embedded FILTER.

Second, Figure 5.23(b) shows that there is also a clear distinction between the behaviour of the faulty circuit and behaviour of the fault-free circuit at the AOTB output. Therefore, the fault can also be successfully detected at the primary chip outputs using the proposed output wrapper cells and the analogue test bus AOTB. This means that the proposed architecture can provide test access of the observation at the primary SoC outputs for the embedded core testing case.

Summarizing, the presented simulation results indicate that the proposed architecture can successfully provide the test access for embedded analogue core testing in the low-frequency circuits. However, as a limitation of this approach, there might be the performance degradation problem if it is applied to RF circuit testing.

5.4.4 Simulation results for the interconnection testing case

Another part of the required tests for an embedded analogue core is the verification of the quality of the interconnection lines between cores. The simulation result for the case that a bad connection (e.g. caused by vias) is present between the FILTER and the VIC is presented in Figure 5.24. During the simulation, the output wrapper cells of the FILTER and the input wrapper cells of the VIC operate in the "interconnection test" mode with the switch status listed in Table 5.1 and Table 5.2. In this example, a voltage-to-voltage measurement has been employed. A DC voltage source is connected at the input of the AITB and the DC voltage at the output of AOTB is observed.

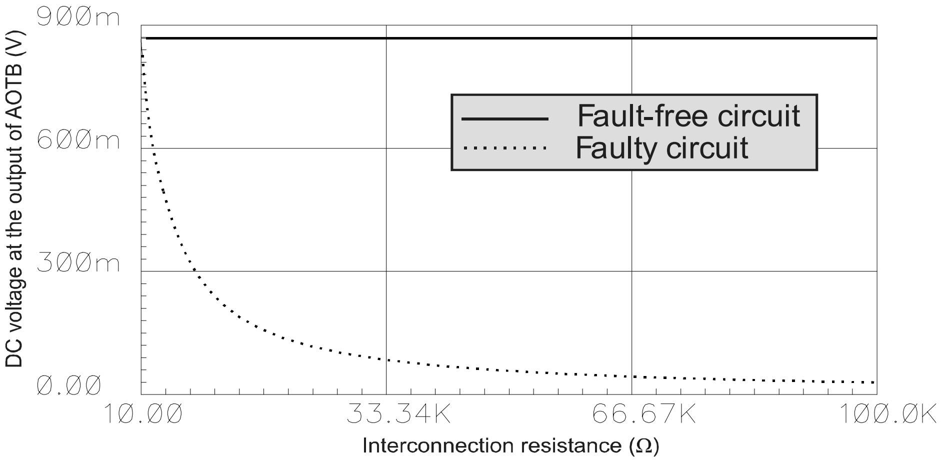


Figure 5.24: Simulation results showing an increased resistance between the FILTER and the VIC interconnection which is detected by employing the proposed core-based testing architecture. (Load resistance at AOTB = 1kΩ).

In Figure 5.24, the x-axis represents the assumed interconnection resistance and the y-axis represents the DC voltage measured at the output of the AOTB. The interconnection resistance is ideally close to zero if there is no interconnection fault (fault-free circuit). The results in Figure 5.24 show that the difference between the behaviour of the fault-free circuit and the faulty circuit become increasingly large as the interconnection resistance increases. Due to this difference between the behaviour of the faulty circuit and the fault-free circuit as shown in Figure 5.24, the interconnection fault can be detected by employing the proposed architecture.

5.4.5 Silicon area overhead of the proposed architecture

In order to calculate the overhead of the proposed architecture, it is not fair to only look at the number of transistors. The reason is that the transistors used in the analogue part of the example SoC have much larger dimensions than the minimum transistors used in digital circuits. For instance, half of the amount of transistors used in the FILTER have a gate width of 250µm and a length of 5µm, while the transistor used in our digital circuits have a gate width of 5.3µm and length of 0.8µm in our process (0.8µm AMS). Therefore, the silicon area overhead is used to evaluate our proposed architecture. For the sake of simplicity, the gate width and length of the transistors are directly multiplied to calculate the relative silicon overhead in our research. This means, the relative silicon area of the transistor with width of 250μm and length of 5μm is 1250μm^2.

The relative silicon areas of the analogue cores in the test path are listed in Table 5.3. The relative silicon area of the DfT architecture is listed in Table 5.4. There are 6 input analogue wrapper cells and 6 output analogue wrapper cells used in our test path example. Three WIRs are employed because each core requires one WIR. The TAP controller is not listed in Table 5.4 because there is only one TAP controller for the whole chip and it provides the function for all analogue and digital wrapper cells in the chip. Actually, the design in [Hui02] shows that the relative silicon area of the TAP controller is only 2035μm^2.

Table 5.3: The relative silicon area of the embedded cores.

Circuit type	**Relative silicon area of each core (μm^2)**	**Total relative silicon area of the embedded cores (μm^2)**
Pre-amplifier	32342	**139359**
FILTER	42332	
VIC	64685	

Table 5.4: The relative silicon area of the DfT circuitry.

DfT circuit type	**Relative silicon area of each type DfT circuit (μm^2)**	**Total relative silicon area of the whole DfT circuit (μm^2)**
Analogue input wrapper cell (6x)	6945	**17782**
Analogue output wrapper cell (6x)	6843	
WIR (3x)	3994	

With the data listed in Table 5.3 and Table 5.4, the overhead of the proposed core-based testing architecture can be calculated to be 12.8%. It is claimed in [Bus01] that the silicon area overhead of digital DfT is typically found to be in the range of 5% to 10%. However, the results in [teB02] show that an area overhead of 25% is already quite good for the DfT circuits in the testing of digital asynchronous circuits. Therefore, the silicon area overhead of our proposed core-based testing architecture seems to be acceptable for the analogue testing of embedded cores.

5.5 Conclusions

In this chapter, a new mixed-signal P1500-compatible core-based testing architecture has been proposed for the first time. The digital cores with the P1500 test wrapper can be directly used in this architecture to accomplish extra test access. The new analogue input and output wrapper cells and analogue test buses have been designed and used to provide the test access for the analogue testing of embedded mixed-signal cores. In order to verify the viability of this proposed new architecture and associated circuits, an analogue part of a locally designed low frequency mixed-signal SoC has been used as an example. The simulation results show that the proposed architecture has an extremely small influence on the normal operation of the example circuit. The maximal distortion with regard to the DC performance and AC amplitude are 0.0125% and 0.35% respectively. The simulation results also show that the proposed mixed-signal core-based testing architecture can provide test access for the testing of embedded mixed-signal cores and the interconnection testing successfully. Moreover, the calculated silicon area overhead results for this DfT architecture show that the overhead is acceptable for analogue embedded core testing.

5.6 References:

[11494] Homepage of IEEE 1149.4 mixed-signal test bus working group, http://grouper.ieee.org/groups/1149/4/

[Adh99] S. Adham et al., “Preliminary outline of the IEEE P1500 scalable architecture for testing embedded cores,” *Proc. of VLSI Test Symposium (VTS)*, pp. 483-488, 1999.

[Agr98] V. D. Agrawal, “Design of mixed-signal systems for testability,” *Integration, the VLSI Journal*, vol. 26, pp. 141-150, 1998.

[Ber02] M. H. ten Berge, “Structural design for core based analog testing – analogue infrastructure,” Research report, University of Twente, the Netherlands, 2002.

[Ble93] H. Bleeker, P. van den Eijnden and F. de Jong, “Boundary-scan test: a practical approach,” Kluwer Academic Publisher, ISBN 0-7923-9296-5, 1993.

[Bus01] M. L. Bushnell and V. D. Agrawal, “Essentials of electronic testing for digital, memory & mixed-signal VLSI circuit,” Kluwer Academic Publishers, ISBN: 0-7923-7991-8, 2001.

[Cro97] A. Cron, “IEEE P1149.4 – almost a standard,” *Proc. of International Test Conference*, pp. 183-190, 1997.

[Dol02] Dolphin Integration, "Jazz Line of analog and mixed-signal virtual components," http://www.dolphin.fr/flip/analog/analog_overview.html

[Fan00] L. Fang, M. Stancic and H. G. Kerkhoff, “Mixed-signal core-based testing,” *Inf. Proc. of European Test Workshop (ETW) 2000*, pp. 279-280, May 2000.

[Hui02] M. C. Huijgen, “Structural design for core based analog testing – digital infrastructure,” Research report, University of Twente, the Netherlands, 2002.

[Jar91] M. Jarwala and S. J. Tsai, "A framework for design for testability of mixed analog/digital circuits," *Proc. of IEEE Custom Integrated Circuits Conference*, pp. 13.5.1-13.5.4, 1991.

[Nag99] N. Nagi, “System-on-a-chip mixed-signal test: issues, current industry practices and future trends,” *Proc. of International Mixed Signal Testing Workshop*, pp. 201-211, 1999.

[Oze00] S. Ozev, I. Bayraktaroglu and A. Orailoglu, "Test synthesis for mixed-signal SoC paths," *Proc. of Design, Automation and Test in Europe (DATE) conference*, pp. 128-133, 2000.

[P1500] Homepage of IEEE P1500 working group, http://grouper.ieee.org/groups/1500/.

[Par99] K.P. Parker, “The Boundary-scan handbook: analog and digital,” Kluwer Academic Publisher, ISBN 0-7923-8277-3, 1999.

[Rob97] G. W. Roberts, "DFT techniques for mixed-signal integrated circuits," *Proc. of Circuits and systems in the information age*, pp. 251-271, 1997.

[Sta00] M. Stancic, L. Fang and H. G. Kerkhoff, "Detecting parametric faults in embedded cores of mixed-signal ICs" *Proc. of 6th IEEE International Mixed-Signal Testing Workshop (IMSTW),* pp. 20-21, June 2000.

[Sta02a] M. Stancic and H. G. Kerkhoff, “Test generation for mixed-signal circuits using testability analysis,” *Inf. proceedings of the 7th European Test Workshop (ETW02)*, pp. 249-254, 2002.

[Sta02b] M. Stancic, L. Fang, M. Weusthof, R. Tijink and H. Kerkhoff, “ A new test generation for embedded analogue cores in SoC,” *Proc. of International Test Conference (ITC)*, Baltimore, USA, pp. 861-869, 2002.

[Sta03] M. Stancic, “ Test selection for analogue cores based on testability transfer function,” Ph.D. thesis, University of Twente, Enschede, the Netherlands, 2003.

[Sun98] S. Sunter, “Chapter 7: Implementing the 1149.4 standard mixed-signal test bus,” *Analog and mixed-signal test*, Editor: B. Vinnakota, Prentice Hall, ISBN: 0-13-786310-1, 1998.

[teB02] F. te Beest, A. Peeters, M. Verra, K. van Berkel and H. G. Kerkhoff, “Automatic Scan insertion and test generation for asynchronous circuits,” *Proc. of International Test Conference (ITC)*, Baltimore, USA, pp. 804-813, 2002.

[Yuf99] A. Yufera and A. Rueda, “Programmable low-voltage continuous time filter for audio applications*”, Proceedings of IEEE International Symposium on Circuits and Systems*, Orlando, USA, pp. 200-203, 1999.

[Wag88] K. D. Wagner and T. W. Williams, “Design for testability of mixed signal integrated circuits,” *Proc. of International Test Conference*, pp. 823-828, 1988.

Chapter 6

Conclusions and recommendations

6.1 Conclusions

The objective of this thesis is to find solutions for some crucial problems in the system-level testing of embedded analogue cores in SoC. Due to the complexity of mixed-signal SoCs, a hierarchical approach has been employed in our research for the testing of embedded analogue cores in SoC. In this approach, a test set for each stand-alone core in the SoC is generated first. Next, some test translation schemes are employed to translate the core-level test into a system-level test because the test-input signals can only be controlled at the primary SoC inputs and the test responses can only be observed at the primary SoC outputs during production testing. The main task of the test translation is to propagate (backwards and forwards) those core-level test-input signals and test-output responses of each embedded core to the primary SoC inputs and outputs. Meanwhile, the corresponding tolerance boxes should also be propagated to the primary SoC outputs.

A new test-pattern generation method based on the analogue testability analysis, for the stand-alone core has been proposed in [Sta03]. The following crucial issues of the test translation are dealt with in this thesis:

— analogue test-signal backtrace

— tolerance-box propagation

— mixed-signal P1500-compatible core-based testing architecture

— mixed-level modelling technique for reducing the analogue fault-simulation time.

● *Analogue test-signal backtrace*

Test-signal backtracing is the procedure to determine a stimulus at the primary SoC inputs that will produce the desired test signal at the inputs of the embedded cores under test. This is a major step in translating the core-level test patterns into system-level test patterns. While digital backtrace procedures are relatively easy to implement with the help of Boolean mathematics, there is no generic approach for backtracing analogue signals in a mixed-signal IC because the analogue test signal is continuous and the respective input-output relationships of the components used in analogue circuits are often expressed by the nonlinear equations.

The PID (Proportional Integral Derivative) feedback loop based backtrace method proposed in *Chapter 3* provides a good solution to the test-signal backtracing for the testing of embedded analogue cores. With this proposed method, the test-input signal for the embedded core can be backtraced to the primary SoC inputs in the time as well as frequency domain. Moreover, the computational effort for the backtrace procedure is low since it is quite easy to implement the PID controller in a high-level language during simulation. The presented theoretical analysis, simulation and measurement results show that the test-signals of the embedded cores can be backtraced successfully to the primary SoC inputs.

- *Tolerance-box propagation*

The tolerance box plays a very important role in analogue testing because the specification of an analogue circuit allows some variations in the circuit characteristics. Tolerance-box generation is a very CPU-time consuming procedure with the conventional Monte-Carlo approach. During the test translation, the tolerance-box at the output of the embedded core under test should be propagated forwards together with the test response to the primary SoC outputs.

Chapter 2 of this thesis proposes a new fast core-based tolerance-box generation and propagation approach for the testing of embedded analogue cores by using the sensitivity-analysis technique. The application of this approach to an example circuit shows that this approach is about 140 times faster than the conventional Monte Carlo simulation approach while the loss in the accuracy is smaller than 1.4%. Therefore, the proposed approach is very useful for the tolerance-box propagation in the testing of embedded analogue cores.

- *Mixed-signal P1500-compabitable core-based testing architecture*

The testabilities of the embedded cores are limited by the behaviour of other preceding/succeeding cores in the SoC. It is possible that the test patterns generated for stand-alone cores cannot be successfully translated to the system-level test patterns due to limited test access.

Chapter 5 proposes a new mixed-signal P1500-compatible core-based testing architecture. The digital cores with the P1500 test wrapper can be directly used within this architecture to enable extra test access because of its good compatibility. The new analogue input- and output- wrapper cells and analogue test buses have been designed and used to provide the test access for the analogue testing of embedded mixed-signal cores. In order to verify the viability of this proposed new architecture and the associated circuits, an analogue part of a locally designed mixed-signal SoC has been used as an example. The simulation results show that the proposed architecture has a very small influence on the normal operation of the example circuit. The simulation results also show that the proposed mixed-signal core-based testing architecture can successfully provide the test access for the testing of embedded mixed-signal cores as well as testing the interconnection between cores. Moreover, the calculated silicon area overhead shows that the overhead is acceptable for the analogue testing of embedded mixed-signal cores.

- *Mixed-level modelling technique for reducing the analogue fault-simulation time*

Structural fault-model based testing is a potential approach to develop low-cost test signals for analogue circuit testing. In this approach, the manufacturing defects are modelled with certain fault models. During the test-signal selection procedure, these fault models are first inserted into the circuit netlist to construct the faulty circuits. Then, the analogue fault simulation is employed to select the test signals and evaluate the performance of the test signals in terms of fault coverage.

Due to the fact that the list of faulty circuits is always very large and analogue circuit simulation is CPU-time consuming, analogue fault simulation becomes unacceptable long if the analogue circuit becomes more complex. One solution to reduce the massive fault-simulation time is to use high-level models for the faulty-free blocks/cores in the fault simulation. Crucial for this method is the generation of suitable high-level models for the fault-free analogue functional blocks/cores.

In *Chapter 4*, a new mixed-level modelling technique with three stages is proposed to speed up analogue fault simulation. In this structure, the original transistor-level circuits are reused for the input stage and output stage. The functional stage is an equation-based part to represent the function of the original fault-free block. This technique has been applied to an analogue block in a real-life industrial mixed-signal chip. The simulation results indicate that this kind of models can work properly in a fault-simulation environment.

Summarising, during the test translation, the test signal of the embedded cores can be backtraced by using the proposed PID feedback loop-based approach and the tolerance-boxes can be quickly propagated forwards while applying the proposed sensitivity-analysis approach. The proposed P1500-compatible core-based testing architecture can provide the extra test access for the embedded cores with low testability. Therefore, the core-level test pattern for embedded analogue cores can be successfully translated into the system-level by combining the methods presented in this thesis.

6.2 Original contributions of this thesis

The following original contributions have been proposed in this thesis.

● A PID feedback loop-based test-signal backtrace approach, which provides an effective method to backtrace the test signals of the embedded cores to the primary SoC inputs.

● A sensitivity-analysis based tolerance-box generation and propagation method to effectively propagate the tolerance-boxes at the output of the embedded cores to the primary SoC outputs.

● A mixed-level modelling technique for fault-free blocks/cores that can reduce the analogue fault-simulation time while retaining the same results as the conventional CPU-time consuming fault simulation at transistor level.

● A mixed-signal P1500-compatible core-based testing architecture to provide extra test access for embedded cores with low testability.

6.3 Recommendations for future research

The main recommendations for future research in the area of testing mixed-signal embedded cores in SoC are as following:

♦ Many results presented in this thesis have been obtained by simulation. More measurements can be carried out to optimise the methods proposed in this thesis in future research.

♦ It is assumed that the unexpected behaviour of the faulty preceding/succeeding blocks/cores can be isolated by reusing the input and output interface circuit of the original design. The limitations of isolation have to be investigated in more details in future research.

♦ In this thesis, the testing of embedded analogue cores in SoC has been discussed. In a mixed-signal SoC, there are also other important types of cores, being ADC and DAC. How to test embedded data converters in SoCs is an issue for future research.

♦ The target vehicles of this research work are (embedded) audio/video range analogue and mixed-signal cores in CMOS telecommunication ICs. Due to the expanding wireless applications, the market demand for RF ICs is becoming increasingly strong. How to apply the method developed in this research to RF chip testing is still an open question.

♦ A mixed-signal SoC always has an on-chip microprocessor and ADC and/or DAC cores. Therefore, the possibility to use those already existing resources to build BIST (Build In Self Test) circuitry for analogue cores has to be investigated in the future.

6.4 Reference

[Sta03] M. Stancic, “Test selection for analogue cores based on testability transfer function,” Ph.D. thesis, University of Twente, Enschede, the Netherlands, 2003.

Appendix A

Parameter-deviation model and tolerance-box propagation algorithm

In this appendix, more theoretical details about the parameter-deviation model and the tolerance-box propagation algorithm are presented.

A.1 Parameter-deviation model

In this section, how to use the parameter-deviation model proposed in Chapter 2 (Figure 2.2) to model the correlation between the physical parameter deviations of different cores is discussed.

For a general case, assume that there are M embedded analogue cores in the tolerance-box propagation path in the mixed-signal SoC. Without lose of any generality, one parameter deviation Δp_k is considered. For the core j $(j = 1,2,\cdots,M)$, the real physical parameter deviation is $\Delta p_k^{R,j}$. Here "*R*" is used to indicate "real physical parameter deviation". For this parameter deviation, the corresponding global parameter deviation is Δp_k^G and the local parameter deviations are $\Delta p_k^j (j = 1,2,\cdots,M)$ in the proposed parameter-deviation model. The real physical parameter deviations can now be expressed as:

$$\Delta p_k^{R,j} = a_j \cdot \Delta p_k^G + b_j \cdot \Delta p_k^j \quad (j = 1,2,\cdots,M) \tag{A.1}$$

where $a_j (j = 1,2,\cdots,M)$ and $b_j (j = 1,2,\cdots,M)$ are constant coefficients. Using equation (A.1), the mean value of these real parameter deviations can be derived as [Pee93]:

$$\mu_k^{R,j} = a_j \cdot \mu_k^G + b_j \cdot \mu_k^j \quad (j = 1,2,\cdots,M) \tag{A.2}$$

where μ_k^G and μ_k^j are the mean values of Δp_k^G and $\Delta p_k^j (j = 1,2,\cdots,M)$. Since Δp_k^G and $\Delta p_k^j (j = 1,2,\cdots,M)$ are mutually independent in the parameter-deviation model, using equation (A.1), the variance of $\Delta p_k^{R,j}$ can be expressed as [Pee93]:

$$(\sigma_k^{R,j})^2 = (a_j)^2 \cdot (\sigma_k^G)^2 + (b_j)^2 \cdot (\sigma_k^j)^2 \quad (j = 1,2,\cdots,M) \tag{A.3}$$

where $(\sigma_k^G)^2$ and $(\sigma_k^j)^2$ are the variances of Δp_k^G and $\Delta p_k^j (j = 1,2,\cdots,M)$ respectively.

The covariance C_{XY} between two random variables X and Y can be used to indicate the correlation between them and is defined as [Pee93]:

$$C_{XY} = E[(X - \mu_X) \cdot (Y - \mu_Y)] \tag{A.4}$$

where $E[(X - \mu_X) \cdot (Y - \mu_Y)]$ is the mean value of the random variable. Using equation (A.4), the covariance between any two real parameter deviations can be calculated with the following equation:

$$C_{j1,j2}^R = E[(\Delta p_k^{R,j1} - \mu_k^{R,j1}) \cdot (\Delta p_k^{R,j2} - \mu_k^{R,j2})] \tag{A.5}$$

where $j1 = 1,2,\cdots,M$, $j2 = 1,2,\cdots,M$ and $j1 \neq j2$. With equations (A.1), (A.2) and (A.5) and the fact that Δp_k^G and $\Delta p_k^j (j = 1,2,\cdots,M)$ are mutually independent, the covariance between $\Delta p_k^{R,j1}$ and $\Delta p_k^{R,j2}$ can be derived as:

$$C_{j1,j2}^R = a_{j1} \cdot a_{j2} \cdot (\sigma_k^G)^2 \tag{A.6}$$

Additionally, the correlation coefficient ρ_{XY} between two random variables X and Y can be used to measure the degree of correlation between them and is defined as [Pee93]:

$$\rho_{XY} = \frac{C_{XY}}{\sigma_X \cdot \sigma_Y} \tag{A.7}$$

where σ_X and σ_Y are the standard deviation of X and Y. Using equations (A.6) and (A.7), the correlation coefficient between $\Delta p_k^{R,j1}$ and $\Delta p_k^{R,j2}$ can be derived as:

$$\rho_{j1,j2}^R = \frac{a_{j1} \cdot a_{j2} \cdot (\sigma_k^G)^2}{\sigma_k^{R,j1} \cdot \sigma_k^{R,j2}} \tag{A.8}$$

where $\sigma_k^{R,j1}$ and $\sigma_k^{R,j2}$ can be calculated by using equation (A.3).

A.2 Tolerance-box propagation algorithm

In this section, the detailed mathematical derivation of the tolerance-box propagation algorithm is given.

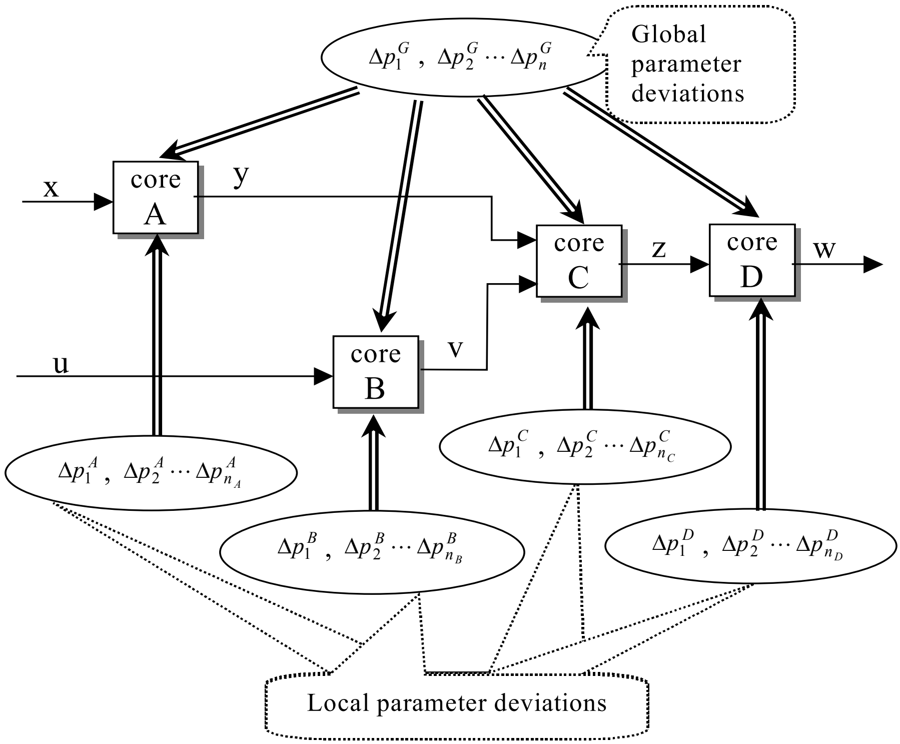

Δp_1^G , $\Delta p_2^G \cdots \Delta p_n^G$
Global parameter deviations
x
core A
y
core C
z
core D
w
u
core B
v
Δp_1^A , $\Delta p_2^A \cdots \Delta p_{n_A}^A$
Δp_1^B , $\Delta p_2^B \cdots \Delta p_{n_B}^B$
Δp_1^C , $\Delta p_2^C \cdots \Delta p_{n_C}^C$
Δp_1^D , $\Delta p_2^D \cdots \Delta p_{n_D}^D$
Local parameter deviations

Figure A.1: Illustration of the system structure with different categories of parameter deviations.

For the example system structure shown in Figure A.1, by using equations (2.26) and (2.45), the deviation of the output y of core A under all the global and local parameter deviations and the input signal deviation can be expresses as the following equation:

$$\Delta y = \sum_{i=1}^{n} (s_{G,p_i}^{A} \cdot \Delta p_i^G) + \sum_{i=1}^{n_A} (s_{p_i}^{A} \cdot \Delta p_i^A) + s_x^A \cdot \Delta x \tag{A.9}$$

where $s_{G,p_i}^{A} (i = 1,2,\cdots,n)$ are the sensitivities for the output of the core A with respect to the global parameter deviations $\Delta p_i^G (i = 1,2,\cdots,n)$ and $s_{p_i}^{A} (i = 1,2,\cdots,n_A)$ are the sensitivities for the output of the core A with respect to the local parameter deviations $\Delta p_i^A (i = 1,2,\cdots,n_A)$, and s_x^A is the core sensitivity with respect to the core input x.

Similarly, the deviation of the output v of core B under all the deviations can be expressed as:

$$\Delta v = \sum_{i=1}^{n} (s_{G,p_i}^{B} \cdot \Delta p_i^G) + \sum_{i=1}^{n_B} (s_{p_i}^{B} \cdot \Delta p_i^B) + s_u^B \cdot \Delta u \tag{A.10}$$

where $s_{G,p_i}^{B}(i=1,2,\cdots,n)$ and $s_{p_i}^{B}(i=1,2,\cdots,n_B)$ are the sensitivities for the output of the core B with respect to the global parameter deviations $\Delta p_i^G(i=1,2,\cdots,n)$ and its local parameter deviations $\Delta p_i^B(i=1,2,\cdots,n_B)$, and s_u^B is the core sensitivity with respect to the core input u.

For core C, let $s_{G,p_i}^{C}(i=1,2,\cdots,n)$ and $s_{p_i}^{C}(i=1,2,\cdots,n_C)$ be the sensitivities for the output of the core C with respect to the corresponding global and local parameter deviations, and s_y^C and s_v^C be the core sensitivity with respect to the core input y and v. With equation (2.26) and the core sensitivity defined in equation (2.45), the deviation of the output of the core C under all parameter deviations and input deviations can be derived to be:

$$\Delta z=\sum_{i=1}^{n}(s_{G,p_i}^{C}\cdot\Delta p_i^G)+\sum_{i=1}^{n_C}(s_{p_i}^{C}\cdot\Delta p_i^C)+s_y^C\cdot\Delta y+s_v^C\cdot\Delta v \qquad \text{(A.11)}$$

Therefore by substituting equations (A.9) and (A.10) one can obtain:

$$\Delta z=\sum_{i=1}^{n}\left(s_{G,p_i}^{C}+s_y^C\cdot s_{G,p_i}^{A}+s_v^C\cdot s_{G,p_i}^{B}\right)\cdot\Delta p_i^G+\sum_{i=1}^{n_A}(s_y^C\cdot s_{p_i}^{A}\cdot\Delta p_i^A)$$

$$+\sum_{i=1}^{n_B}(s_v^C\cdot s_{p_i}^{B}\cdot\Delta p_i^B)+\sum_{i=1}^{n_C}(s_{p_i}^{C}\cdot\Delta p_i^C)+s_x^A\cdot s_y^C\cdot\Delta x+s_u^B\cdot s_v^C\cdot\Delta u \qquad \text{(A.12)}$$

Finally for core D, by using a similar derivation for equation (A.9), the output deviation under all deviations can be derived as:

$$\Delta w=\sum_{i=1}^{n}(s_{G,p_i}^{D}\cdot\Delta p_i^G)+\sum_{i=1}^{n_D}(s_{p_i}^{D}\cdot\Delta p_i^D)+s_z^D\cdot\Delta z \qquad \text{(A.13)}$$

Then by using equation (A.12) and equation (A.13), one can get:

$$\Delta w=\sum_{i=1}^{n}\left(s_{G,p_i}^{D}+s_z^D\cdot s_{G,p_i}^{C}+s_z^D\cdot s_y^C\cdot s_{G,p_i}^{A}+s_z^D\cdot s_v^C\cdot s_{G,p_i}^{B}\right)\cdot\Delta p_i^G$$

$$+\sum_{i=1}^{n_A}(s_z^D\cdot s_y^C\cdot s_{p_i}^{A}\cdot\Delta p_i^A)+\sum_{i=1}^{n_B}(s_z^D\cdot s_v^C\cdot s_{p_i}^{B}\cdot\Delta p_i^B)+\sum_{i=1}^{n_C}(s_z^D\cdot s_{p_i}^{C}\cdot\Delta p_i^C)$$

$$+\sum_{i=1}^{n_D}(s_{p_i}^{D}\cdot\Delta p_i^D)+s_x^A\cdot s_y^C\cdot s_z^D\cdot\Delta x+s_u^B\cdot s_v^C\cdot s_z^D\cdot\Delta u \qquad \text{(A.14)}$$

In the above equation, $s_{G,i}^{D}(i=1,2,\cdots,n)$ and $s_{p_i}^{D}(i=1,2,\cdots,n_C)$ are the sensitivities for the output of the core D with respect to the global parameter deviations and local parameter deviations respectively, and s_z^D is the core sensitivity of core D.

According to equations (2.27) and (2.28), the final tolerance box for the whole test path containing core A, core B, core C and core D with the structure shown in Figure A.1 can then be expressed by:

$$[w_0 + \min\{\Delta w\}, w_0 + \max\{\Delta w\}] \tag{A.15}$$

where w_0 is the nominal value of the circuit output performance with nominal parameters.

In the above derivation, only four analogue cores are connected as shown in Figure A.1. For the case that more than four cores are connected in a different structure, the same propagation procedure can be employed to calculate the final tolerance-box. Assume that there are *M* analogue embedded cores in the test path, which is one portion of the whole mixed-signal SoC. This test path has *K* inputs and *L* outputs as shown in Figure A.2. The input signals and output signals are $u_i (i = 1,2,\cdots,K)$ and $w_i (i = 1,2,\cdots,L)$ respectively. The connection of these *M* embedded cores is not specified in the figure because it is a general structure and the connection can be any kind of topology.

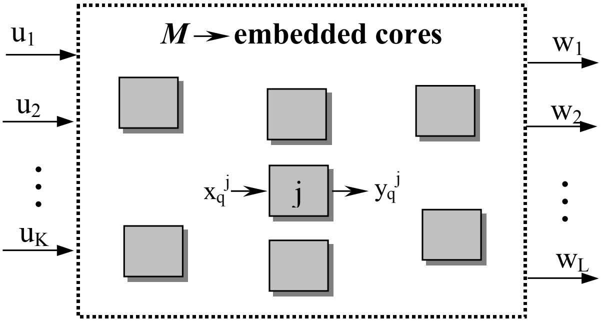


Figure A.2: Illustration of the test path with *M* embedded cores, *K* inputs and *L* outputs.

Without loss of any generality, let $\Delta p_i^G (i = 1,2,\cdots,n)$ be the global parameter deviations and $\Delta p_i^j (i = 1,2,\cdots,n_j)$ be the local parameter deviations for the *j*th $(j = 1,2,\cdots,M)$ core. Assume that the inputs and the outputs for the *j*th core are $x_q^j (q = 1,2,\cdots,I_j)$ and $y_q^j (q = 1,2,\cdots,O_j)$ respectively. By using the similar procedure for the derivation of equation (A.11), the output deviation of the *j*th core under all the parameter deviations and input deviations can be derived to be:

$$\Delta y_q^j = \sum_{i=1}^{n} (s_{G,p_i}^j \cdot \Delta p_i^G) + \sum_{i=1}^{n_j} (s_{p_i}^j \cdot \Delta p_i^j) + \sum_{i=1}^{I_j} (s_{x_i}^j \cdot \Delta x_i^j) \quad (q = 1,2,\cdots,O_j) \tag{A.16}$$

where s_{G,p_i}^j, $s_{p_i}^j$ and $s_{x_i}^j$ are the sensitivities with regard to the global parameter deviations, local parameter deviations and the input deviations of the *j*th core. Since these *M* embedded cores are connected in a certain way, the Δx_i^j in equation (A.16) can be the output deviation from the previous core or one of the input deviations $\Delta u_k \, (k = 1,2,\cdots,K)$. Therefore, using the similar procedure for the derivation of equation (A.14) and substituting Δx_i^j in equation (A.16) with the output deviation of the previous core or one of the input deviations, one can obtain the final output deviations as:

$$\begin{aligned}\Delta w_l = &\sum_{i=1}^{n} \left(f\left(s_{G,p_i}^1, s_{G,p_i}^2, \cdots, s_{G,p_i}^M, s_{x_1}^1, s_{x_2}^1, \cdots, s_{x_{I_1}}^1, s_{x_1}^2, s_{x_2}^2, \cdots, s_{x_1}^M, s_{x_2}^M, \cdots, s_{x_{I_M}}^M \right) \cdot \Delta p_i^G \right) \\ &+ \sum_{j=1}^{M} \sum_{i=1}^{n_j} \left(g_j \left(s_{p_i}^j, s_{x_1}^1, s_{x_2}^1, \cdots, s_{x_{I_1}}^1, s_{x_1}^2, s_{x_2}^2, \cdots, s_{x_1}^M, s_{x_2}^M, \cdots, s_{x_{I_M}}^M \right) \cdot \Delta p_i^j \right) \\ &+ \sum_{k=1}^{K} \left(h_k (s_{x_1}^1, s_{x_2}^1, \cdots, s_{x_{I_1}}^1, s_{x_1}^2, s_{x_2}^2, \cdots, s_{x_1}^M, s_{x_2}^M, \cdots, s_{x_{I_M}}^M) \cdot \Delta u_k \right) \quad (l = 1,2,\cdots,L) \quad \text{(A.17)}\end{aligned}$$

where the functions f, $g_j \, (j = 1,2,\cdots,M)$ and $h_k \, (k = 1,2,\cdots,K)$ are dependent on the connection of these M embedded cores. These functions can be easily found by using the similar substitution procedure as used for the derivation of equation (A.14). Similar to equation (A.15), the final tolerance box for the whole test path can be expressed as:

$$[w_l^0 + \min\{\Delta w_l\}, w_l^0 + \max\{\Delta w_l\}] \quad (l = 1,2,\cdots,L) \qquad \text{(A.18)}$$

where $w_l^0 \, (l = 1,2,\cdots,L)$ are the nominal outputs and $\Delta w_l \, (l = 1,2,\cdots,L)$ are the output deviations defined in equation (A.17).

Reference

[Pee93] P. Z. Peebles, “Probability random variables, and random signal principles,” 3rd edition, McGraw-Hill, ISBN:0-07-112782-8, 1993.

Appendix B

DC convergence of the proposed test-signal backtrace approach

In this appendix, the DC convergence of the PID feedback-loop based test-signal backtrace approach proposed in Chapter 4 will be analysed. The discussion will take place from circuit simulation point of view.

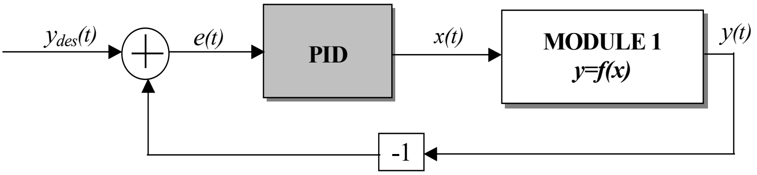


Figure B.1: Basic structure of the test-signal backtrace implementation using PID feedback loop.

The basic structure of the proposed test-signal backtrace implementation using the PID feedback loop is shown in Figure B.1. In the figure, the function of the PID controller can be described as:

$$x(t) = K_P \cdot e(t) + K_I \cdot \int_0^t e(\tau)d\tau + K_D \cdot \frac{de(t)}{dt} \qquad \text{(B.1)}$$

where $e(t)$ is the error between the desired signal $y_{des}(t)$ and the MODULE 1 output signal $y(t)$, and $x(t)$ is the output signal of PID controller. The factors K_P, K_I and K_D are the control parameters of the proportional, integral and derivative action respectively. MODULE 1 represents all the circuits between the primary SoC input and the input of the embedded core under test. We now assume that the MODULE 1 output $y(t)$ in Figure A.1 is related to its input $x(t)$ by the following nonlinear equation:

$$y(t) = f(x(t)) \qquad \text{(B.2)}$$

The error signal between the desired signal and output of MODULE 1 is:

$$e(t) = y_{des}(t) - y(t) \tag{B.3}$$

The output of the PID controller at time $t = \tau$ is

$$x(t) = K_P \cdot e(\tau) + K_I \cdot \int_0^\tau e(t)dt + K_D \cdot \frac{de(t)}{dt} \tag{B.4}$$

Without loss of any generality, assume that the time required for the previous circuit analysis is $\tau - T$, where T has a small value. Then, one can obtain:

$$\int_0^\tau e(t)dt = \int_{\tau-T}^\tau e(t)dt + \int_0^{\tau-T} e(t)dt = A + \int_{\tau-T}^\tau e(t)dt \approx A + T \cdot e(\tau) \tag{B.5}$$

where $A = \int_0^{\tau-T} e(t)dt$. Since T has a very small value, the following equation holds:

$$\frac{de(t)}{dt} = \frac{e(\tau) - e(\tau - T)}{T} \tag{B.6}$$

By using equation (B.4), (B.5) and (B.6), one can obtain:

$$x(\tau) = (K_P + K_I \cdot T + \frac{K_D}{T}) \cdot e(\tau) + B \tag{B.7}$$

where $B = K_I \cdot A - \frac{K_D \cdot e(\tau - T)}{T}$ is a known value for the circuit analysis at time $t = \tau$.

Let the gain of the PID controller be $K = K_P + K_I * T + \frac{K_D}{T}$. Therefore, the DC analysis of the circuit shown in Figure B.1 at time $t = \tau$ requires solving the following nonlinear and linear equations:

$$y = f(x) \tag{B.8}$$

$$e = y_{des} - y \tag{B.9}$$

$$x = K \cdot e + B \tag{B.10}$$

where τ is omitted in $x(\tau)$, $y(\tau)$ and $e(\tau)$ for the simplicity of writing.

A circuit simulator such as Spice usually solves the above equations iteratively by liberalizing the nonlinear system of equations at each iteration. Let x_k be the input to the circuit at the end of k th iteration. Let $x = \alpha$ be the DC input voltage that satisfies the relation

$$y_{des} = f(\alpha) \tag{B.11}$$

Let $g(x) = K \cdot [y_{des} - f(x)] + B - x$. Hence, the DC analysis procedure is to find the solution to the following equation:

$$g(x) = 0 \tag{B.12}$$

The solution for the $(k+1)$ th iteration x_{k+1} can be expressed as:

$$x_{k+1} = x_k - \frac{g(x_k)}{g^{'}(x_k)} = x_k + \frac{K \cdot [y_{des} - f(x_k)] + B - x_k}{K \cdot f^{'}(x_k) + 1} \tag{B.13}$$

where $g^{'}(x_k)$ is the derivative of function $g(x)$ with respect to x when $x = x_k$. Let $\delta_k = x_k - \alpha$ and $\delta_{k+1} = x_{k+1} - \alpha$ be the deviations from the true solution α at the end of k th and $(k+1)$ th iterations respectively. From equation (B.13), one can obtain:

$$\delta_{k+1} - \delta_k = x_{k+1} - x_k = \frac{K \cdot \left[y_{des} - f(x_k)\right] + B - x_k}{1 + K \cdot f^{'}(x_k)} \tag{B.14}$$

Applying Taylor's expansion rule $f(x_k) = f(\alpha) + f^{'}(\alpha) \cdot \delta_k + f^{''}(\alpha) \cdot \frac{\delta_k^2}{2}$, one can get:

$$\delta_{k+1} - \delta_k = \frac{K \cdot \left[y_{des} - f(\alpha) - f^{'}(\alpha) \cdot \delta_k - f^{''}(\alpha) \cdot \frac{\delta_k^2}{2} \right] + B - x_k}{1 + K \cdot f^{'}(x_k)} \tag{B.15}$$

By using equation (B.13) and $x_k = \delta_k + \alpha$, equation (B.15) can be rewritten as:

$$\delta_{k+1} - \delta_k = \frac{-K \cdot \left[f^{'}(\alpha) \cdot \delta_k + f^{''}(\alpha) \cdot \frac{\delta_k^2}{2} + \frac{\delta_k + \alpha}{K} \right] + B}{1 + K \cdot f^{'}(x_k)} \tag{B.16}$$

Similarly, by using Taylor's expansion rule $f^{'}(x_k) \approx f^{'}(\alpha) + f''(\alpha) \cdot \delta_k$ to equation (B.16), one can obtain:

$$\delta_{k+1} - \delta_k = \frac{-K \cdot \left[f^{'}(\alpha) \cdot \delta_k + f^{''}(\alpha) \cdot \frac{\delta_k^2}{2} + \frac{\delta_k + \alpha}{K} \right] + B}{1 + K \cdot [f^{'}(\alpha) + f^{''}(\alpha) \cdot \delta_k]} \tag{B.17}$$

In order to simplify eequation (B.17), one can get:

$$\delta_{k+1} = \frac{f^{''}(\alpha) \cdot \frac{\delta_k^2}{2} + \frac{B - \alpha}{K}}{\frac{1}{K} + \left[f^{'}(\alpha) + f^{''}(\alpha) \cdot \delta_k\right]} \tag{B.18}$$

Let a and b be any positive number, then their arithmetic mean is greater than or equal to their geometric mean [Abr72], i.e. :

$$a + b \geq 2\sqrt{a \cdot b}\ . \tag{B.19}$$

Since $K_I \cdot T > 0$ and $\frac{K_D}{T} > 0$, then by using equation (B.19) one can obtain:

$$K_I \cdot T + \frac{K_D}{T} \geq 2\sqrt{K_I \cdot K_D} \ . \tag{B.20}$$

Hence with regard to the gain of the PID controller $K = K_P + K_I \cdot T + \frac{K_D}{T}$, one can have:

$$K \geq K_P + 2\sqrt{K_I \cdot K_D} \ . \tag{B.21}$$

As discussed in Chapter 4, $K_P + 2\sqrt{K_I K_D}$ has a large value in the proposed test-signal backtrace structure. Therefore K is also a large number. Based on this point, equation (B.18) can be approximated as:

$$\delta_{k+1} \approx \frac{f^{''}(\alpha) \cdot \frac{\delta_k^2}{2} + \frac{B-\alpha}{K}}{f^{'}(\alpha) + f^{''}(\alpha) \cdot \delta_k} \tag{B.22}$$

For the case that $f^{'}(\alpha) \neq 0$, equation (B.22) can be rewritten as:

$$\delta_{k+1} = \frac{f^{''}(\alpha) \cdot \frac{\delta_k^2}{2} + \frac{B-\alpha}{K}}{f^{'}(\alpha) \cdot \left[1 + \frac{f^{''}(\alpha)}{f^{'}(\alpha)} \cdot \delta_k\right]} \tag{B.23}$$

Let $\lambda = \frac{f^{''}(\alpha)}{f^{'}(\alpha)}$, then by using the power series expansion [Abr72], the following approximation can be made:

$$\frac{1}{1 + \lambda \cdot \delta_k} \approx 1 - \lambda \cdot \delta_k \tag{B.24}$$

Equation (B.23) can therefore be rewritten as:

$$\delta_{k+1} \approx \frac{\lambda}{2} \cdot \delta_k^2 + \left[\frac{B-\alpha}{Kf^{'}(\alpha)}\right] - \frac{\lambda^2}{2} \cdot \delta_k^3 - \left[\frac{B-\alpha}{Kf^{'}(\alpha)}\right] \cdot \lambda \cdot \delta_k \tag{B.25}$$

The term $\frac{B-\alpha}{Kf^{'}(\alpha)}$ is the steady-state error in the final solution of the backtracing problem, where B is the value determined in the previous time slot $\tau - T$. As we know from control theory, B is getting increasingly closer to α as time passes by due to the integral action in the PID controller [Mor01]. Moreover, the gain of the PID controller K has a high value. Therefore the stead-state error will approach zero:

$$\frac{B-\alpha}{K \cdot f^{'}(\alpha)} \to 0 \tag{B.26}$$

and

$$\frac{B-\alpha}{K \cdot f'(\alpha)} \cdot \lambda \cdot \delta_k \to 0 \tag{B.27}$$

Hence equation (B.25) can be simplified as:

$$\delta_{k+1} \approx \frac{\lambda}{2} \cdot \delta_k^2 - \frac{\lambda^2}{2} \cdot \delta_k^3 \tag{B.28}$$

Hence it can be concluded that if λ has a finite upper bound, and the error at any iteration of the solution is proportional to the square of the error in the previous iteration. In other words, the proposed PID feedback based backtrace approach converges towards the final solution at a quadratic rate.

For the case that $f'(\alpha) = 0$ and $f''(\alpha) \neq 0$, equation (B.22) can be simplified to:

$$\delta_{k+1} \approx \frac{f''(\alpha) \cdot \frac{\delta_k^2}{2} + \frac{B-\alpha}{K}}{f''(\alpha) \cdot \delta_k} = \frac{\delta_k}{2} + \frac{B-\alpha}{f''(\alpha) \cdot \delta_k \cdot K} \tag{B.29}$$

As already discussed, B is getting increasingly closer to α as time passes and the gain of the PID controller K has a large value. Therefore the last term in (B.29) approaches to zero:

$$\frac{B-\alpha}{f''(\alpha) \cdot \delta_k \cdot K} \to 0 \tag{B.30}$$

With expression (B.29), one can obtain:

$$\frac{\delta_{k+1}}{\delta_k} \approx \frac{1}{2} < 1 \tag{B.31}$$

Approximation (B.31) shows that the simulation of the proposed signal backtrace structure is converging.

Finally, for the case that $f'(\alpha) = 0$ and $f''(\alpha) = 0$, equation (B.18) can be rewritten as:

$$\delta_{k+1} = B - \alpha \tag{B.32}$$

Since B is getting increasingly closer to α as the iteration continues, δ_{k+1} is getting increasingly smaller. According to the above analysis, it can be concluded that the simulation will converge. This means that the DC signal can be backtraced successfully with the proposed structure.

References

[Abr72] M. Abramowitz and I. A. Stegun, "Handbook of mathematical functions with formulas, graphs, and mathematical tables," John Wiley, New York, USA, ISBN: 0-471-80007-4, 1972.

[Mor01] K. Morris, "Introduction to feedback control," Academic press, ISBN:0-12-507660-6, 2001.

List of publications

1 **L. Fang** and H. G. Kerkhoff, "Test signal Backtrace in Mixed-Signal SoC Testing," Proc. of ProRISC2002 workshop, Veldhoven, the Netherlands, ISBN: 90-73461-33-2, pp. 276-282, 2002.

2. M. Stančić, **L. Fang**, M. H. H. Weusthof, R. M. W. Tijink and H. G. Kerkhoff, “Embedded Analogue Core Testing in Silicon Systems” Proc. of PROGRESS workshop on Embedded Systems, Utrecht, the Netherlands, ISBN: 90-73461-34-0, pp. 230-238, 2002.

3. M. Stančić, **L. Fang**, M. H. H. Weusthof, R. M. W. Tijink and H. G. Kerkhoff, "A New Test Generation Approach for Embedded Analogue Cores in SoC," Proc. of International Test Conference (ITC), October 8-10, Baltimore, USA, pp. 861-869, 2002.

4. **L. Fang** and H. G. Kerkhoff, "Sensitivity Analysis Based Tolerance-Box Generation and Propagation in Mixed-Signal SoC Testing ", Proc. of 7th IEEE European Test Workshop, Corfu, Greece, pp. 255-260, 2002.

5. **L. Fang** and H. G. Kerkhoff, "PID Feedback Based Backtrace Method for Mixed-Signal SoC Testing," Proc. of 8th IEEE International Mixed-Signal Testing Workshop, Montreux, Switzerland, pp. 147-152, June 18-21, 2002.

6. **L. Fang** and H. G. Kerkhoff, "Tolerance-Box Generation in Mixed-Signal SoC Testing," Proc. of ProRISC2001, ISBN 9073461294, pp. 342-347. Veldhoven, the Netherlands, November 2001.

7. **L. Fang**, G. Gronthoud and H. G. Kerkhoff, "A High-Level Modelling Algorithm for Analogue Fault Simulation," Proc. of ProRISC2001, ISBN 9073461294, pp. 348-355. Veldhoven, the Netherlands, November 2001.

8. **L. Fang**, G. Gronthoud and H. G. Kerkhoff, "Reducing Analogue Fault-Simulation Time by Using High-Level Modelling in Dotss for an Industrial Design", Proc. of IEEE European Test Workshop (ISBN 0-7695-1877), Stockholm, Sweden, May 29 - June 1, 2001, pp.61-67.

9. **L. Fang** and H. G. Kerkhoff, "Tolerance-Box Propagation in Analogue Testing," Proc. of 7th IEEE International Mixed-Signal Testing Workshop, Atlanta, U.S.A, pp. 156-160, June 13-15, 2001.

10. M. Stancic, **L. Fang** and H. G. Kerkhoff, "Testing of Parametric Faults in Embedded Analog Cores of System-on-Chip," Proc. of ProRISC2000, ISBN 9073461243, pp. 515-518. Veldhoven, November 2000.

11. **L. Fang**, M. Stancic and H. G. Kerkhoff, "Mixed-signal core-based testing," Digest of ETW 2000: IEEE European Test Workshop, Cascais, Portugal, May 23-26, 2000, pp. 279-280.

12. M. Stancic, **L. Fang** and H. G. Kerkhoff, "Detecting Parametric Faults in Embedded Cores of Mixed-Signal ICs" IMSTW2000: 6th IEEE International Mixed-Signal Testing Workshop, Montpellier, France, pp. 20-21, June 21-23, 2000.
13. M. Stancic, **L. Fang** and H. G. Kerkhoff, "Analogue Core-Based Test Pattern Generation," Proc. of ProRISC99, ISBN 90-73461-18-9, November 1999, Mierlo, pp. 461-466.
14. **L. Fang**, M. H. H. Weusthof and H. G. Kerkhoff, “Test Singal Backtracing for Testing Embedded Analogue Cores in SoC,” accepted for IMSTW2003: 9th IEEE International Mixed-Signal Testing Workshop, Spain, June 25-27, 2003.