INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI®

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Testability Analysis for Mixed Analog/Digital Circuit Test Generation and Design for Test

by

Sam DuPhat Huynh

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

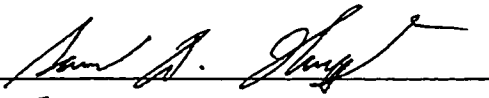
1999

Program Authorized to Offer Degree: Department of Electrical Engineering

UMI Number: 9936425

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to University Microfilms, 1490 Eisenhower Place, P.O. Box 975, Ann Arbor, MI 48106, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature 

Date June 7, 1999

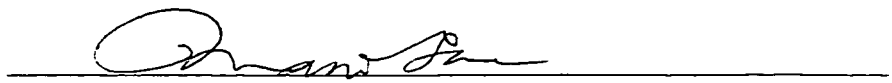
University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Sam DuPhat Huynh

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

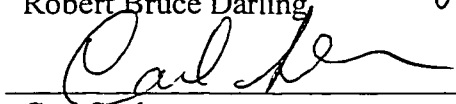


Mani Soma

Reading Committee:

Robert Bruce Darling.

Robert Bruce Darling



Carl Sechen

Date: June 7, 1999

University of Washington

Abstract

Testability Analysis for Mixed Analog/Digital Circuit Test Generation and Design for Test

by Sam DuPhat Huynh

Chairperson of the Supervisory Committee: Professor Mani Soma
Department of Electrical Engineering

Industry trends aimed at procuring greater levels of circuit performance have triggered a proliferation of analog and digital subsystems fabricated side-by-side on the same die. The combined requirements for both high-speed digital and high-precision analog functionality pose unique challenges to mixed analog/digital circuit designers and test engineers. Specifically, monolithic mixed-signal ICs are often difficult to test (verify its full functionality) and can require a significant amount of time. Long test times and difficulty in developing test programs has been identified as a significant contributor to this important problem.

This dissertation investigates computer-aided design (CAD) methodologies for analyzing the testability of mixed-signal integrated circuits (ICs). Testability information, if provided early in the design process, can help identify potential design and test problems and aid in the subsequent redesign. In our approach, SPICE compatible model parameters are used to efficiently generate a set of testability models which are used in novel design-for-test methodologies and automatic test pattern generation algorithms. The new methods are applicable to virtually any analog, digital or mixed-signal circuits and systems.

This manuscript also discusses the implementation of the testability analysis algorithms in an industry-compatible test program development system called REIGN. For validation, the strategy is effectively demonstrated using several mixed-signal benchmark circuits.

TABLE OF CONTENTS

LIST OF FIGURES

Number Page

LIST OF TABLES

Number Page

Acknowledgments

I thank my research advisor, Mani Soma, for giving me a chance, for his patience, and for the freedom he gave me to do it --- right or wrong -- in my own way, on my own terms. I learned a lot from this man, and I value his guidance. Committee members Bruce Darling, John Sahr, Carl Sechen and Vipin Kumar expressed genuine interest in my doctoral research and asked great questions.

I had the excellent fortune to work with some truly gifted engineers in the industry. I thank Dr. Peter Young, Dr. Chung-ping Chien, and Tuyen Vu of the Boeing Company, Joe Mielke, Edward Perkins, Allen Harstine, and Mike Jones of IMS, Inc., Dr. William Kao of Cadence Design Systems, Prof. Abhijit Chatterjee of Georgia Tech. University, Dr. Bill Russell of Wright Lab, and Dr. William R. Simpson of the Institute for Defense Analysis. I also acknowledgment Dr. Mehdi Katoozi of InControl, Inc. for imparting some of his vast technical knowledge and philosophies of life to me during my "formative" years.

I acknowledge my predecessors in Prof. Soma's research lab -- Giri Devarayanadurg, Aida Charoenrook, Yongje Lim, Kyungim Son and Dr. Heung-Joon Park for imparting a sense of discipline. I am also indebted to my current colleagues -- Prashant Goteti, Mahendra Lodha, Jinyan Zhang, Seongwon Kim and Khalid Al-fayyadh. Of course, this work would be possible if not for the hard work of Pam Eisenheim, program manager.

On a personal level -- I thoroughly enjoyed my time away from campus, and James Hewitt, Simon Wu, John Ling and Richard Saito provided the requisite sanity maintenance. James in particular found the wherewithal to read a rough draft of this manuscript when he absolutely did not have the time. I also thank the many people I have had the privilege of meeting and getting to know as friends during undergraduate and graduate school, who in one form or another inspired me to achieve for excellence and accept nothing less. Special thanks to Elisa Alderidge and Joyce St. Clair who are constant reminders that there's more to life then just electrical engineering. Finally, this effort is wholly dedicated with much love to my family -- Mom, Dad, Aunt, Grandma, Angellica, Phil, and Jane -- who are, hands down, the greatest people I will ever know. I especially thank them for their unconditional love and support through the "dog" years of graduate school.

Chapter 1
A Rationale for Testability Analysis

The inherent noise immunity associated with digital circuits has been a key factor influencing the remarkable evolution of digital VLSI design. The attainable complexities of purely digital, standard-cell-based integrated circuits are primarily constrained only by process feature size. On the other hand, many hardware tasks related to amplification and signal conditioning are best implemented in silicon using analog circuit techniques. It is not surprising that industry trends aimed at integrating higher levels of circuit functionality have triggered a proliferation of analog and digital subsystems fabricated side-by-side on the same die.

In any mixed-signal system containing both analog and digital subsystems, the inherent discrete signal levels of the digital subsystems and the continuous signal levels of the analog subsystems impose critical constraints on developing test programs to test the mixed-signal system. Long test times and difficulty in developing test programs has been identified as a significant contributor to this important problem [9], [59].

1.1 The Mixed-Signal Testing Problem

Testing of a system is an "experiment" in which the system is exercised and its resulting response is analyzed to ascertain whether it behaved correctly. An explicit listing of a step-by-step procedure for the "experiment" is referred to as a *test program*. The time required to completely execute the test program from start to finish is referred to as the *test time*. In particular, the mixed-signal testing problem refers to testing a system which has both analog and digital subsystems tightly coupled together in an *IC* (integrated circuit), *MCM* (multi-chip module) or *PCB* (printed circuit board). The inherent nonlinear relationship between the continuous levels of analog signals and discrete levels of digital signals makes testing a mixed-signal system especially difficult in terms of generating an appropriate stimuli and analyzing the resulting response. The stimuli and response can be either an analog or a digital signal depending on the type of inputs and outputs of the system.

Therefore, the ultimate goal is two fold:

- Reduce the amount of time required to develop a test program.
- Reduce the test time.

Test program development time can be reduce through the application of design-for-test (*DFT*) methodologies and test time reduction can be accomplished by developing automatic test pattern generation (*ATPG*) algorithms and built-in self-test (*BIST*) techniques. Both test program development time and test time are directly related to the product's time to market and by reducing these times, the product's time to market is also shortened. A product's time to market is inversely proportional to profit; longer time to market, less profit, shorter time to market, higher profit [3]. Recent studies have shown that test development time exceeds product design time [80]. Simply stated, it costs more to test then to design the product.

1.2 THE IMPORTANCE OF MIXED-SIGNAL C/O ANALYSIS

In testing a mixed-signal system, it is common practice to partition the system into

its corresponding analog and digital subsystems [36]. Test programs are separately developed for these subsystems and subsequently, these subsystems are tested separately. Testing the analog subsystem is the bottleneck in mixed-signal test in terms of long test times and the relative difficulty in developing a test program [64]. The presented work in this thesis can be used to evaluate the controllability and observability of analog and mixed-signal circuits and systems. *Controllability* is the ability to establish a specific signal value at each node in a circuit by setting values on the circuit's primary inputs. *Observability* is the ability to determine the signal value at any node in a circuit by controlling the circuit's primary inputs and observing its primary outputs. Figure 1.1 shows how controllability/ observability (C/O) analysis fits into the design process for ICs. C/O analysis is usually embedded as part of a test program development procedure as high lighted in the shaded region. C/O analysis can be used in two different ways.

1. Good controllability and observability information is of significant use to designers especially if it is available early in the design cycle. During the design phase, C/O analysis can help identify potential problematic nodes and guide the subsequent redesign for the purposes of enhancing circuit controllability and observability. C/O analysis may be used as a feedback tool for the design engineer. As the design engineer rearranges circuit topology and possibly adds test points, the design engineer can observe the effects of these actions on the circuit's C/O and factor this information into the design process. This is important in order to avoid expensive redesign at a later stage especially after first silicon.

2. C/O analysis guides and eases the task of test generation. For example, an automatic test generation program will be more likely to quickly succeed in generating test vectors if it has access to node controllability and observability information. This intelligent guidance has the potential of substantially improving the performance of automatic test generation programs.

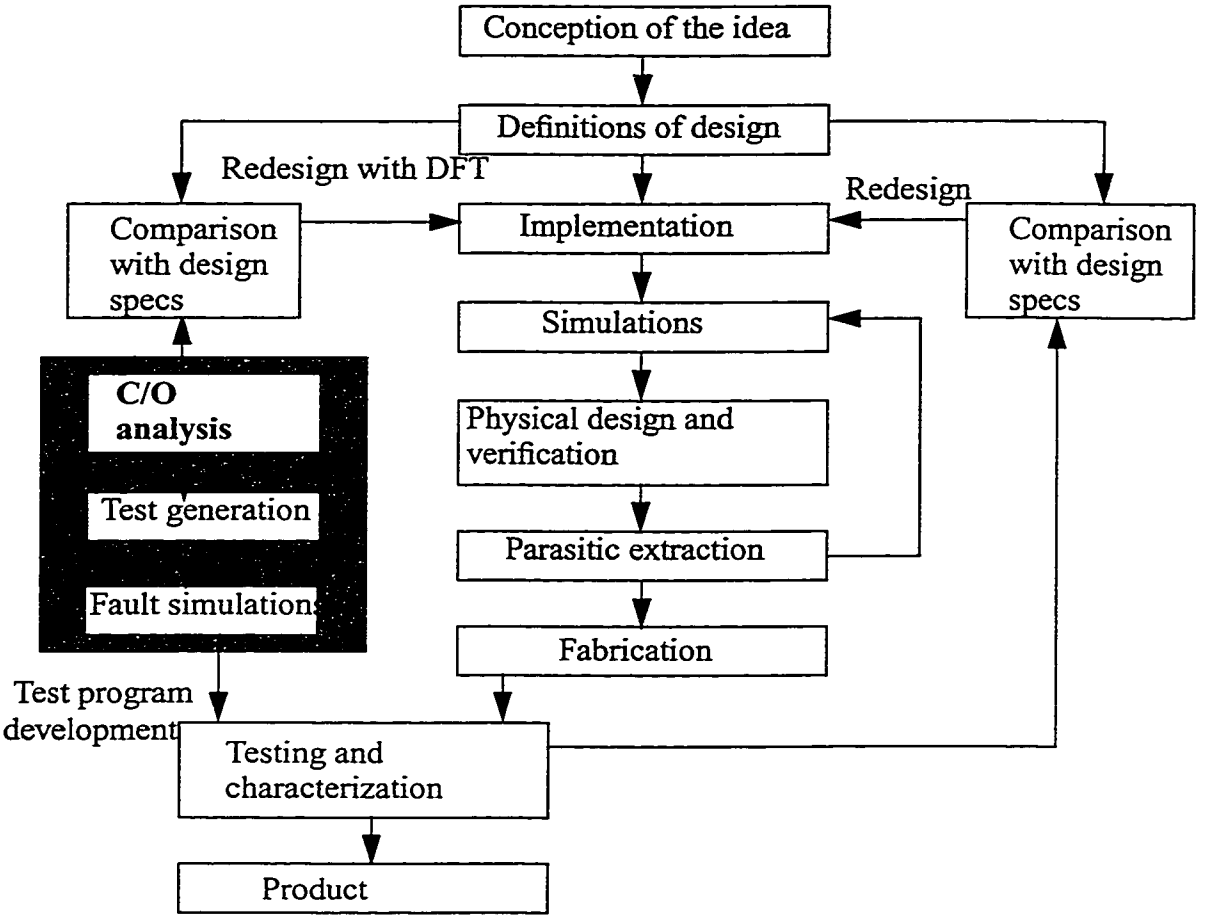


Figure 1.1: Design and test process for ICs

The primary objective of this work is to present a C/O analysis procedure which directly addresses the problem with strong consideration to both efficiency and practical implementation. A comprehensive overview of the relevant underlying fundamentals and an exposition of the current state of the art provide an essential review of the theoretical framework. Following a thorough elaboration on the C/O analysis formulation, we present experimental results which validate the procedure. Implementation details confirm the usefulness of the overall approach. Acknowledging the nature of the contribution as only a limited part of a larger scientific problem, suggestions for future research directions are provided in a closing chapter. The reference section represents a bibliography on C/O analysis and related topics currently in existence.

Chapter 2
Theoretical Framework

This chapter describes the testing fundamentals and design issues relevant to C/O analysis. These subjects are increasingly important, as the cost of testing is becoming the major component of the manufacturing cost of a new product. Today, design and test are no longer separate issues. The emphasis on the quality of the shipped products, coupled with the growing complexity of the VLSI design, require testing issues to be considered early in the design process so that the design can be modified to simplify the testing process.

2.1 TESTING PRINCIPLES

An instance of an incorrect operation of the system being tested (or *UUT* for unit under test or *DUT* for device under test) is referred to as an observed *error*. The causes of the observed errors may be design errors, fabrication errors, fabrication defects, and physical failures. Examples of design errors are:

- incomplete or inconsistent specifications

- incorrect mappings between different levels of design
- violations of design rules

Errors occurring during fabrication include:

- wrong components
- incorrect wiring
- shorts caused by improper soldering
- process variations

Fabrication defects are not directly attributable to human error, rather, they result from an imperfect manufacturing process. For example, a dust particle landing on the wafer results in either a (unintentional) short or open circuit between two nets. Other fabrication defects include improper doping profiles, mask alignment errors, and poor encapsulation. Errors and defects which lead to eventual failures of the UUT are collectively referred to as *faults*. For studies of physical faults in integrated circuits, refer to [5], [55], [79]. A fault is detected by observing an error caused by it. The basic assumptions regarding the nature of the faults are referred to as *fault models*.

2.2 FAULT MODELING

At any level of abstraction, a system or circuit can be viewed as a black box, processing the information carried by its inputs to produce its outputs. A *functional model* of a system is a mathematical representation of the system. A *structural model* describes a box as a collection of interconnected smaller boxes called components or elements. A structural model is often hierarchical such that a component is in turn modeled as an interconnection of lower-level components. The bottom-level boxes are called primitive elements and their functional model is assumed to be known. A structural model always

carries information regarding the function of its components. Faults defined in conjunction with a structural model are referred to as *structural faults*; their effect is to modify the interconnections among components. *Functional faults* are defined in conjunction with a functional model. For example, the effect of a functional fault may be to change the truth table of a component.

Unless explicitly stated otherwise, we will always assume that we have at most one fault in the UUT. This simplifying *single-fault assumption* greatly simplifies the analysis and computations. But even when multiple faults are present, the tests derived under the single-fault assumption are usually applicable for the detection of multiple faults, because a multiple fault can be detected by the tests designed for the individual single faults that compose the multiple one. The total number of possible faults assumed for the UUT is referred to as the *fault universe.*

In general, structural fault models assume that components are fault-free and only their interconnections are affected. Typical faults affecting interconnections are shorts and opens. A short, also referred to as a bridging fault, is formed by connecting nodes not intended to be connected, while an open results from the breaking of a connection. For example, in many technologies, a short between ground or power and a signal line can make the signal remain at a fixed voltage level. The corresponding logical fault consists of the signal being stuck at a fixed value (see figure 2.1).

2.2.1 Fault modeling for analog circuits

In the analog domain, there are no standard fault models, unlike in the digital domain where the classic stuck-at-fault is the *de facto* standard [1]. The debate in the analog domain focuses on four popular models: structural faults, parametric faults, inductive fault analysis (IFA), and behavioral based models. Structural faults consists of shorts (bridging), opens and stuck-at-value. Experiments data seemed to suggest that these types of faults are rare in analog circuits [70] and they do not provide much insight about the

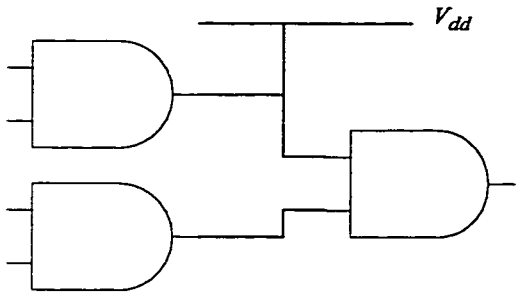


Figure 2.1a: Stuck-at-1 fault

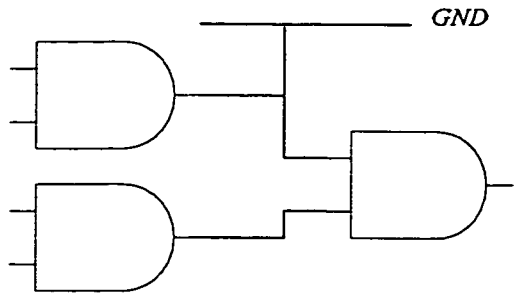


Figure 2.1b: Stuck-at-0 fault

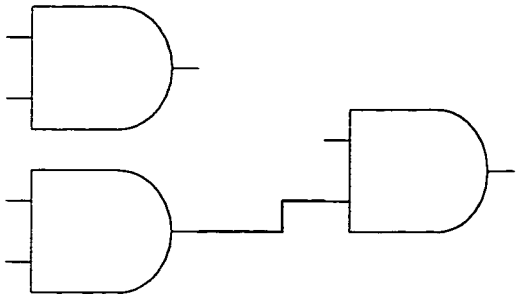


Figure 2.1c: Stuck open fault

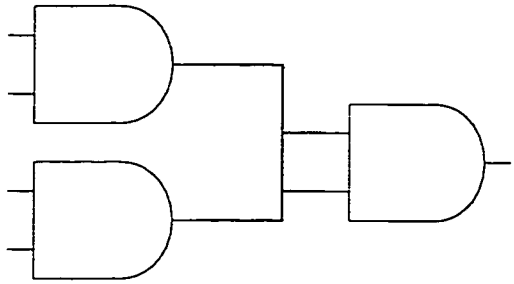


Figure 2.1d: Bridging fault

nature of the actual faults in the CUT. Parametric faults [23], or component variations are better suited for analog circuits. However, these types of faults imply an infinite fault universe. IFA [46], [47], [63] seemed to suggest the most relevance to analog circuits, but the major disadvantage is the long simulation and analysis time required to generate the fault models. IFA also requires layout information of the CUT. Behavioral based fault models [44] offer the best simulation times, however the methods used to generate the fault models are *ad hoc* at best. Since behavioral models are component-dependent and it is difficult to model non-linear functions, it is difficult to standardize a technique for this type of models.

2.3 AUTOMATIC TEST PATTERN GENERATION (ATPG)

Test generation (TG) is a complex problem with many interacting aspects. the most important are

- The cost of TG

- The quality of the generated test
- The cost of applying the test

The cost of TG depends on the complexity of the TG method. Random TG (*RTG*) is a simple process that involves only generation of random vectors. However, to achieve a high-quality test -- measured by the fault coverage of the generated test -- we need a large set of random vectors. Even if TG itself is simple, determining the test quality -- for example, by fault simulation -- may be an expensive process. Moreover, a longer test costs more to apply because it increases the time of the testing experiment. RTG generally works without taking into account the function or the structure of the circuit to be tested. In contrast, deterministic TG produces tests by processing a model of the circuit. Compared to RTG, deterministic TG is more expensive, but it produces shorter and higher-quality tests. Deterministic TG can be manual or automatic. In this section, we will focus on automatic TG (ATG) methods.

Deterministic TG can be fault-oriented or fault-independent. In a fault-oriented process, tests are generated for specified faults of a fault universe (defined by an explicit fault model). Fault independent TG works without targeting individual faults. Figure 2.2 shows a general view of a deterministic TG system. Tests are generated based on a model of the circuit and a given fault model. The generated tests include both the stimuli to be applied and the expected response of the fault-free circuit.

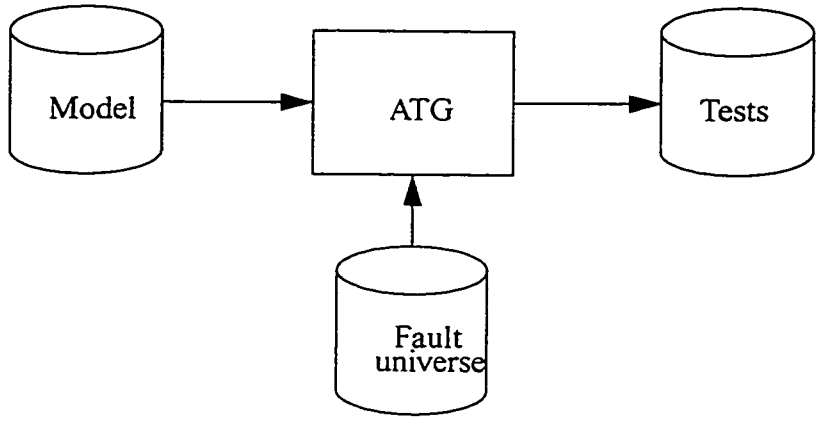


Figure 2.2: Deterministic test generation system

2.3.1 ATPG for analog circuits

A survey of literature shows that previous approaches to test generation have focused on generating tests for each individual class of faults. For the parametric faults [29], [49] work has focused on test ordering with an emphasis on finding an ordering of the specification test which will result in the maximal fault coverage. Recent work [53] generates impulse response based tests for analog LTI systems. In [43], a method is presented to correlate specifications to non-specification tests. In [22], [66], [67], and [69] optimization and sensitivity based approaches are presented for determining test frequencies for element diagnosis. In [81], the author presented an approach for linear analog circuits where a quadratic error objective was maximized.

Fault based approaches to analog testing have focused on the construction of *fault dictionaries* before manufacture through simulation. In [54], the authors recognizing that failures in analog circuits may assume a continuum of values, proposed a band-fault approach for linear analog circuits where the signature for each fault assumed the form of a band and the decision of pass/fail made on the basis of comparison of bands. Recently in [78], an approach based on interval analysis is presented for generating fault bands for linear analog circuits. In [50] the authors presented an algorithm to construct the signatures and evaluate the detectability of a set of DC measurements. A search technique in the frequency domain similar to the digital ATPG approach to select the best test frequency for linear circuits was presented in [51]. Most of the prevailing methods are fault analysis methods, where the effect of faults are analyzed under a known excitation.

In [67], the authors used sensitivity analysis as the basis for selecting test frequencies but no automated procedure was described. A min-max framework was presented in [13] and [14] however this approach requires building a surface response of the circuit under test (CUT). The authors described building the surface response through simulations and for small circuits, simulation times can be tolerable but not so for large analog

and mixed-signal systems. In [12], a method was described to shorten simulation times through behavioral modeling of macro blocks. The authors in [84] described a hierarchical specification driven technique to capture faults using simulation based methods. Simulation times may become intolerable. In [16] a hypothesis testing framework for test based on current monitoring was presented where the distributions needed for hypothesis testing are restricted to be gaussian. In [82] an optimization is performed over the continuous set of input values with a goal of obtaining one test for all the faults. This optimization may have no feasible solution if such a test does not exist. Further, the distance measure used neglects the sensitivity of the faulty circuit. In this dissertation, we study the problem of constructing a test set which can exploit the structural differences between the good and the defective circuits.

For any test generation algorithm, automatic or manual, to be successful, the generated test set must be of sufficient quality. Typically, the quality of a generated test (or test set) is measured by its *fault coverage* which is calculated by performing *fault simulations*.

2.4 FAULT SIMULATION

Fault simulation consists of simulating a circuit in the presence of faults. Comparing the fault simulation results with those of the fault-free simulation of the same circuit simulated with the same applied test set T, we can determine the faults detected by set T. *Fault coverage* is the ratio of the number of faults it detects and the total number of faults in the assumed fault universe.

2.4.1 Fault simulation for analog/mixed-signal circuits

Analog fault simulation is a very tedious and expensive task. The number of fault simulations that have to be carried out is estimated to be a power of two to six higher than a digital circuit of comparable complexity [38]. There are also a large number of possible errors due to deviation errors. Furthermore, until recently use was made of conventional circuit simulators, like SPICE and other derivatives, which suffer from poor reliability,

improper fault modeling insertion, inflexibility and inefficiency in terms of CPU time.

Serial fault simulation is the simplest method of simulating faults and it has been the choice for analog fault simulation in the past. As the name implies, serial fault simulation simulates one fault at time, hence this method is computationally intensive. Recent advances [26], [27], and [86] have shown that some of the techniques from the digital domain can be borrowed and applied to the analog domain. In [26] and [27], the concept of concurrent fault simulation was developed for analog circuits called CONCERT and experimental results showed that CONCERT is two orders of magnitude faster than serial fault simulation. Concurrent fault simulation refers to simulating multiple faults simultaneously.

Thus far, the discussion has focused on modeling faults and generating a test set based on the modeled faults. The test set is subsequently evaluated through some form of fault simulation to determine the test set's fault coverage or effectiveness in detecting the modeled faults. However, in some cases, certain faults cannot be detected or tests cannot be generated to detect a certain fault. Hence other techniques must be applied to help with test generation and fault detection. One such technique is called design-for-test (DFT).

2.5 DESIGN-FOR-TEST

Most DFT techniques deal with either the redesigning of an existing design or the addition of extra hardware to the design with the goal of improving *controllability* and *observability* of the circuit. Most approaches require circuit modifications and can affect such factors as area, I/O pins, and circuit delay. Hence a critical balance exists between the amount of DFT to use and the gain achieved. Typically, DFT can be classified into two categories:

- Test point insertion
- Circuit reconfiguration

2.5.1 Test point(s) selection

Test points are inserted to improve the circuit's controllability, observability or both. There are three types of test points, referred to as control points (CP), observation points (OP) and control/observation points (COP). Control points are primary inputs used to enhance controllability, observation points are primary outputs used to enhance observability, and control/observation points are both primary inputs and outputs used to enhance controllability and observability.

The major constraint associated with using test points is the large demand on I/O pins. This problem can be alleviated in several ways. To reduce output pins, a multiplexer can be used, as shown in figure 2.3. Here the $N = 2^n$ observation points are replaced by a single output Z and n inputs required to address a selected observation point. The main disadvantage of this technique is that only one observation point can be observed at a time; hence test time increases. A similar concept can be used to reduce input pin requirements for control inputs. The values of the $N = 2^n$ control points are serially applied to the input Z. Using a demultiplexer, these N values are stored in the N latches that make up register R (figure 2.4).

Another way to enhance observability and/or controllability is by using a scan register (SR). A scan register is a register with both shift and parallel load capability. The storage cells in the register are used as observation points and/or control points. Over the years, many different implementations for scan have been proposed and the interested reader is recommended to follow up on the references [2], and [45]. IBM has developed several scan designs referred to as Level-Sensitive Scan Design (LSSD), which have been used in many IBM systems [11]. Perhaps the most popular of scan implementations is the boundary scan in particular, the IEEE 1149.1 Testability Bus Standard [74] and its mixed-signal cousin IEEE P1149.4 Proposed Testability Bus Standard [57].

Although designing digital shift registers are trivial, it is rather challenging to design an analog counterpart. Authors in [35], [40], [71], and [73] proposed designs of analog shift registers based on the switched current concept. Current based scan techniques are similar to digital scan except that the scanned signal is a current.

A voltage signal can be converted to a current using a voltage to current converter (V-to-I). The current is sampled onto an analog scan bus, consisting of a series of analog scan "flip-flops" operated by a 2-phase clock (see figure 2.5). Each flip-flop has a master/slave current mirror and mimics the function of a digital scan flip-flop. If the sampled signal is a current, there is no need for the voltage to current converter. However, there are several issues with this design. The current mirror never has an ideal gain of 1, thus the accuracy of the signal being scanned is limited by the length of the scan chain. To observe more signals, an automatic calibration technique has to be incorporated into the design to reduce this source of errors to meet accuracy requirements. Clock feed-through is another source of error. Switched-current designers have already devised several techniques [28] to reduce this error. These techniques improve the measurement accuracy at a slight layout expense for each current mirror.

In some cases where adding more hardware is impossible because of silicon area constraints, then other methods to improve the circuit's controllability and observability

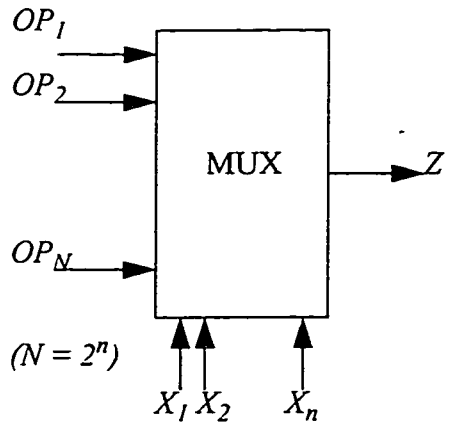


Figure 2.3: Multiplexing monitor points

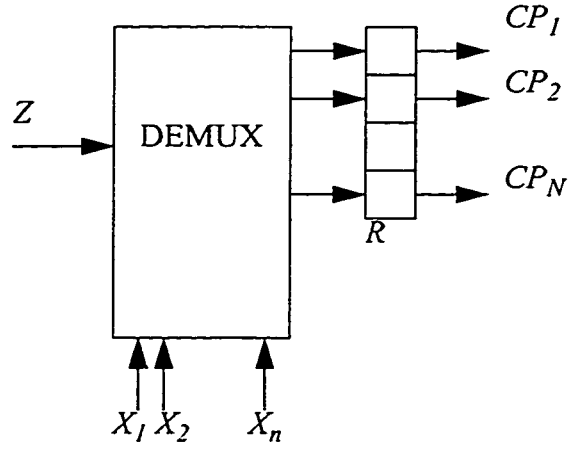


Figure 2.4: Demux'ing control points

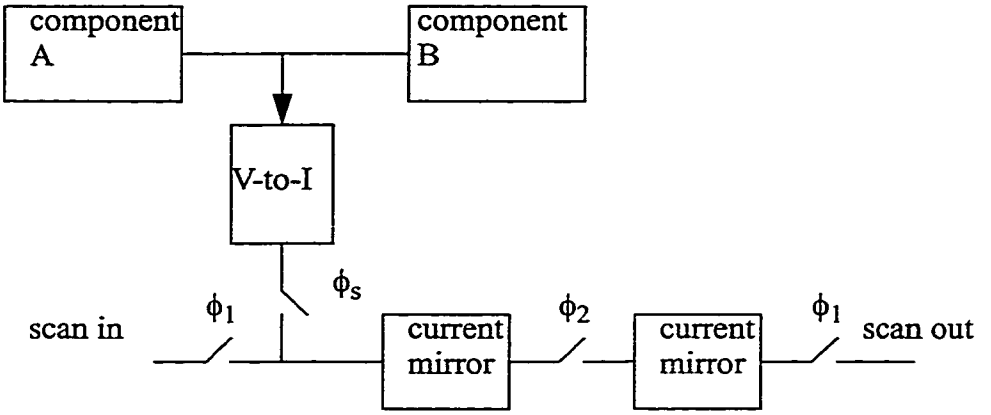


Figure 2.5: An analog scan chain implementation

must be used. One such method is circuit reconfiguration, where the circuit is physically altered to enhance its controllability and observability.

2.5.2 Circuit reconfiguration

Circuit reconfiguration is a concept which refers to the partitioning and rearranging of the subsystems in the DUT in such a way as to enhance controllability and observability. In mixed-signal systems, it is common practice to partition the analog and digital subsystems into separate blocks and apply different testing strategies to each separate blocks.

Circuits with feedbacks are a class of circuits difficult to test. In feedback circuits, the current output is dependent on both the current input and the previous output. An example is the phase locked loop (PLL) circuit. The PLL is a hard mixed-signal circuit to test due to the presence of tight feedback and largely varying time constants. The PLL has two feedbacks, a global feedback from output to input and a local feedback in the voltage controlled oscillator (VCO). A general rule is to provide logic to break the feedback path(s) [1] in test mode. [4] and [20] showed a DFT design for breaking the feedback path of the VCO with minimal impact on the overall PLL performance. [72] showed a DFT design to reconfigure a filter as a gain amplifier to test for various frequency domain parameters.

2.6 CONTROLLABILITY/OBSERVABILITY CONSIDERATIONS

As discussed in Chapter 1, C/O information can be used in two different ways: (*i*) to help with ATPG and (*ii*) to aid in DFT. Any type of test generation algorithm would require fault models. Fault modeling is a separate research field by itself, therefore, this manuscript will not go any further then the exploratory introduction discussed earlier in section 2.2. Instead, the generation of fault models is assumed to be done by others and this work will only use the resulting models.

As figure 1.1 shows, the intent is have test program development a concurrent activity with circuit design. This flow implies that layout information of the circuit under test might not be available initially. Without layout information, it will be difficult to apply IFA techniques to generate fault models. However, design is an iterative process between simulations and physical design (layout). Therefore IFA techniques may be used during the final stages of design when the layout is completed. During the initial design and test program development phases, the two fault models which are applicable are parametric and catastrophic faults. Since component variations is a continuous function, the fault universe for parametric faults is infinite. Therefore, only samples from this universe are studied. The intent is to sample this fault universe such that the sampled universe is a good representative of the continuous fault universe. One can also interpolate between sample points to infer more results. Behavioral based fault models is not considered in this work because there are no standardized technique to apply these faults. In this work, circuits and systems are examined in the context of parametric and catastrophic faults.

C/O information combined with the parametric and catastrophic fault models forms the backbone for the automatic test pattern generation methodology and design-for-test techniques presented in this manuscript. The generated test set is fault simulated to reduce its size while maintaining a high fault coverage. Smaller test sets are more desirable because the size of the test set is linearly proportional to test time which implies that

with a smaller test set, the test time is also reduced.

Fault simulation is a complex and challenging problem by itself as evidence by the amount of research literature available (see section 2.4). This work does not include the research nor development of fault simulation, rather it uses the results of other researchers whose research focus is on fault simulation especially in the analog domain. For the ease of implementation, the fault simulation strategy chosen is that of serial fault simulation utilizing the HSPICE circuit simulator.

Chapter 3
Previous and Concurrent Research

This chapter reviews the relevant current and previous research in the field of controllability/observability (C/O) analysis.

3.1 Overview and Historical Perspective

Various algorithms for C/O analysis of digital circuits have been proposed since 1976 and have since developed into a mature field as evidenced by the wealth of published literature. Analog C/O analysis, while having been proposed since 1977, has not matured to the same extent. This can be attributed to the phenomenal growth in the popularity of digital circuits.

In digital C/O analysis, the fundamental assumption that all signal nets are independent fails to deal with reconvergent fanout nodes and accounts for large deviations between the computed measures and actual test difficulty. Since reconvergent fanouts are quite common in digital systems, many implementations of the algorithms have resorted to *ad hoc* techniques which significantly increased the computational complexity. As a direct consequence, brute force conversion of automated digital C/O analysis methodolo-

gies to the mixed-signal paradigm has been largely unsucessful.

3.2 DIGITAL C/O ANALYSIS

Two major classifications of C/O measures are cost measures and probabilistic measures. Cost measures assign low values to nodes easy to control or observe, thus the lower the C/O measure, the better the C/O of the node. Probabilistic measures assign probabilities to nodes, either in controlling them or observing them, thus higher probabilities means a better C/O.

3.2.1 Probabilistic Approaches

Stephenson and Grason [21], [77] developed C/O measures as the probabilities for controlling and observing nodes, without distinguishing between combinational and sequential types. These algorithms produce either two figures of merit per node (one for control probability and one for observe probability) or three figures of merit (two for control probability due to the two possible nodal logical values and one for observe probability). For each digital component, a set of equations defining the controllability transfer function and observability transfer function are derived, relating input and output probabilities of that component. The computations of the probabilities for all nodes then use graph-based techniques and these component-based probability functions. CAMELOT [6] developed by Bennets, Maunder and Robinson used the same concepts. Other approaches for testability analysis reviewed include VICTOR [58], ITTAP [17], and BETA [10]. Their approaches are similar to the ones discussed above and therefore will not be discussed here.

3.2.2 Cost Based Methods

Goldstein [18], [19] proposed six cost functions to characterize each digital node for testability:

1. Combinational-0 controllability: cost of setting a node to 0 in combinational circuits
2. Combinational-1 controllability: cost of setting a node to a 1 in combinational circuits
3. Sequential-0 controllability: cost of setting a node to 0 in sequential circuits
4. Sequential-1 controllability: cost of setting a node to 1 in sequential circuits
5. Combinational observability: cost of observing a node in combinational circuits
6. Sequential observability: cost of observing a node in sequential circuits

The algorithm for controllability and observability analysis assumes that all signal nets are independent, all circuit inputs are controllable (initial cost either 1 or 0), and all circuit outputs are observable (initial cost 0). For each gate or sequential component, a set of equations is derived to relate the cost functions of the output to those of the inputs. The computations of the cost figures for all nodes then proceed using graph-traversal algorithms and these component-based cost functions. TESTSCREEN developed by Kovijanic [41], [42] and COMET developed by Berg and Hess [7] used a similar approach.

3.3 ANALOG C/O ANALYSIS

Analog C/O measures, while having been proposed since 1977, have not matured to the extent of the digital measures. Sen and Saeks [60], [61], [62] proposed a measure for analog linear circuits using component connection model (CCM) and multi frequency analysis technique. CCM is employed to distinguish between components (whose transfer functions depend on frequencies and sensitivities of elements under study) and connections (whose transfer functions are purely algebraic equations since they are assumed fault-free). Visvanathan *et. al.* [83] and G. Iuculano *et. al.* [37] used a similar approach.

Stenbakken and Souders [76] considered linear models and defined the *t*-testability for circuits. The *t*-testability measure of a circuit, given a candidate set of test points, is the percentage of the circuit components whose testability factors are greater than or equal to a threshold value *t*. *t* is called the circuit testability factor. The components whose testability factors are less then *t* are said to be untestable. However, the circuit testability factor *t* seems to be chosen in an arbitrary manner without a clear guideline. Stenbakken *et. al.* [75] subsequently used the concept of *t*-testability to define ambiguity groups, which are groups of components whose sensitivities to measurement errors are approximately equal in all the measure data. The authors develop more algorithms to determine component ambiguity groups based on the sensitivity model, and to decide where to insert test points to improve C/O.

Hemink *et. al.* [24], [25] developed a tool called TASTE along the same line of taking into account measurement errors in parameter estimations and deriving sets of inseparable parameters that need more measurements. The approach relies on the sensitivity matrix that relates the sensitivities of the parameters p to the measurements x. If the rank of this matrix is less than the number of parameters, then there exist inseparable parameters. If the rank of the matrix is larger than the number of parameters, then fewer measurements are needed, saving test cost.

Liu *et. al.* [44] defines the testability of a circuit as the extent to which faults can be identified from measurements. There is no equation in this work to define a testability measure. Instead, the authors focus on the fault diagnosis problem, assuming the linear error model relating the measurement errors to the parameter estimation errors. The least square estimate of the parameter errors can be computed based on the sensitivity matrix U, which can be derived using behavioral modeling. The same concept of ambiguity groups is described. The null space of U is the number of ambiguity groups. A fault diagnosis procedure is described using this technique.

Slamani *et. al.* [65], [67], [68] defines the observability measure for analog circuits as follows:

> The defect observability of a component x_i is defined as the sensitivity of the output parameter T_j with respect to the variations of component x_i.

Note that the sensitivity matrix in this case relates the output parameter to a component value without any concern for measurements. According to this definition, the higher the sensitivity the higher the observability. Sensitivities defined in the manner can be automatically computed by a circuit simulator such as SPICE. The paper proceeds to study fault diagnosis issues by defining fault equivalence, fault masking, fault dominance, and fault isolation.

3.4 MIXED-SIGNAL TESTABILITY ANALYSIS

In this dissertation, we propose an approach that is different from the methods presented previoulsy for analog circuit C/O analysis. We are not interested in diagnosis evaluation or analysis for analog circuits or the study of dependencies between the set of diagnosis equations to component deviations. Instead, we want to develop C/O models for analog and digital circuits which can be used for test point insertion (a DFT application) and implemented into a CAD tool. Our objective is to develop algorithms for C/O analysis of mixed-signal systems based on these models.

This dissertation describes the development of two measures to characterize the controllability/observability properties of the internal nodes of an analog circuit. The intent is to provide a quantitative measure of the difficulty of controlling and observing the signal values of internal nodes. We define two separate *testability measures*:

1. *Controllability measure*: the relative difficulty of setting a node to a

specific value. This measure is normalized to range from 0.0 to 1.0 with a 1.0 being totally controllable and 0.0 being totally uncontrollable. Primary Inputs (PIs) are by definition totally controllable, therefore the controllability at all PIs are 1.0.

2. *Observability measure*: the relative difficulty of propagating an error from an internal node to a primary output. This measure is normalized to range from 0.0 to 1.0 with 1.0 being totally observable and 0.0 being totally unobservable. Primary Outputs (POs) are by definition totally observable, therefore the observability at all POs are 1.0.

These measures can be used to guide design-for-test (DFT) with the purpose of enhancing C/O. One popular DFT methodology is test point insertion and a test point can be either a control point, observe point or both [35], [40]. Examples of this methodology include scan (IEEE Standard 1149.1, *etc.*). and multiplexer techniques. The controllability and observability measures can serve as the basis to decide where to place control and/or observe points into the circuit. These points will not necessary be placed on nodes with the lowest controllability/observability measures but on nodes whose enhanced controllability/observability measures enhances the overall controllability/observability of the circuit. Furthermore when combined with fault models, these measures can be used to help generate test patterns which may externally be applied using an arbitrary waveform generator. In this work, *testability analysis* refers to the procedure used for calculating a circuit's (or systems') controllability and observability measures.

Chapter 4
Testability Analysis Formulation

In works published to date, the most prevalent strategy for mixed-signal C/O analysis is to partition the digital and analog subsystems and analyze each subsystem separately using different techniques. Two popular methods used in digital circuits are probabilistic based methods and cost functions. Sensitivity analysis, in one form or another, is the method of choice for analog circuits. The main drawback lies in the wide differences between the methods used. As a whole mixed-signal system, it is difficult to combine the results, however, in cases where data are combined, the results are inconclusive at best.

This chapter introduces a strategy to solve the mixed-signal C/O problem as a whole instead of partitioning the mixed-signal system into its corresponding digital and analog subsystems. The strategy borrows the concept of transfer functions from linear systems analysis and formulates the testability transfer factors. Nodal equations are formed based on these testability transfer factors and these equations are solved for the testability of the mixed-signal system. This chapter describes the derivations and relevant procedures of this strategy.

4.1 TESTABILITY TRANSFER FACTORS

The following derivations are for *components* and a component is defined as a building block in which the designer can use but does not have access to the internal of the component. Examples include resistors, capacitors, inductors, operational amplifiers (op-amp), and operational transconductance amplifiers. Higher level component examples include such macro blocks as analog-to-digital converters (ADCs), digital-to-analog converters (DACs), phase locked loops (PLLs) and others.

The concept of Testability Transfer Factor (TTF), testability models, of a component was first devised by Stephenson and Grason [77] for digital circuits and later extended to the analog domain by Huynh, Soma and Zhang [34]. The underlying idea is that test information must be propagated through components to other components or to primary outputs, a means must be employed to determine how controllability and observability is affected in route. The TTF of a component represents:

1. The ease of achieving an arbitrary signal on its outputs by exercising its inputs.
2. The ease of determining whether a specific signal occurred on its inputs by examining the values on its outputs.

4.1.1 Models for Passive Linear Components

Consider two subcircuits in a circuit coupled through a resistor with resistance R at nodes a and b (refer to figure 4.1). *PIs* indicate primary inputs and *POs* indicate primary outputs. Node b is controllable by node a only if a current of magnitude I can flow from node a to node b. The current I is described by Ohm's law as:

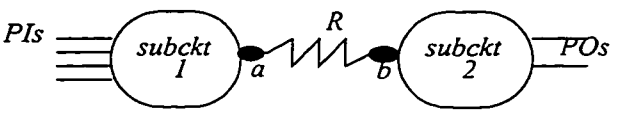


Figure 4.1: Two subcircuits coupled through resistor R

$$I = \frac{V_a - V_b}{R} \tag{4.1}$$

As the resistance R increases, the current I decreases and as the resistance R decreases, the current I increases. Respectively these two cases are known as tending towards the open circuit condition and tending towards the short circuit short condition. Current cannot flow into an open circuit and voltages are identical for both nodes in a short circuit. Then we can conclude that node b is perfectly controllable by node a if $R = 0$ (short circuit). That is any voltage appearing on node a will also appear on node b. Furthermore, if R equals to infinity (open circuit), then node b is perfectly uncontrollable by node a since no current can flow into an open circuit.

Similarly, node a is perfectly observable by node b if $R = 0$ because any voltage appearing on node a will appear on node b as well. Also, node a is unobservable by node b if R is equal to infinity because with zero current, voltage at node a can not be inferred even though the voltage at node b is known.

From the above analysis, we can conclude that controllability and observability increases as R approaches zero (short circuit condition) and controllability and observability decreases as R approaches infinity (open circuit condition). This can be expressed quantitatively as:

$$T_f(R) = 1 - \frac{R}{\alpha OC} \tag{4.2}$$

Equation (4.2) is a quantitative expression which states that information cannot be propagated through an open circuit while information is fully propagated through a short circuit. The open circuit condition (OC) is commonly modeled by a resistor with a high resistance [36]. In this work, the open circuit condition is modeled by a 10 mega-ohm resistor. A 10 mega-ohm resistor is sufficient for our work because most of the resistance values are well below 1 mega-ohm. Furthermore, the OP27 op-amp (a high performance op-amp) has an input impedance of well over 6 mega-ohm (at DC) and that is considered an open circuit for most analog applications.

α is referred to the technology adaptor coefficient. Assume an experimental setup shown in figure 4.2 where an ampere meter is attached in series with a 1 volt (voltage) source and the circuit under test (CUT). Assume the CUT is a variable resistor and the loop current (I) is measured by the ampere meter. The purpose of this experiment is to determined the smallest measurable loop current, I. Given the 1 volt source, then the largest measurable impedance for this setup is:

$$R_{CUT + AmpMeter} = \frac{1}{I} \tag{4.3}$$

Assume the resistance of the ampere meter is much, much less then the CUT, then the resistance of the CUT can be approximated as:

$$R_{CUT} = \frac{1}{I} \tag{4.4}$$

This assumption is valid since the resistance of most ampere meters are on the order of ohms while the resistance of the CUT is on the order of mega-ohms. Therefore, the technology adaptor coefficient is defined as:

$$\alpha = \frac{R_{CUT}}{OC} \tag{4.5}$$

In this work, α is chosen to be 1, however, for equipments with higher or lower resolutions, α can be chosen (measured) accordingly.

In most specifications, resistors are specified with a certain tolerance:

$$R_{tot} = R_{nom} \pm \Delta R \tag{4.6}$$

Where R_{tot} is the total resistance, R_{nom} is the nominal resistance, and ΔR is the allowable

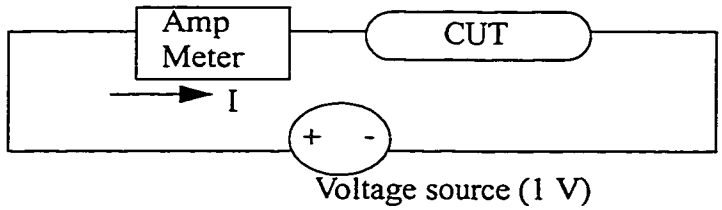


Figure 4.2: Experimental setup for determining α

tolerance. Substituting equation (4.6) into equation (4.2), results in:

$$T_R = 1 - \frac{R_{nom}}{\alpha OC} \mp \frac{\Delta R}{\alpha OC} \tag{4.7}$$

Which can be rewritten as:

$$T_R = T_{Rnom} \mp \Delta T \tag{4.8}$$

Where:

$$T_{Rnom} = 1 - \frac{R_{nom}}{\alpha OC} \tag{4.9}$$

$$\Delta T = \frac{\Delta R}{\alpha OC} \tag{4.10}$$

Therefore, equation (4.8) can be used to map specification space into testability space for analysis. We can also calculate a tolerance band around the nominal value, a concept similar to the Pahwa's and Rohrer's fault bands [54].

Notice that equation (4.9) is identical to equation (4.2). In the fault free circuit, R_{tol} is determined by equation (4.6) and therefore T_R is determined by equation (4.8). In case of a fault (either parametric or catastrophic), ΔR is above the specified tolerance, T_{Rf}(TTF of a resistor at fault) will be outside of the tolerance bands described by equation (4.8).

A temperature dependent resistor can also be modeled by equations (4.2). The HSPICE temperature dependent resistor model [48] has the following form:

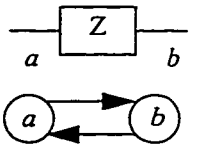
$$R(T) = R(1.0 + TC1 \bullet \Delta t + TC2 \bullet \Delta t^2) \tag{4.11}$$

Where TC1 is the first order temperature coefficient, TC2 is the second order temperature coefficient, T_{nom} is the nominal temperature in degrees Kelvin, and T is the temperature under analysis. Δt is defined as the difference between T and T_{nom} ($\Delta t = T - T_{nom}$). TC1 and TC2 are a function of the process technology and material used to fabricate the resistor. They represent the rate of change of resistance with respect to temperature. $R(T)$ can be calculated first by equation (4.11) and then substituted into equation (4.2) to calculate

the TTF.

Capacitors and inductors can be viewed as frequency dependent resistors. Therefore, the resistance R in equation (4.2) can be replaced by $|R(\omega)|$ or in general $|Z(\omega)|$ where Z represents the component's impedance and ω is the radian frequency. Table 1 summarizes the passive components TTF. The bi-directional nature of these components can be modeled a signal flow graph (SFG). The bi-directionality is modeled as two uni-directional links with identical weights but opposite directions. The weights of each link is the TTF value.

Table 1: TTF for R, L, and C

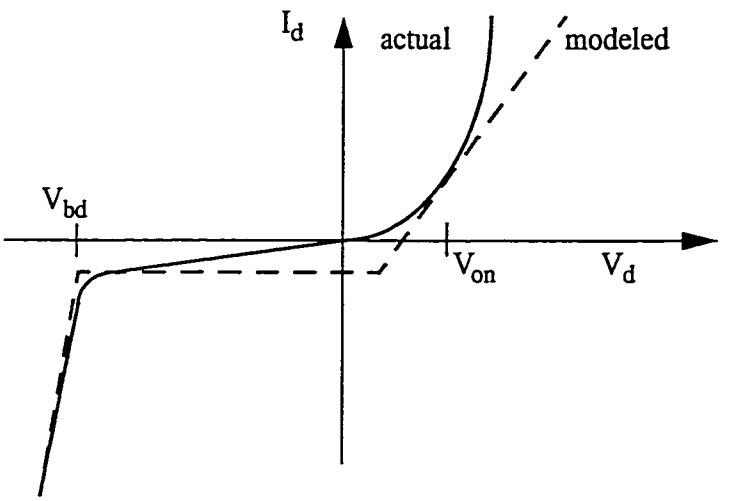
Circuit element	TTF	SFG
R, L, C	$1 - \frac{\lvert Z(\omega, temp)\rvert}{\alpha OC}$	

Also, the temperature dependence equation for resistors (4.11) is directly applicable to capacitors and inductors. The capacitance and inductance temperature dependence equations have the same form. The effective capacitance or inductance can be calculated by replacing the resistance value and its temperature coefficients with either capacitance or inductance and their corresponding temperature coefficients.

4.1.2 Models for Nonlinear Components

Diodes

Diodes can be viewed as a nonlinear resistor as shown by its I-V characteristic in figure 4.3. Piece-wise linear techniques can be use to model its nonlinearity. The diode's conductance can be modeled as three separate linear regions of operation:



V_{bd} = Breakdown voltage
V_{on} = Turn on voltage of diode
V_d = Voltage across diode
I_d = Current through diode

Figure 4.3: I-V characteristic of diodes

1. Forward bias: $V_d > V_{on}$

   In this region of operation, the conductance of the diode is equal to the partial derivative of I_d with respect to V_d. By evaluating this partial derivative at V_{on}, the result is a linear conductance or equivalently, a linear resistor as described by:

$$g_f = \left.\frac{\partial I_d}{\partial V_d}\right|_{V_{on}} = \frac{1}{R_f} \tag{4.12}$$

   Typically, V_{on} is 10 times V_T where V_T is the thermal voltage (26 mV at room temperature). However, a quick DC simulation will produce a much more accurate V_{on}.

2. Reverse bias: $V_{bd} < V_d < V_{on}$

   In the reverse bias region, the diode is basically an open circuit with only the reverse saturation current flowing. Therefore, the conductance in this region can be estimated by:

$$g_r = \left|\frac{I_s}{V_{on} - V_{bd}}\right| = \frac{1}{R_r} \tag{4.13}$$

3. Breakdown: $V_d < V_{bd}$

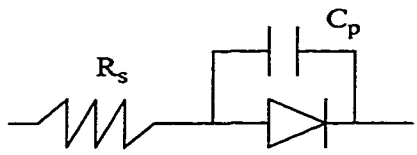


Figure 4.4: Complete diode model

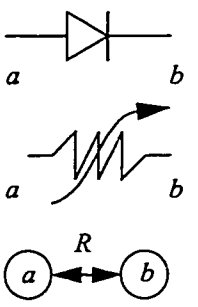
In the breakdown region, the voltage across the diode remains constant while the current through it can be of any magnitude. However, most diodes have a maximum power rating which limits the maximum amount of current which can flow through the device. Therefore, the conductance or equivalently the resistance in this region is:

$$g_{bd} = \frac{|I_{max}|}{|V_{bd}|} = \frac{1}{R_{bd}} \tag{4.14}$$

Figure 4.3 shows the actual I-V characteristic of a typical diode superimposed with the piece-wise linear model used. The region of operation and the various parameters can be determined by running a quick DC simulation. Once the resistance is known, equation (4.2) can be used to calculate the TTF of the diode.

Lead resistance and junction capacitance can be incorporated into the diode model by placing a resistor in series and a capacitor in parallel with the diode as shown in figure 4.4. R_s and C_p values can be either estimated, read from data sheets or one can run a simulation to determine these values. Once these values are know then, table 1 can be used to calculate the appropriate TTFs. Table 2 summarizes the diode TTF.

Table 2: TTFs for diodes

TTFs	SFG
Forward bias: $g_f = \left.\frac{\partial I_d}{\partial V_d}\right\|_{V_{on}} = \frac{1}{Rf}$ Reverse bias: $g_r = \left\|\frac{I_s}{V_{on} - V_{bd}}\right\| = \frac{1}{R_r}$ Breakdown: $g_{bd} = \frac{\|I_{max}\|}{V_{bd}} = \frac{1}{R_{bd}}$	

MOSFETs

A typical MOSFET is characterized by its I-V characteristics. The output current (drain to source current) is dependent on its input voltage (gate to source voltage) and output voltage (drain to source voltage) as shown in figure 4.5. The following derivation is for N type MOSFETs, however this derivation is also valid for P type MOSFETs with the differences being the signs of the voltages and currents.

The drain to source current, i_{DS}, is a multi-variable function:

$$i_{DS} = f(v_{GS}, v_{DS}) \tag{4.15}$$

The equivalent circuit model for the MOSFET is shown in figure 4.6. The transconductance, g_m, is the partial derivative of the drain to source current, i_{DS}, with respect to the

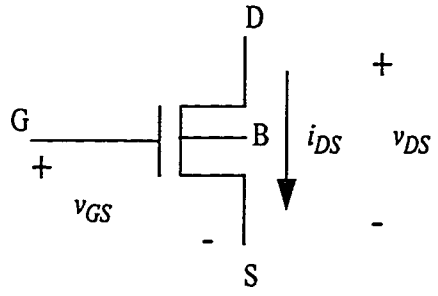


Figure 4.5: N type MOSFET

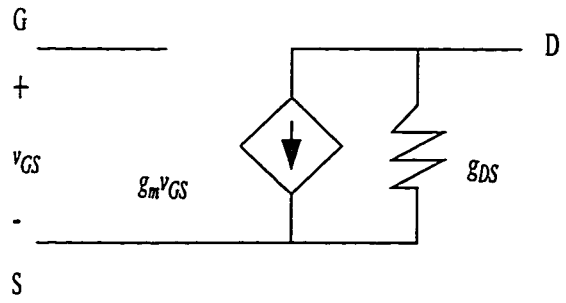


Figure 4.6: MOSFET equivalent circuit

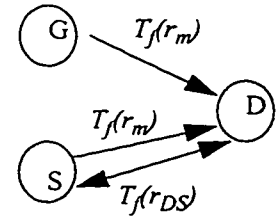
gate to source voltage, v_{GS} and the channel conductance, g_{DS}, is the partial derivative of the drain to source current, i_{DS}, with respect to the drain to source voltage, v_{DS}.

$$g_m = \frac{\partial i_{DS}}{\partial v_{GS}} = \frac{1}{r_m} \tag{4.16}$$

$$g_{DS} = \frac{\partial i_{DS}}{\partial v_{DS}} = \frac{1}{r_{DS}} \tag{4.17}$$

Once r_m and r_{DS} are known, then equation (4.2) can be used to calculate the TTF for each component. Table 3 shows the signal flow graph representation of the MOSFET with T_f representing the testability transfer factors as a function of the respective component. The appendix details the derivations for g_m and g_{DS} for the MOSFET.

Table 3: TTFs for MOSFETs

TTFs	SFG
$g_m = \frac{\partial i_{DS}}{\partial v_{GS}} = \frac{1}{r_m}$ $g_{DS} = \frac{\partial i_{DS}}{\partial v_{GS}} = \frac{1}{r_{DS}}$	

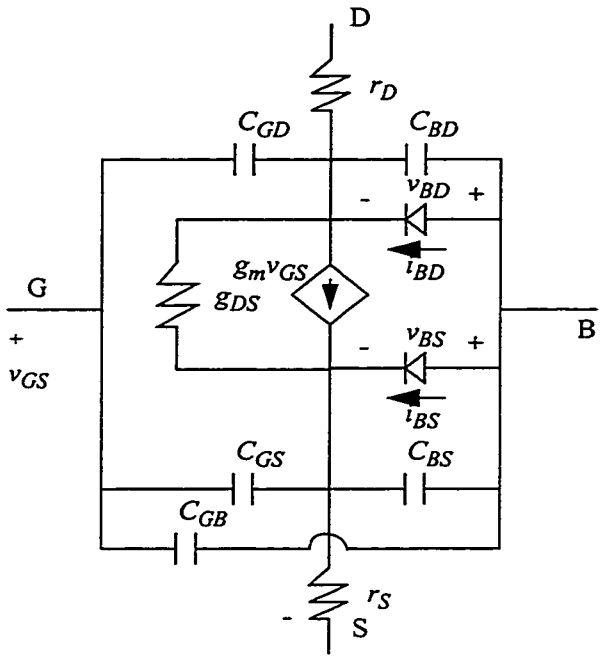


Figure 4.7: Complete MOSFET model

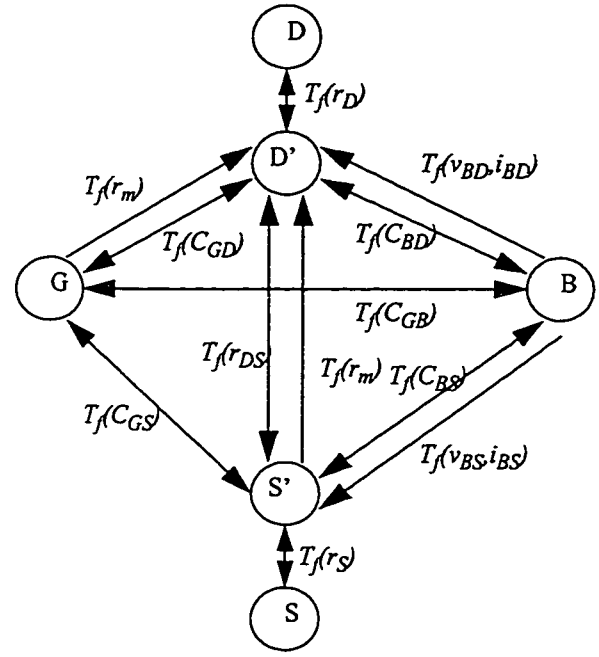


Figure 4.8: Complete SFG for MOSFET

The above derivation was done assuming an ideal MOSFET neglecting parasitic resistances, capacitances and body effects. These issues can be addressed by incorporating the HSPICE model [48] as shown in figure 4.7. The testability transfer factor for capacitors, resistors, and diodes are discussed in previous sections. Once all the parasitic values are known, figure 4.7 can be transformed into a SFG as shown in figure 4.8. TTFs for other types of transistors (such as BJTs) can be derived using a similar analysis.

4.1.3 Models for Transformers

An ideal transformer with single primary and single secondary is shown in figure

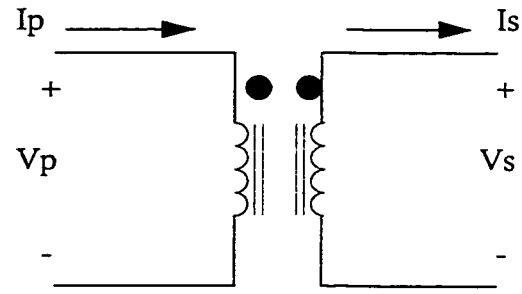


Figure 4.9: An ideal transformer

4.9. Most importantly, transformers are characterized by its turns ratio between its primary and secondary. Let N_p represent the number of turns for the primary side and N_s represent the number of turns for the secondary side. The following are the ideal transformer equations which relate the primary and secondary voltages and currents.

$$\frac{V_p}{N_p} = \frac{V_s}{N_s} \tag{4.18}$$

$$N_p I_p = N_s I_s \tag{4.19}$$

Where N_p is the number of turns of the primary, N_s is the number of turns of the secondary, I_p is the current on the primary side, I_s is current on the secondary side, V_p is the voltage on the primary side and V_s is the voltage on the secondary side.

Fundamental field device theory states that transformers can be modeled by two coupled inductors with inductances defined by equations (4.20) and (4.21).

$$L_p = \frac{(N_p)^2 \mu A}{2\pi r_{ave}} \tag{4.20}$$

$$L_s = \frac{(N_s)^2 \mu A}{2\pi r_{ave}} \tag{4.21}$$

Where L_p is the inductance of primary, L_s is the inductance of secondary, A is the cross-sectional area of the core, r_{ave} is the average radius of the core, and μ is the magnetic permeability of the core material. Typically, L_p and L_s are written as follows:

$$L_p = (N_p)^2 P_p \tag{4.22}$$

$$L_s = (N_s)^2 P_s \tag{4.23}$$

Where P_p and P_s is referred to as the permeance of the primary and secondary respectively. For a symmetrical core and constant magnetic field, $P_p = P_s = P$ which implies

(from equations (4.22) and (4.23)) that the ratio of the primary and secondary inductances is equal to the ratio of the number of turns of the primary and secondary. Quantitatively,

$$\frac{L_p}{L_s} = \frac{N_p}{N_s} \tag{4.24}$$

If we write the Kirchhoff's law equations around the two loops in figure 4.9, we have the equations:

$$V_p = j\omega L_p I_p + j\omega M I_s \tag{4.25}$$

$$V_s = j\omega M I_p + j\omega L_s I_s \tag{4.26}$$

Where M is the mutual inductance defined as:

$$M = \sqrt{L_p L_s} \tag{4.27}$$

Combining equations (4.19) with (4.25) and (4.26), it can be shown that:

$$Z_p = \frac{V_p}{I_p} = j\omega\left(L_p + M\frac{N_p}{N_s}\right) \tag{4.28}$$

$$Z_s = \frac{V_s}{I_s} = j\omega\left(L_s + M\frac{N_s}{N_p}\right) \tag{4.29}$$

From equations (4.28) and (4.29), we see that Z_p and Z_s are impedances of the primary and secondary. Once Z_p and Z_s are determined, we can use the impedance equation to calculate its testability transfer function, T_f.

For an ideal transformer, the voltage and current of the secondary side are functions of the physical structure of the transformer and the voltage and current of the primary side. Hence, if the voltage and current of either side is known, then the voltage and current of the other side can easily determined (calculated). This implies that both sides are controllable and observable. For an ideal transformer, the mutual coupling is 100% between the two sides. Therefore, the testability transfer factor between the two sides is 1.0. Figure 4.10 shows the signal flow graph (SFG) for the ideal transformer.

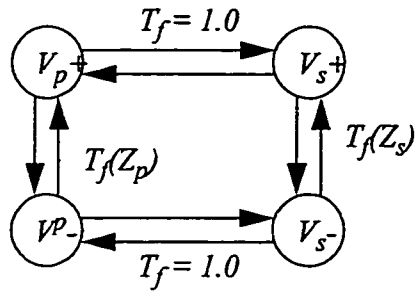


Figure 4.10: SFG for an ideal transformer

Single primary and multiple secondary transformer

We can extend the derivations above to include general transformers of single primary and multiple secondaries. The equations for an ideal transformer with a single primary and k secondaries are:

$$\frac{V_p}{N_p} = \frac{V_{s1}}{N_{s1}} = \frac{V_{s2}}{N_{s2}} = \ldots = \frac{V_{sk}}{N_{sk}} \tag{4.30}$$

$$N_p I_p = N_{s1} I_{s1} + N_{s2} I_{s2} + \ldots + N_{sk} I_{sk} \tag{4.31}$$

Kirchhoff's law equations can be written around each loop (both primary and secondaries) to result in the following:

$$V_p = j\omega L_p I_p + j\omega M_{ps1} I_{s1} + j\omega M_{ps2} I_{s2} + \ldots + j\omega M_{psk} I_{sk} \tag{4.32}$$

$$V_{s1} = j\omega M_{ps1} I_p + j\omega L_{s1} I_{s1} \tag{4.33}$$

$$V_{sk} = j\omega M_{psk} I_p + j\omega L_{sk} I_{sk} \tag{4.34}$$

Equations (4.30) through (4.34) can combined to yield:

$$Z_p = \frac{V_p}{I_p} = j\omega\left(L_p + M_{ps1}\frac{N_p}{N_{s1}} + M_{ps2}\frac{N_p}{N_{s2}} + \ldots + M_{psk}\frac{N_p}{N_{sk}}\right) \tag{4.35}$$

$$Z_{s1} = \frac{V_{s1}}{I_{s1}} = j\omega\left(L_{s1} + \frac{1}{k}M_{ps1}\frac{N_{s1}}{N_p}\right) \tag{4.36}$$

$$Z_{sk} = \frac{V_{sk}}{I_{sk}} = j\omega\left(L_{sk} + \frac{1}{k}M_{psk}\frac{N_{sk}}{N_p}\right) \tag{4.37}$$

In general, equations (4.35) through (4.37) can be summarized as:

$$Z_p = \frac{V_p}{I_p} = j\omega\left(L_p + N_p \sum_{i=1}^{k} \frac{M_{psi}}{N_{si}}\right) \tag{4.38}$$

$$Z_{si} = \frac{V_{si}}{I_{si}} = j\omega\left(L_{si} + \frac{1}{k}M_{psi}\frac{N_{si}}{N_p}\right) \tag{4.39}$$

Equation (4.38) describes the impedance seen at the terminals of the primary and equation (4.39) is the impedance seen at the terminals of secondary i where is: $1 \le i \le k$. Notice that equations (4.38) and (4.39) are equivalent to equations (4.28) and (4.29) with $k = 1$ (for a single primary).

Non-ideal effects

Two major non-ideal effects, core loss and series impedance, contribute to the transformer's deviations from ideal. The core is not infinitely permeable, it does require ampere turns to establish the flux, and in addition, there are internal energy losses in the core when the flux varies with time. The power absorbed by the transformer under the above conditions is lost to the circuit and goes into heat in the core. This non-ideal effect is referred to as core loss and is typically modeled [8] by a shunt combination of a resistive element (R_o) and an inductive element (L_o).

For an ideal transformer, we assumed that no complex power (real power or reactive voltamperes) was consumed in the transformer. In the actual transformer, both real power and reactive voltamperes are "consumed." The real power goes into the heat dissipated by the winding resistance and the reactive voltamperes go into the charging of the magnetic field between the windings. By placing resistance (R_w) in series with the ideal transformer mode, we can account for real power loss, and similarly, we can account for

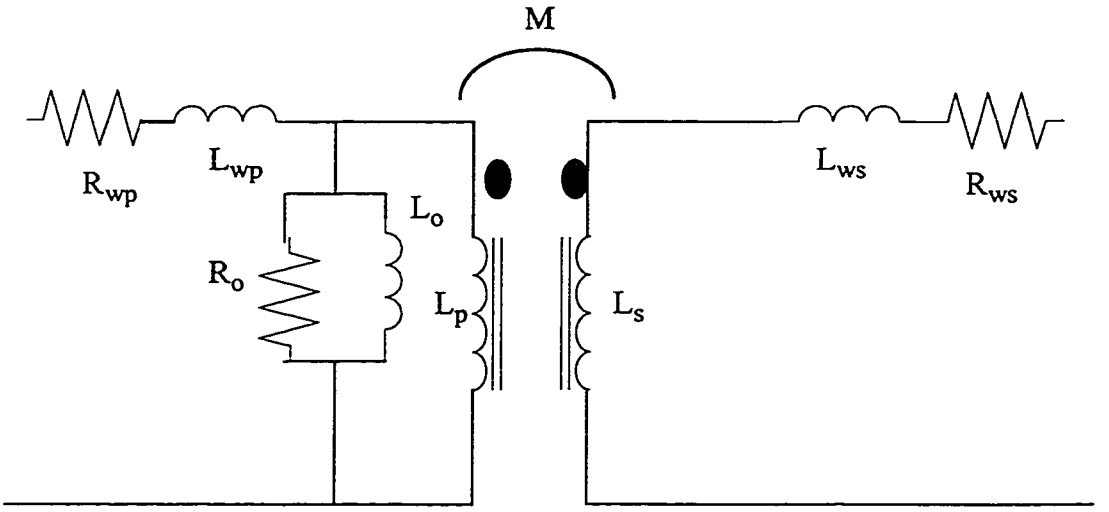


Figure 4.11: Complete circuit model for transformers

the reactive voltampere loss by placing an inductance (L_w) in series with the ideal transformer windings.

For an ideal transformer, the mutual coupling is a perfect 100%, however due to the two major non-ideal effects discussed above, the mutual coupling is less than 100%. The coefficient of mutual coupling, K, is defined as the ratio of the actual (real) amount of coupling to that of the theoretical value or:

$$K = \frac{M}{\sqrt{L_p L_s}} \tag{4.40}$$

Where M is the mutual inductance, L_p and L_s are the inductance on the primary and secondary side. Figure 4.11 shows the complete circuit model for an actual transformer and figure 4.12 shows the corresponding SFG. Typically, these parasitics are given as specifications for the transformer. Otherwise, these parasitics can also be determined by running an electrical level simulation.

4.2 HIERARCHICAL MODELING

Hierarchical modeling [30] refers to developing testability transfer factors for macro blocks, *i.e.* operational amplifiers, filters, phased locked loops, analog-to-digital

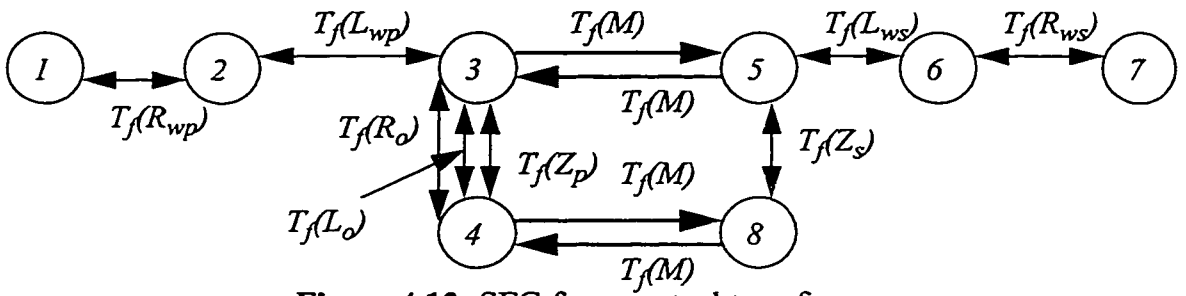


Figure 4.12: SFG for an actual transformer

and digital-to-analog converters, *etc.* A hierarchical model describes a box as a collection of interconnected smaller boxes called components or elements. The components are in turn modeled as an interconnection of lower level components. The bottom level boxes are primitive elements and their testability transfer factor is assumed to be known. Most primitive circuit elements were discussed in section 4.1.

TTF for MOSFETs was discussed in subsection 4.1.2. Further, MOSFETs can be viewed as a transconductance amplifier, a differential voltage is converted to an output current. Using a similar analysis, we can develop TTFs for the three other classes of amplifiers, voltage amplifier (operational amplifiers), current amplifiers and transresistance (differential input current is converted to an output voltage).

Figure 4.13 illustrates the equivalent circuit model for an operational amplifier (op-amp). It is very similar to the equivalent circuit model for the MOSFET (figure 4.6). The op-amp's output terminal can be "tickled" by stimulating the input terminals with a small differential signal. The output of the op-amp is equal to the differential gain times the voltage difference between the input terminals. An ideal op-amp will only respond to

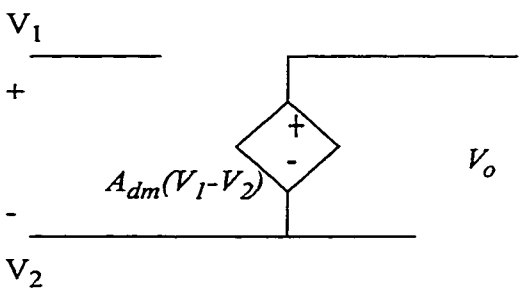


Figure 4.13: Equivalent circuit for an op-amp

differential signals and common mode signals are rejected. Thus, it is more appropriate to consider both the common mode gain and differential mode gain when calculating the TTF from the input terminals to the output terminal. For an op-amp with a high common mode rejection ratio (CMRR), it is easier to control the output and observe the inputs. With a low CMRR, it is more difficult to control the output and observe the inputs. Since the CMRR may have a wide range of values, the logarithm function is chosen for computation of the TTF. The underlying idea is similar to expressing the CMRR in decibels, however, the TTF values are bounded in the region between zero and one.

$$T_f(A_{dm}, A_{cm}) = 1 - \frac{1}{\log\left(\frac{A_{dm}}{A_{cm}} + 10\right)} \tag{4.41}$$

Where A_{dm} is the differential gain and A_{cm} is the common mode gain. A_{dm} can be a user defined function of frequency. For a first order approximation, A_{dm} is a three pole gain function [15] shown in equation (4.42).

$$A_{dm}(f) = \frac{A_{dc}}{\sqrt{\left(1 + \left(\frac{f}{f_1}\right)^2\right)\left(1 + \left(\frac{f}{f_2}\right)^2\right)\left(1 + \left(\frac{f}{f_3}\right)^2\right)}} \tag{4.42}$$

Where A_{dc} is the differential gain at DC, f is the frequency under analysis, f_1 is the first (dominate) pole, f_2 is the second pole and f_3 is the third pole. TTFs for other types of amplifiers can be developed using a similar analysis. The SFG for op-amp is shown in figure 4.14.

The model derived thus far is for an ideal op-amp. Non-ideal effects may be modeled by circuits elements, similar to the MOSFETs, and once the effects have been properly modeled by circuit elements, the TTFs can be evaluated. For hierarchical modeling, the approach is similar to network theory where a higher level block is model with a collection of interconnected lower level blocks (circuit equivalent). This traversal is continued till a level is reached where all its lower level blocks have its TTFs defined.

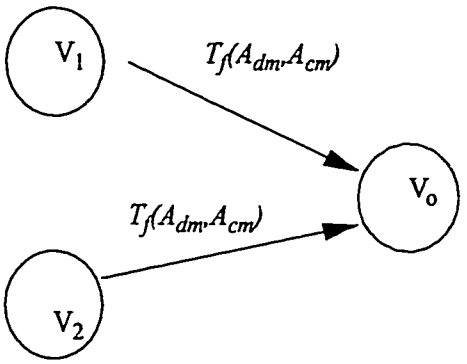


Figure 4.14: SFG for op-amps

4.4 CONTROLLABILITY, OBSERVABILITY AND TESTABILITY MEASURES

The input controllability of a component represents the ease of achieving an arbitrary signal value at the component's inputs and it depends on the controllabilities of the input connections and the components it is connected with. The output controllability of a component represents the ease of producing an arbitrary signal value on the outputs of the component. It depends on the input controllability and the TTF of the component or:

$$C_{out} = T_f C_{in} \tag{4.43}$$

Where C_{in} is the input controllability of the component, C_{out} is the output controllability of the component, and T_f is the TTF of the component.

To calculate the controllability of a node, transform the circuit schematic into a signal flow graph with each vertex representing the nodes in the circuit and each edge representing the components in the circuit. The direction of the edges represent the direction of signal flow through the component and the weight of each edge is the corresponding component's TTF. Uni-directional components such as operational amplifiers and diodes are represented by uni-directional edges while bi-directional components such as resistors, capacitors and inductors are represented by bi-directional edges. Bi-directional edges are modelled with two uni-directional edges with the same weights but opposite directions. The controllability for any node (or any vertex in the signal flow graph) i is:

$$C_i = \frac{1}{F_{in}} \sum_{m=1}^{F_{in}} C_m \bullet TTF_m \qquad (4.44)$$

Where C_i is the controllability of node *i*, F_{in} is the number of incoming edges to node *i*, C_m is the controllability at source node of incoming edge *m*, and TTF_m is the TTF of incoming edge *m*.

The output observability of a component represents the ease of determining whether or not the expected signal value occurs there by observing the signal values at the primary outputs of the circuit. The input observability of a component represents the ease of determining whether or not the expected signal value occurs there by observing the signal values at the primary outputs of the circuit. Since the TTF represents the ease of propagating a signal through the component, we have:

$$O_{in} = T_f \bullet O_{out} \qquad (4.45)$$

Where O_{in} is the input observability of the component, O_{out} is the output observability of the component, and T_f is the TTF of the component.

The observability of any node (or any vertex in the signal flow graph) is:

$$O_i = \frac{1}{F_{out}} \sum_{m=1}^{F_{out}} O_m \bullet TTF_m \qquad (4.46)$$

Where O_i is the observability of node *i*, F_{out} is the number of outgoing edges of node *i*, O_m is the observability at destination node of outgoing edge *m*, and TTF_m is the TTF of outgoing edge *m*.

Using non-weighted averages for equations 4.44 and 4.46 implies that each link has the same rank. This also implies that all components (TTFs) connected to that node has equal value and that all signals propagating to that particular node have equal importance. To use weighted averages will imply that some class of components and hence

some class of signals will have a higher rank then others. This is contrary to circuit analysis where all currents are summed to zero at each node with out regard to any particular types of currents or class of components having a higher rank.

Since test generation will be difficult if either controllability or observability is low, the function chosen for the testability analysis measure in this work is the geometric mean as shown in equation (4.47). However, the testability measure can be user defined for various different applications.

$$T_i = \sqrt{C_i \bullet O_i} \tag{4.47}$$

Where T_i is the testability of node i, C_i is the controllability of node i and O_i is the observability of node i. These testability measures are incorporated into a testability analysis procedure which is used to analyze analog circuit testability as well as providing a basis for testability enhancement and test generation.

4.5 TESTABILITY ANALYSIS PROCEDURE

The steps used by the testability analysis algorithm are as follows:

Step 1. Calculate the TTFs for each component using equations in this chapter. The TTFs may be stored in a library structure.

Step 2. Form signal flow graph from schematic. Each component becomes an edge and each node becomes a vertex in the graph. Each edge is weighted by the TTF of the corresponding component. The direction of each edge is determined by the type of components. Unidirectional components like operational amplifier and such are represented by uni-directional edges while bi-directional components like resistors and such are represented by bi-directional edges.

Step 3. Formulate controllability and observability equations at each vertex of the signal flow graph according to equations (4.44) and (4.46).

Step 4. Form controllability and observability matrix equations derived in Step 3.

$$\overrightarrow{SC} = [C]\overrightarrow{CM} \quad (4.48)$$

$$\overrightarrow{SO} = [O]\overrightarrow{OM} \quad (4.49)$$

Where equation (4.48) is the controllability matrix equation with $[C]$ as the controllability matrix, $\overrightarrow{CM}$ as the controllability measure vector and $\overrightarrow{SC}$ is the source controllability vector. Equation (4.49) is the observability matrix equation where $[O]$ is the observability matrix, $\overrightarrow{OM}$ is the observability vector and $\overrightarrow{SO}$ is the source observability vector.

Step 5. Solve equations (4.48) and (4.49) for $\overrightarrow{CM}$ and $\overrightarrow{OM}$ respectively. The controllability and observability vectors are the controllability and observability measures of each node in the circuit.

Step 6. Calculate the testability of each node according to equation (4.47)

Step 7. If a frequency response is desired, increment to the next frequency point and repeat steps 1 through 6 untill the range of frequency of interest is completed.

A flow chart summarizing the testability analysis formulation is shown in figure 4.15. A netlist describing the circuit is used as input. The result is a profile of the circuit's testability as a function of frequency (frequency response of testability). This profile shows how testability changes as frequency changes.

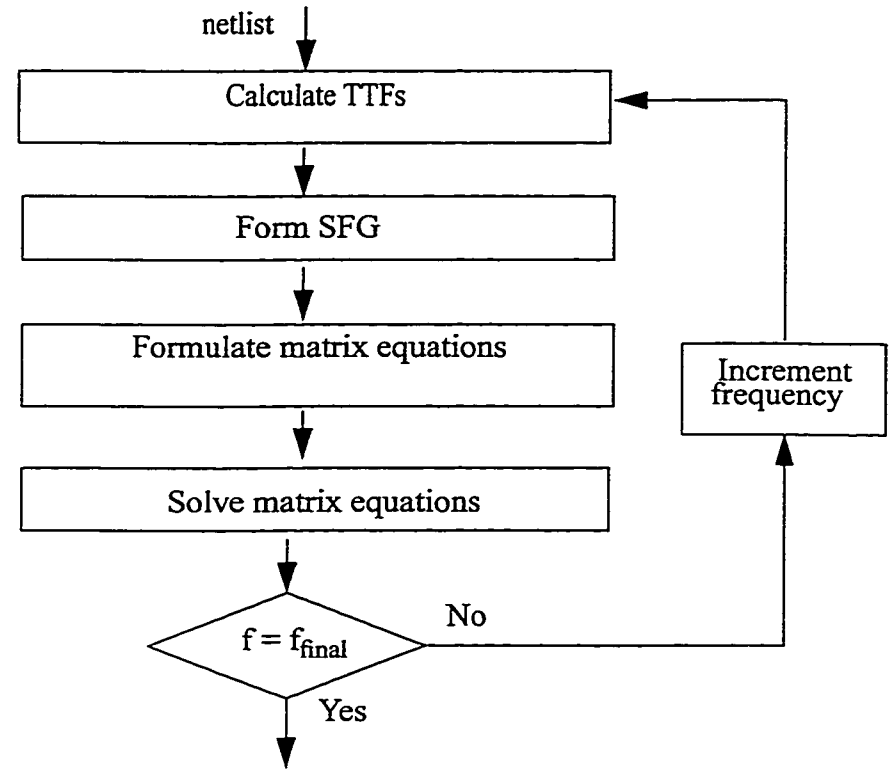


Figure 4.15: Testability analysis flow chart

Chapter 5
Applications for Testability Analysis

This chapter describes the applications for testability analysis. Testability information may be used to aid in design-for-test applications such as test point insertion. A test point may be either a control point, observe point or both. Testability information can also guide and ease the task of test generation. This intelligent guidance has the potential of substantially improving the performance of automatic test generation programs.

5.1 DESIGN-FOR-TEST APPLICATIONS

One use of the controllability and observability measures described in this dissertation is to aid in the design-for-test of analog circuits. The measures can be used to decide where to place control and observe points into the circuit. These points will not necessary be placed on nodes with the lowest controllability/observability values but on nodes whose enhanced controllability/observability enhances the total controllability/observability of the analog system.

To ensure a 100% controllability and 100% observability would require a test point to be inserted into every node in the CUT. This is highly impractical because of the exces-

sive amount of silicon area required to implement the test points, interconnection routing and I/O pins. Typically, the amount of silicon area allocated for DFT is limited to range from 5% - 15% of total chip area. Therefore, only a limited amount of test points can be inserted. The objective is to optimally place these test points to maximize the CUT's testability.

Testability enhancement problem statement:

Given a mixed-signal circuit with N total nodes and M total test points where N >> M. The objective is to optimally insert the M test points into N nodes such that it yields the maximum circuit controllability and observability.

Since the order of the selected nodes is not important, what is important is which nodes are selected. Mathematically, this problem is reduced to selecting M objects out of N where the *combination* is important and the *permutation* does not matter or equivalently:

$$\begin{bmatrix} N \\ M \end{bmatrix} = \frac{N!}{M!(N-M)!} \tag{5.1}$$

For instance, selecting nodes 2 and 8 as test points is the same as selecting nodes 8 and 2 as test points because if both nodes are selected then both nodes will be converted to be test points.

5.1.1 Objective functions

The circuit controllability objective function, C, can be described by:

$$C = \frac{1}{N} \sum_{i=0}^{N} c_i \tag{5.2}$$

Where C is the overall circuit controllability, c_i is the controllability at node i, N is the total number of nodes in circuit, and i is the node index (i^{th} node). The objective is then to insert

a control point(s) into node(s) which maximizes *C*.

An observability objective function, *O*, can be similarly described by:

$$O = \frac{1}{N} \sum_{i=0}^{N} o_i \tag{5.3}$$

Where *O* is the overall observability, o_i is the observability at node *i*, *N* is the total number of nodes in circuit, and *i* is the node index (i^{th} node). The objective is then to insert an observe point(s) into node(s) which maximizes *O*.

If an insert point is both a control and an observe point, then a third objective function, the testability objective function is needed. The testability objective function, *t*, is described by:

$$t = \frac{1}{N} \sum_{i=0}^{N} t_i \tag{5.4}$$

Where *t* is the overall circuit testability and t_i is the testability measure at node *i* and is a user defined function of c_i and o_i or:

$$t_i = f(c_i, o_i) \tag{5.5}$$

For any automatic test pattern generation algorithm to be successful, the generated test vector must be able to excite the fault targeted and the resulting error must be able to propagate to the primary outputs. Loosely translated, this implies that nodes inside the CUT must be both controllable and observable. These measures when mapped into the physical domain have the following properties:

Full controllability (1.0) refers to the fact that the node (or net) can be set to any specific value. The further it deviates from 1.0, the more difficult it is to set a value to that internal node.

Similarly, *full observability* (1.0) refers to the fact that any signal value on

an internal node (inside the CUT) can be propagated to the primary outputs. If a node has an observability measure of less than 1.0, it means that not all the information can be propagated to the outputs. The further it deviates from 1.0, the more difficult it is propagate any information from that internal node to the primary outputs.

These properties imply that a circuit with only high controllability does not result in a very testable circuit because if the errors (faults) cannot be observed, then faults may escape undetected. Also, a circuit with only high observability does not result in a very testable circuit because if faults can not be excited (controlled), then errors may not be observed. Therefore, one important property of a testable circuit is to have both high controllability and observability measures. The testability objective function may be stated mathematically as the geometric mean of both controllability and observability:

$$t_i(c_i, o_i) = \sqrt{c_i \bullet o_i} \qquad (5.6)$$

Each node in the CUT can be described by a triplet, (O, C, T), corresponding to observability, controllability and testability. To enhance controllability only (assuming the observability level is satisfactory), equation 5.2 may be used. To enhance observability only (assuming the controllability level is satisfactory), equation (5.3) may be used. Similarly, testability may be enhanced (if both its observability and controllability is unsatisfactory) by using equation (5.4). These three functions provide flexibility to designers for DFT applications as they can adapt to different DFT methodologies (controllability only, observability only, or testability).

5.1.2 Test point insertion algorithm

The test point insertion algorithm described in this sub-section was originally developed by the author and was later modified by Jinyan Zhang [85]. Based on the testability analysis methodology presented in previous chapters and the objective functions discussed in sub-section 5.1.1, we developed an algorithm to insert test points into a

mixed-signal system with the intent of increasing its overall testability metric. There are two strategies for test point insertion:

1. With a fixed number of test points available for insertion, find a placement such that it maximizes the testability (or controllability or observability) metric.

2. With a fixed threshold level for testability (or controllability or observability), minimize the number of test points required to achieve the fixed threshold level.

The circuit under test (CUT) is described by a SPICE compatible netlist, initial state of the circuit, and a list of primary inputs and outputs. The testability profile of the CUT before test points are inserted is calculated using the initial state. Then the nodes are sorted according to their fan-in and fan-out TTF values. If a device has a very low TTF value, it will reduce the testability transfer between its inputs and outputs. Simply put, devices with low TTF values restrict the amount of information which can propagate through the device from its inputs to its outputs. The controllability of the nodes connected to the outputs of such devices will be adversely affected. Similarly, the observability of the nodes connected to the inputs will be degraded. Nodes that are outputs of low TTF-value devices have greater potential to increase the overall controllability if re-configured as control points. The same applies to nodes that are inputs of low TTF-value devices for increasing the observability. Nodes which have large fan out can also improve the overall testability. The sorting algorithm [85] (developed by Jinyan Zhang) considers the above criteria. In the testability analysis algorithm presented in chapter 4, a circuit is represented by TTF matrices (*i.e.*, controllability matrix and observability matrix). By using the matrices, contributions of fan in branches and fan out branches are included.

For instance, the controllability matrix (C_{matrix}) is formulated from equation (4.44). Summation of the *i-th* row of C_{matrix} is the TTF value that node i gets from its fan-in nodes (denoted as $C_{i\text{-}in}$), whereas summation of the *i-th* column represents the fan-out condition of the node (denoted as $C_{i\text{-}out}$). Decreasing $C_{i\text{-}in}$ or increasing $C_{i\text{-}out}$ will

increase the possibility of selecting node i as a control point. Therefore, the nodes are sorted in the descending order of the values subtracting $C_{i\text{-}out}$ from $C_{i\text{-}in}$. The same algorithm is used with the observability matrix to obtain the search criteria for observable points.

Test point search begins from the top of the sorted sequence. The selected node is re-configured as a primary input or primary output or both according to it pre-defined type. After the re-configuration, the testability is re-evaluated and compared with the previous value. If it is greater than the previous, the point is retained. Otherwise, the selected node is discarded and the next node in the sorted sequence is selected. The search stops when the pre-defined number of test points has been selected or a given maximum number of points have been considered.

5.2 AUTOMATIC TEST PATTERN GENERATION APPLICATIONS

Analog test generation in its simplest form can be viewed as finding an input which maximizes the error between the good and the faulty circuits. For ease of notation, we denote,

$$E = S_g - S_f \tag{5.7}$$

Where E is the magnitude of the error, S_g is the signature for the good circuit, and S_f is the signature for the faulty circuit. We can apply this principle to the testability profiles discussed in the previous sections [31], [32], and [33]. Let

$$E(F) = \left| T_g(F) - T_{fi}(F) \right| \tag{5.8}$$

Where E is the magnitude of the error response, F is the frequency variable, T_g is the testability profile of the good circuit, and T_{fi} is the testability profile of the faulty circuit with fault f_i. Hence, we can define the detectability of fault fi, D_{fi}, as the integral of the error response over a frequency range. Equivalently,

$$D_{fi} = \int_{F_{init}}^{F_{final}} E(F)dF = \int_{F_{init}}^{F_{final}} \left|T_g(F) - T_{fi}(F)\right| dF \qquad (5.9)$$

The larger D_{fi}, the easier it is to detect the fault *fi*. The smaller D_{fi}, the more difficult it is to detect the fault *fi*. If $D_{fi} < D_{thrs}$, then our method cannot generate a test vector to detect fault *fi*. D_{thrs} is a user defined threshold which depends on the amount of tolerable variations. For any $D_{fi} >= D_{thrs}$ we can generate a test vector to detect fault *fi*.

5.2.1 Multi-Frequency Automatic Test Pattern Generation

Automatic test pattern generation procedure

Step 1: Calculate $T_g(F)$.

Step 2: Calculate $T_{fi}(F)$ for fault *fi*.

Step 3: Calculate D_{fi} according to equation (5.9).

Step 4: If $D_{fi} =< D_{thrs}$, then we cannot generate test to detect fault f_i, skip to step 9. Otherwise continue to step 5. D_{thres} is a user defined threshold which depends on the amount of tolerable variations.

Step 5: Transform $E(F)$ into a filter frequency response $|E(e^{j\omega})|$. $E(F)$ contains the frequencies or range of frequencies in which $T_g(F)$ differs from $T_{fi}(F)$. This implies that at these frequencies or range of frequencies, the response of the good circuit differs from that of the faulty circuit. Therefore, we want to generate a signal with these frequency components. To generate $|E(e^{j\omega})|$ from $E(F)$, we place a zero in $|E(e^{j\omega})|$ for all the frequencies or range of frequencies in which $E(F)$ is zero and place a one in $|E(e^{j\omega})|$ for all the frequencies or range of frequencies in which $E(F)$ is non-zero. The phase of $E(e^{j\omega})$ is made linear to eliminate any phase distortion.

Step 6: Calculate inverse transform of $E(e^{j\omega})$ to get $e(t)$.

Step 7: Convolve $e(t)$ with $\delta(t)$ to produce $v_{fi}(t)$. $e(t)$ is the system response of the filter derived from $E(F)$, therefore, by convolving the system response with the $\delta(t)$ function, we produce a signal with the frequency components from step 5. This signal can then be use as a test pattern to detect fault *fi*. $v_{fi}(t)$ can be produce by an arbitrary waveform generator (AWG) in the time domain by using piece wise linear segments.

Step 8: Add $v_{fi}(t)$ to test set **V**.

Step 9: Remove fault *fi* from fault list.

Step 10: Select another fault from fault list and go back to step 2.

Step 11: Repeat till fault list is empty.

5.2.2 Signature Analysis

In the dynamic case, due to noise and tolerance the exact comparison of time-domain waveforms is an experimental impossibility. Hence, any detection scheme must use some measure of the waveforms as a basis for comparison. Such measures on the waveform are also referred to as signatures of the waveform. One popular signature is the frequency response of the circuit or system under consideration. The signatures are obtain by taking time domain measurements at the primary outputs of the circuit and then performing the Fourier transform on the measured data via software.

Figure 5.1 shows the signature analysis methodology used. The test set **V**, generated by the procedure described in subsection 5.2.1, is used as input for both the good circuit and the faulty circuits. The fast Fourier transform (FFT) is then computed for both the good and faulty outputs. The outputs following the fast Fourier transform are then subtracted which takes the difference of the two signatures. If the absolute value of ΔS is equal to or greater then S_{thres}, then the fault *fi* is detected by the test set **V**. Otherwise, the fault *fi* is not detected (absolute value of ΔS less then S_{thres}). S_{thres} is a user defined threshold which depends on the accuracy of the equipment used.

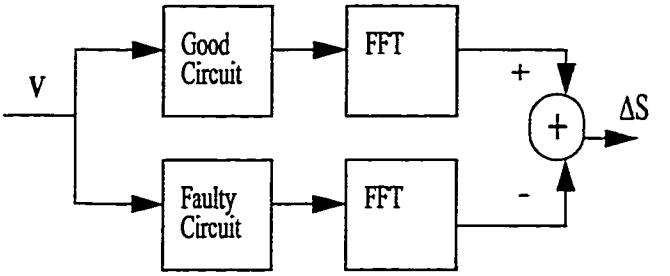


Figure 5.1: Signature analysis methodology

5.2.3 Test Set Compaction

The above algorithm can generate a test vector for each detectable fault. For n_i detectable faults, the test set generated is of size n_i. Clearly, we see that the size of the generated test set is linearly proportional (constant of proportionality equals to 1) to the number of detectable faults. This is hardly an optimal test set since for 1000 detectable faults, then the generated test set contains 1000 test vectors. Therefore, our objective here is to maximally reduce the number of test vectors in the generated test set while maximizing the number of detectable faults.

Let n_f be the total number of faults in the fault list and n_i be the number of faults detected by the test vector v_{fi}, then the fault coverage is defined as the ratio of n_i over n_f or:

$$C_{fi} = \frac{n_i}{n_f} \tag{5.10}$$

Test set compaction procedure

Step 1: Calculate fault coverage for each fault by performing fault simulations for the generated test set (V) and the targeted fault list (F).

Step 2: Remove the test vector with the highest fault coverage from V and add it to T, the compacted optimal test set.

Step 3: Remove all the faults from the fault list covered by the test vector added to T in step 2.

Step 4: Repeat steps 2 and 3 till either F or V is empty.

The results of this procedure is an optimal test set (T) which is a subset of the generated test set (V). T can detect the same faults as V but with less vectors. In some extreme cases, T may equal V. This may happen if the faults are so unique that only one specific test vector can detect it. These types of faults are rare.

5.3 SUMMARY

In this chapter, we have discussed the DFT and automatic test generation applications using the testability formulation discussed in chapter 4. We demonstrated how the testability information can be used in test point insertion. The algorithm developed by Zhang [85] is based on searching through a sorted sequence of nodes. The sequence is sorted based on the nodes controllability and observability metrics and TTF values of the components which are connected to the node.

Testability information is also used in the automatic test pattern generation algorithm discussed in this chapter. The algorithm used the testability information as the basis for its multi-frequency test pattern generation procedure. A test pattern is generated for each detectable fault ($D_{fi} > D_{thrs}$) and all the test patterns together constitute the test set for a particular fault list. A signature analysis methodology was also introduced to evaluate the quality of the test set. Since the test set generated can potentially be large, a test set compaction algorithm was also discussed and we showed how the generated test set can be compacted into a smaller test set. In the next chapter, we discuss in detail the CAD implementation of the algorithms and methodologies presented in this chapter.

Chapter 6
CAD Implementation

The algorithms and methodologies presented in previous chapters have been implemented as part of a verification tool for an IC design and test software system called REIGN. REIGN represents a unified suite of software tools designed to analyze a circuit's testability, provide testability information for subsequent DFT applications and generate test patterns. This chapter presents details regarding implementation of the various components of the REIGN architecture and the overall framework which permits testability analysis of realistic mixed-signal circuits. Specific examples of REIGN applications are also presented.

6.1 REIGN SYSTEM OVERVIEW

Figure 6.1 shows an overview of the REIGN system architecture. The input consists of:

- A SPICE compatible netlist describing the DUT design.
- An initialization file listing the primary inputs and outputs of module(s).
- A file containing the list of faults to be studied. This file is only required if test

generation is to be performed.

- A partitioning file containing user's directives on how to partition the design into smaller modules for the purpose of testability analysis. The default is to output the testability values of all nodes in the design and to analyze the entire design without partitioning.

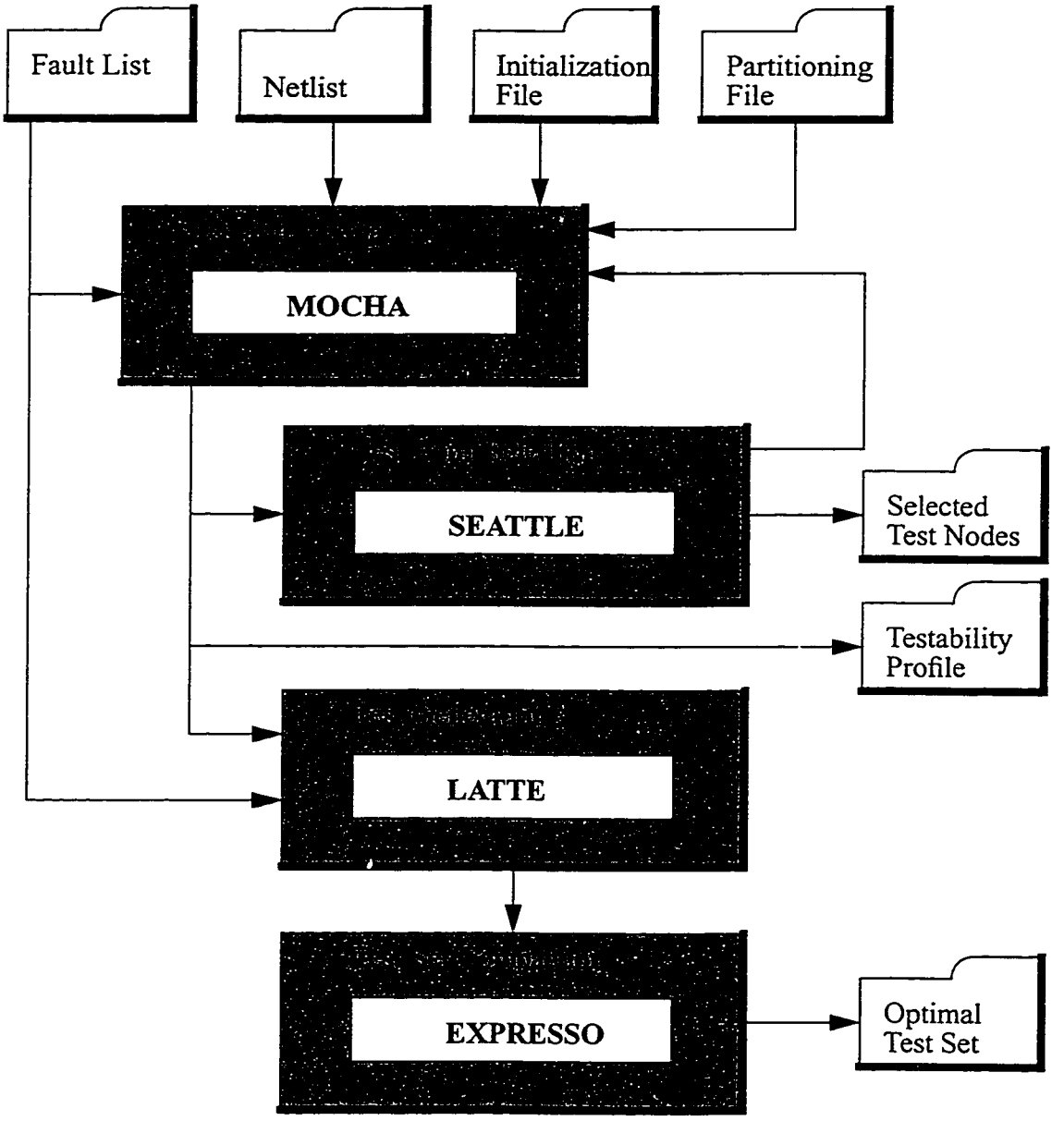


Figure 6.1: System architecture for REIGN

The system generates three types of outputs:

- Testability output - This type of output displays the DC, AC and transient testability profiles of the DUT.
- DFT output - A list of nodes selected as test points
- Test set output - A set of test vectors

The testability and DFT outputs can be used to redesign the circuit if necessary while the set of test vectors can be used to test the circuit.

6.2 TESTABILITY ANALYSIS METHODOLOGIES IN MOCHA

6.2.1 DC Testability Analysis

DC analysis analyzes the circuit at DC (frequency = 0). At DC, capacitors are open circuits and inductors are short circuits. The flow chart outlining the DC analysis procedure is shown in figure 6.2. The parsed netlist together with the partitioning information are used to extract the appropriate models from the components library to use when formulating the controllability/observability nodal equations. The models are the analytical TTF functions described in detail in chapter 4.

Along with the initialization file, controllability and observability equations are formulated at each node. The formulation of these equations are also described in chapter 4. The default initialization is that all primary inputs are perfectly controllable and all primary outputs are perfectly observable. However, users can define any initialization which fits the design requirements. Since the formulated equations constitute a system of linear equations, the equation solver becomes a matrix solver. Furthermore, these are linear equations, therefore, solution convergence is not an issue.

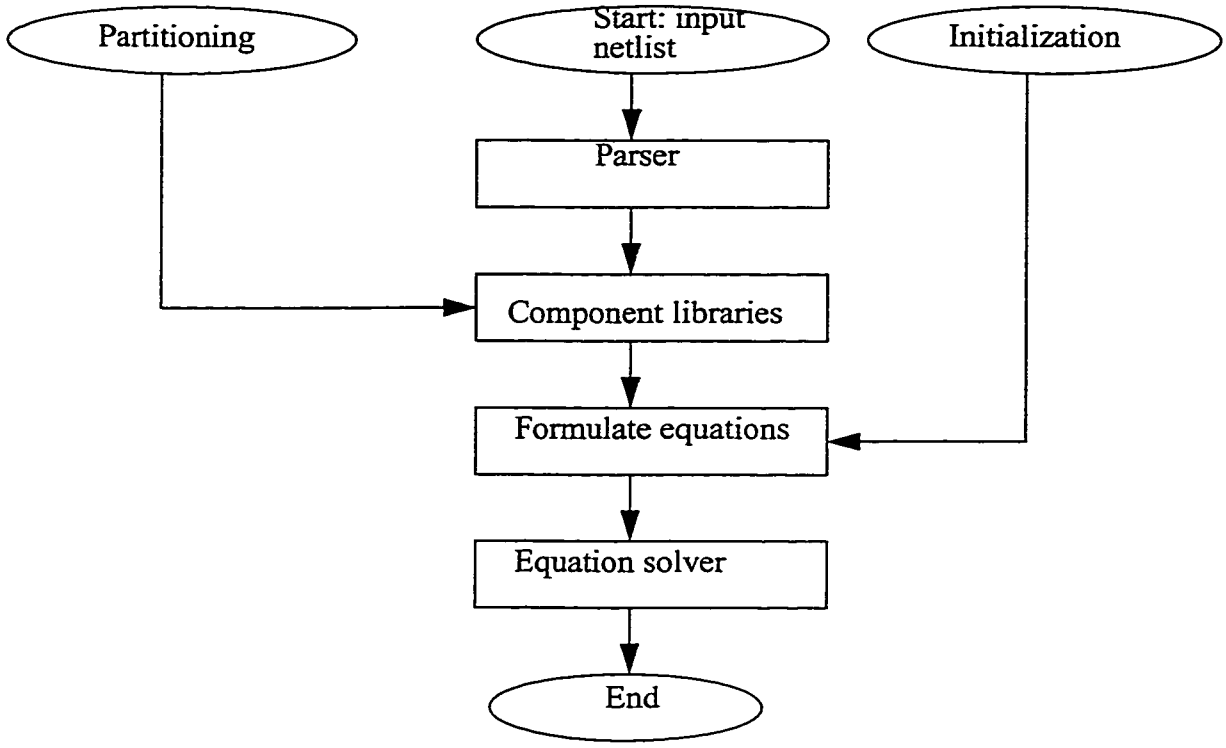


Figure 6.2: Flow chart for DC testability analysis

The LU decomposition and sparse matrix techniques were considered for use as the matrix solver however, we decided not to employ sparse techniques because of the following reasons. The LU decomposition technique is selected because of its simplicity and ease of implementation. The solution method is general enough for most applications.

In most circuits, nodes are connected to a relatively few other nodes. Rarely (almost improbable) do nodes connect to every other nodes in the circuit. As a result, the matrix which comprises the system of linear equations is *sparse*. A system of linear equations is called *sparse* if only a relatively small number of its matrix elements are nonzero. It is inefficient to use general solution methods because most of the computation time is wasted on performing arithmetic operations on an element with a zero entry. Furthermore, it is also wasteful to reserve storage for zero elements. Sparse techniques are fundamentally decomposition schemes but carefully optimized so as to minimize the number of fill-ins, initially zero elements which must become nonzero during the solution process, and for which storage must be reserved. Direct methods for solving sparse equations, then,

```
Begin
        fr = 0;                          /* Start of analysis */
        While (fr < fr_final)
                DC_Analysis(fr, C);      /* Perform DC analysis at freq. fr */
                fr ← fr + Δfr;           /* Increment to next freq. step */
                Update_Freq(fr, C);      /* Update the circuit */
        End
End
```

Figure 6.3: AC testability analysis algorithm

depend crucially on the precise pattern of sparsity of the matrix [56]. Since the pattern of sparsity depends on the circuit, it is impossible to select one sparsity pattern over another. Therefore, we would have had to select a solution method general enough to include a wide range of sparsity patterns which then is akin to the LU decomposition technique. Also, the LU decomposition does not depend on the pattern of sparsity of the matrix, further supporting our decision of selecting the LU decomposition technique.

6.2.2 AC Testability Analysis

Solving for testability as a function of frequency is referred to as AC testability analysis (how testability changes with respect to frequency). The algorithm used is listed in figure 6.3. AC testability analysis is reduced to solving for the DC solution at each at frequency step. Most of the models (capacitors, op-amps, *etc.*) are frequency dependent (a function of frequency), therefore, when the frequency is incremented, the models also need to be updated. Let C represent the circuit under analysis and fr is the frequency variable. The algorithm starts with analyzing the circuit (C) at DC. The frequency variable (fr) is incremented by a Δfr and the circuit is updated with the new fr. If fr is still within the range of analysis, then a DC testability analysis is performed on C. This cycle is repeated until the frequency range for analysis is completed. The frequency range of analysis is defined by users and that depends on the range of interest for the users.

6.3 DFT USING SEATTLE

The testability profiles calculated by MOCHA are employed by the SEATTLE routine to enhance the testability of the CUT. A flow chart illustrating the essential components of SEATTLE is shown in figure 6.4. The SEATTLE routine implements the Zhang algorithm [85] described in chapter 5. The SEATTLE routine utilizes a dual feedback loop structure to search through the design space to find the optimal node to insert test point(s).

The objective functions (equations (5.2), (5.3) and (5.4)) are calculated in the box labeled "testability calculation". The outer feedback loop illustrated in figure 6.4 is for test point insertion while the inner feedback loop is for optimization. There are two constraints for test point insertion: (i) the desired level of testability is lower bounded and (ii) the number of available test points to inserted is fixed. Ideally, a designer would like to have 100% controllability and a 100% observability which means all nodes are perfectly controllable and observable. This is not practical because it means a test point must be inserted into every node of the circuit. Furthermore, this would require a significant amount of silicon area for the test points alone not to mention the additional I/O pins and interconnection routing. Depending on the design and application, it is imperative to achieve a certain level of testability (user defined) given an estimate number of test points (user defined). The objective is to place a test point(s) as to maximize testability.

6.4 FAULT BASED AUTOMATIC TEST GENERATION (LATTE)

The LATTE routine is the implementation of the multi-frequency ATPG algorithm described in chapter 5. The algorithm is listed in figure 6.5. Let:

F = Faults in the fault list ($F = \{f_1, f_2, f_3, \ldots\}$)

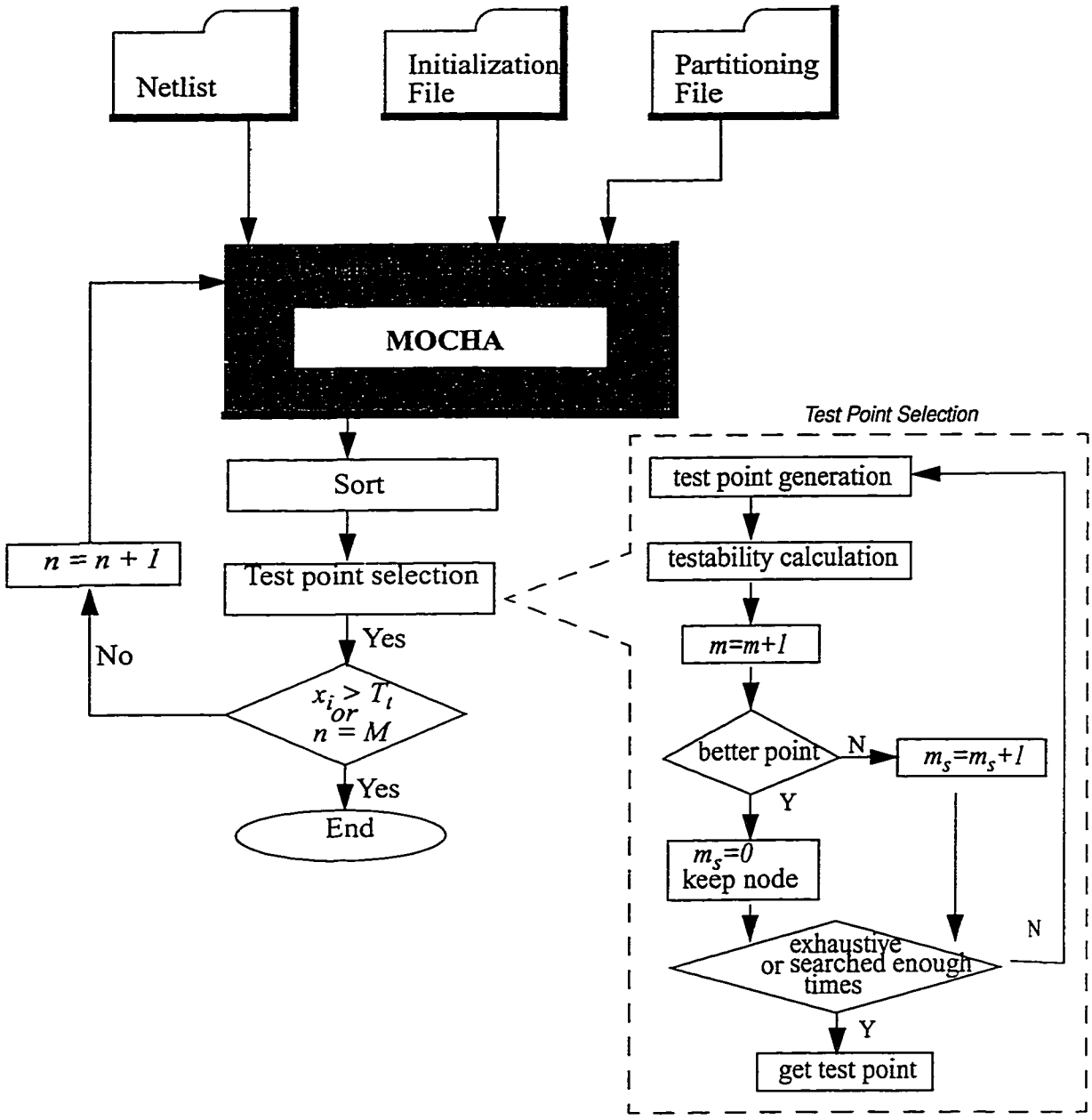
T_g = Testability profile of the good circuit

T_{fi} = Testability profile of the faulty circuit with fault f_i

D_{fi} = Detectability of fault f_i

D_{thrs} = User defined detectability threshold

fr = Linear frequency variable



M = number of available test points to insert
n = number of test points inserted
T_t = Desired level of testability
x_i = Testability at iteration i
m = Number of nodes searched
m_s = Number of searches performed

Figure 6.4: Flow chart for SEATTLE

Begin
    Calculate T_g;
    While $F \neq \varnothing$
        Calculate T_{fi};
        Calculate D_{fi};
        If ($D_{fi} > D_{thrs}$)
            $E(e^{j\omega}) \leftarrow E(fr)$;
            $e(t)$ = INVERSE_TRANSFORM($E(e^{j\omega})$);
            $v_{fi}(t)$ = CONVOLVE($e(t)$, δ));
            $V \leftarrow v_{fi}(t)$;
        End_i
        $F \leftarrow F - f_i$;
    End
End

Figure 6.5: LATTE algorithm

ω = Radian frequency variable ($\omega = 2\pi fr$)

δ = Delta function

v_{fi} = Test vector generated to detect fault f_i

V = Generated test set ($V = \{v_{f1}, v_{f2}, v_{f3}, ...\}$)

The INVERSE_TRANSFORM function is implemented with the inverse fast Fourier transform (IFFT) algorithm. To take advantage of the IFFT algorithm, the number of points, P, required for the IFFT must be a power of 2. In REIGN, P is set to 512, however, this is a user defined parameter.

When selecting P, several aspects must be considered. Since we are using discrete methods to compute a continuous result, large P will provide more accurate results. The main drawback is the computation time required. With a smaller P, the time required to compute the results will be faster, however the results are also less accurate. In selecting P, the user must decide on the trade-off between run time and result accuracy.

Any convolution operation which has the δ function as one of its operands can be

reduced to shifting operations [52]. Hence, the CONVOLVE function just performs shifting operations.

6.5 EXPRESSO - TEST SET COMPACTION

EXPRESSO is the name of the routine which implements the optimal test set design algorithm described in chapter 5, subsection 5.2.3. The algorithm is listed in figure 6.6. Let:

F = Faults in the fault list ($F = \{f_1, f_2, f_3, ...\}$)

V = Generated test set ($V = \{v_{f1}, v_{f2}, v_{f3}, ...\}$)

T = Final optimal test set

C = Fault coverage for each test vector in V ($C = \{c_1, c_2, c_3, ...\}$)

Without a reliable and readily available concurrent (or parallel) analog/mixed-signal fault simulator, the FAULT_SIMULATE routine was accomplished through serial fault simulation using the HSPICE circuit simulator. Each test vector in the generated test set (V) was fault simulated with each fault in the fault list (F). The result is a fault coverage parameter for each test vector. Recall that fault coverage is the ratio of number of detected faults to the total number of faults in the fault list by the generated test vector. Once that

```
Begin
        C = FAULT_SIMULATE(V,F);
        Do
                T ← V[MAX(C)];
                V ← V − V[MAX(C)];
                F ← F − F[MAX(C)];
                C ← C − MAX(C);
        Repeat till F = ∅ or V = ∅;
End
```

Figure 6.6: EXPRESSO algorithm

the fault coverage has been calculated for each test vector, select the test vector which has the highest fault coverage, v_{fi}, and add it to T, the optimal test set ($T \leftarrow V[MAX(C)]$). Remove v_{fi} from V ($V \leftarrow V - V[MAX(C)]$), the faults covered by v_{fi} from F ($F \leftarrow F - F[MAX(C)]$) and the fault coverage for v_{fi} from C ($C \leftarrow C - MAX(C)$). Repeat this selection process till either the fault list (F) or the generated test set (V) is empty. The result is an optimal test set (T) which is a subset of V detecting the equal amount of faults as V but with less test vectors.

6.6 APPLICATIONS TO MIXED-SIGNAL CIRCUITS

The algorithms discussed in this dissertation have been implemented with various C code, shell scripts and MATLAB code. The result is suite of software packages collectively referred to as REIGN. Reign was applied to various circuits from the industry and the suite of analog and mixed-signal benchmark circuits - first release [39] and the results are discussed below.

6.6.1 ATPG for analog circuits and systems

Continuous time state variable filter

The continuous time state variable filter, one of the circuits from the suite of analog and mixed-signal benchmark circuits [39], under consideration is shown in figure 6.7

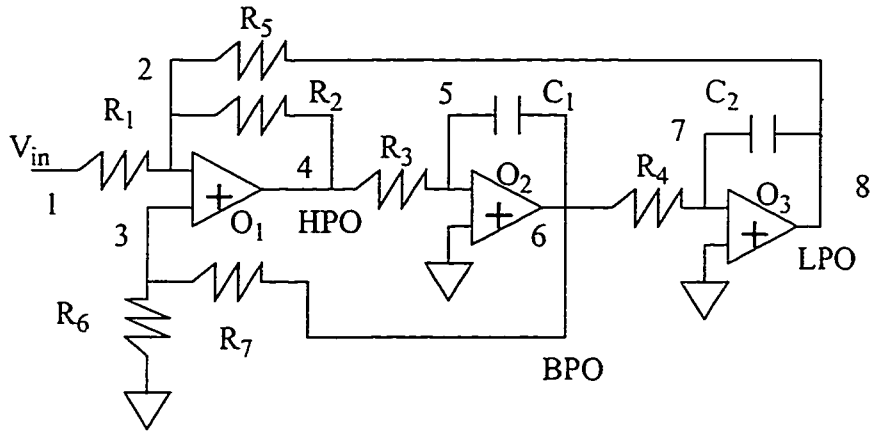


Figure 6.7: State-variable filter

where HPO is high pass output, BPO is band pass output, and LPO is low pass output. The current benchmark circuit has the following values: $R_1 = R_2 = R_3 = R_4 = R_5 = 10K$, $C_1 = C_2 = 20nF$, $R_6 = 3K$ and $R_7 = 7K$. Figure 6.8 shows the schematic of the op-amp used in this circuit.

Table 4 shows a sample list of faults and the associating detectabilities. Figure 6.9 is a plot of fault detectabilities as a function of percent variation. V_{tn} is the threshold voltage for N-type MOSFETs and since it causes deviations in a large area mainly affecting a large number of N-type MOSFETs, we see that its detectabilities are the highest. The aspect ratio of transistors M_{18} in all the op-amps was also varied up +/- 50%.

Table 4: Sample list of faults for the state variable filter

fi	*Faults*	D_{fi}
f_1	In O1, gates of M7 and M9 are shorted	0.0052
f_2	In O1, gate and source of M9 are shorted	0.2776
f_3	In O1, M17 drain open	0.9204
f_4	In O1, M4 gate open	0.0894
f_5	In O2, gates of M7 and M9 are shorted	0.8145

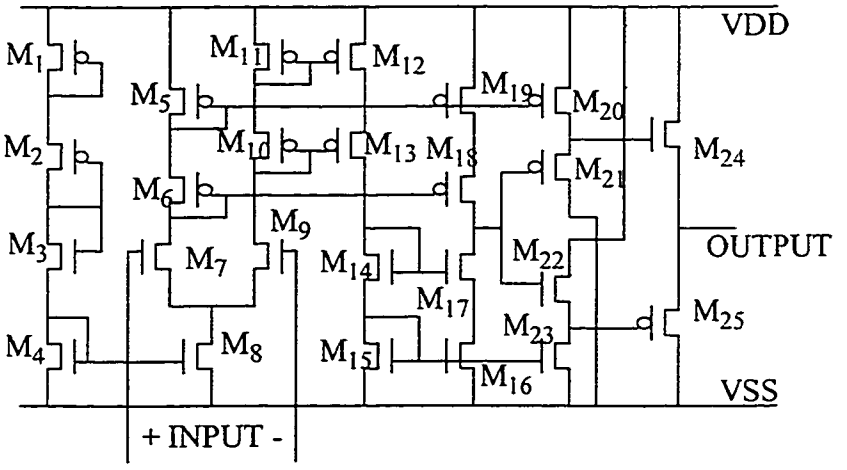


Figure 6.8: Schematic of op-amp

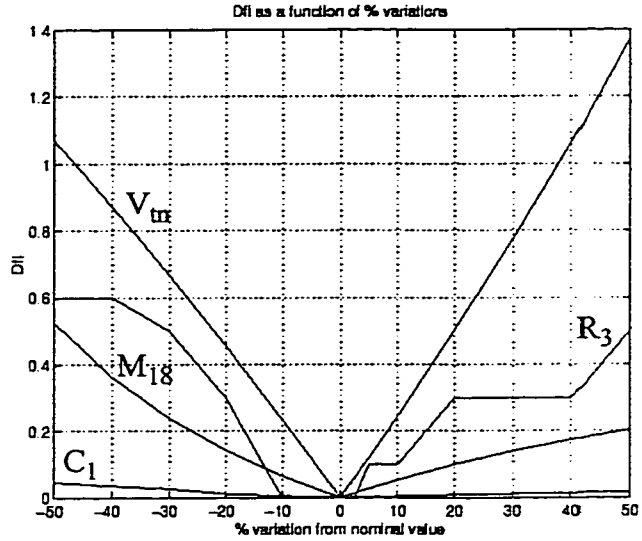


Figure 6.9: D_{fi} vs. % variations

Table 4: Sample list of faults for the state variable filter

fi	*Faults*	D_{fi}
f_6	In O2, gate and source of M9 are shorted	0.3304
f_7	In O2, M17 drain open	0.9878
f_8	In O2, M4 gate open	0.0893
f_9	In O3, gates of M7 and M9 are shorted	0.8108
f_{10}	In O3, gate and source of M9 are shorted	0.3304
f_{11}	In O3, M17 drain open	0.9804
f_{12}	In O3, M4 gate open	0.0893
f_{13}	V_{tn} + 30%	0.7754
f_{14}	V_{tn}- 30%	0.6685
f_{15}	R3 - 50%	0.6000
f_{16}	R3 + 50%	0.5000
f_{17}	C1 - 50%	0.0461
f_{18}	C1 + 50%	0.0165

Figures 6.10 and 6.11 shows the test patterns generated and the signatures of the HPO, BPO and LPO outputs of the state variable filter for faults f_1 and f_7. Fault f_1 has the lowest detectability and we see that is reflected in its signatures. The signatures at the BPO and LPO of the good and faulty circuits are very similar in that they both have similar profiles and magnitudes. On other hand, the signatures for f_7 are significantly different at all three outputs. Although the HPO and BPO have similar profiles, the magnitudes between the good and the faulty circuits are significantly different.

For the 18 faults listed in table 4, the LATTE ATPG algorithm would generate 18 test vectors, one for each fault. By applying the test compaction algorithm described on the test set only eight vectors are required to test for all 18 faults listed in table 4. The generated test set $V = \{v_{f1}, v_{f2}, v_{f3}, v_{f4}, v_{f5}, v_{f6}, v_{f7}, v_{f8}, v_{f9}, v_{f10}, v_{f11}, v_{f12}, v_{f13}, v_{f14}, v_{f15}, v_{f16}, v_{f17}, v_{f18}\}$ when applied to the test compaction algorithm reduced to $T = \{v_{f1}, v_{f4}, v_{f5}, v_{f6}, v_{f7}, v_{f13}, v_{f15}, v_{f17}\}$ or a 55.56% reduction.

Figure 6.11 show the test patterns generated and the signatures of the HPO, BPO and LPO outputs of the state variable filter for faults f_3, f_7, and f_{11}. The same test vector was used to test for all three faults although only fault f_7 is shown. Although the HPO and

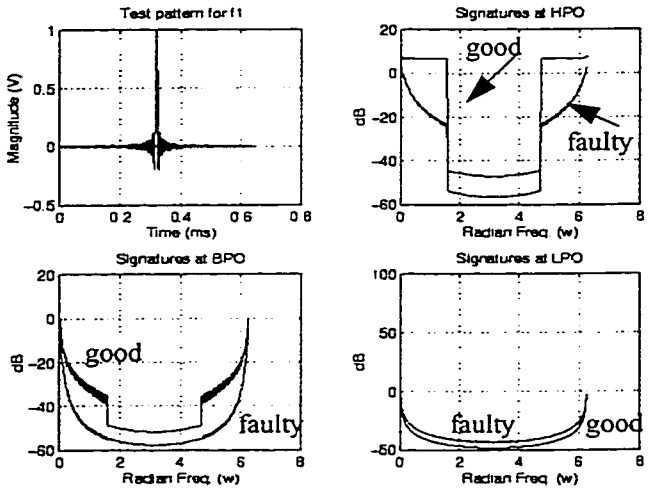


Figure 6.10: Test pattern and signatures for f_1

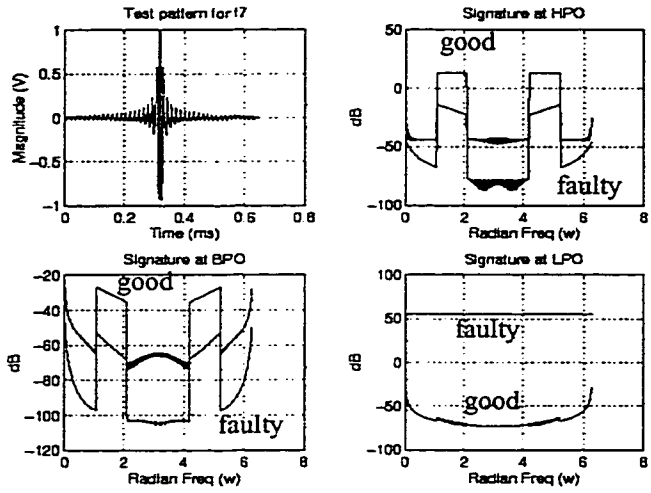


Figure 6.11: Test pattern and signatures for f_7

BPO have similar profiles, the magnitudes between the good and the faulty circuits are significantly different.

Leapfrog filter

The leapfrog filter, a benchmark [39] circuit, under consideration is shown in figure 6.12. All resistors are 10K, $C_1 = C_4 = 0.01\mu F$, and $C_2 = C_3 = 0.02\mu F$. The schematic for the op-amp is shown in figure 6.8. Primary input is node 1 and primary output is node 13.

Table 5 shows a sample list of faults and the associated detectabilities. Figure 6.13

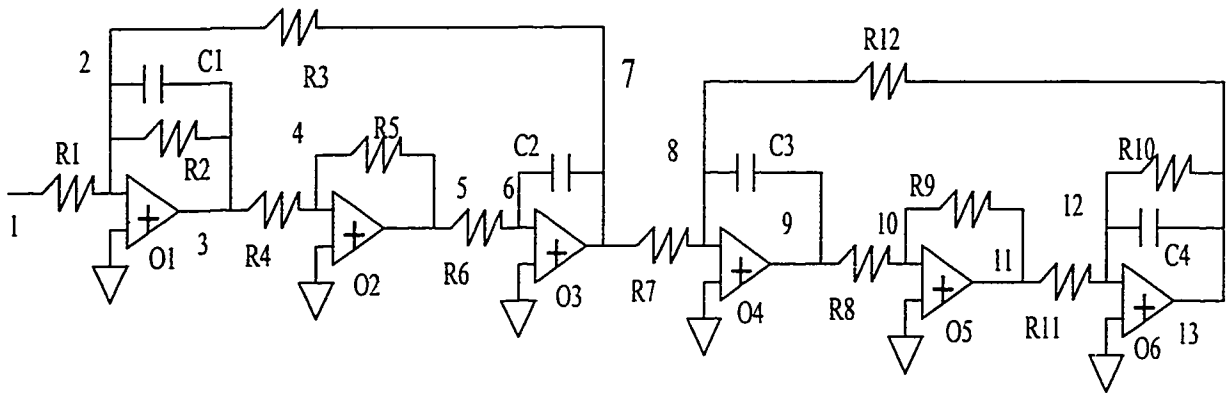


Figure 6.12: Leapfrog filter

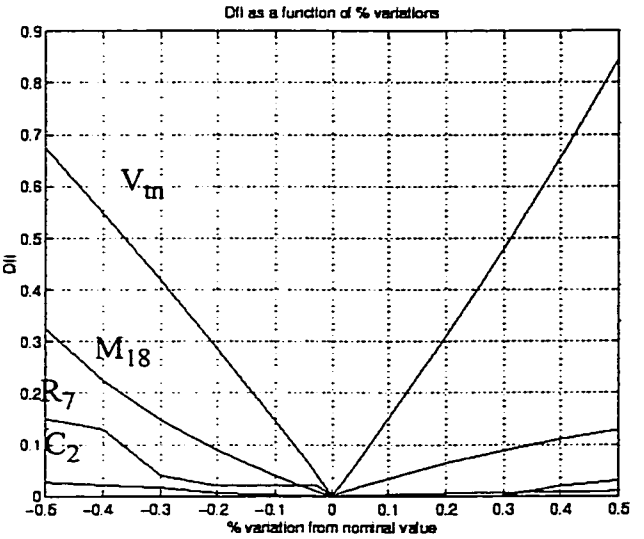


Figure 6.13: D_{fi} vs. % variations

is a plot of fault detectabilities as a function of percent variation. We see that variations in V_{tn} produce the largest detectability. This can be explained, as before in the state variable filter case, by that V_{tn} causes deviations in a large area mainly affecting a large number of N-type MOSFETs. The aspect ratio of MOSFET M_{18} in all the op-amps is varied up to +/ - 50%. Similar to the state variable filter case, variations in one of the capacitors produced the lowest detectability.

Table 5: Sample list of faults for the leapfrog filter

fi	*Faults*	D_{fi}
f_1	In O1,gates of M7 and M9 are shorted	0.1307
f_2	In O1, gate and source of M9 are shorted	0.0031
f_3	In O1, M17 drain open	0.2742
f_4	In O1, M4 gate open	0.0214
f_5	In O2, gates of M7 and M9 are shorted	0.2741
f_6	In O2, neg terminal & node 6 short	0.0159
f_7	In O2, M17 drain open	0.2590

Table 5: Sample list of faults for the leapfrog filter

fi	*Faults*	D_{fi}
f_8	In O2, M4 gate open	0.0215
f_9	In O3, gates of M7 and M9 are shorted	0.3001
f_{10}	In O3, gate and source of M9 are shorted	0.0314
f_{11}	In O3, M17 drain open	0.2672
f_{12}	In O3, M4 gate open	0.0218
f_{13}	In O4, gates of M7 and M9 are shorted	0.3018
f_{14}	In O4, gate and source of M9 are shorted	0.1542
f_{15}	In O4, M17 drain open	0.2709
f_{16}	In O4, M4 gate open	0.0219
f_{17}	In O5, gates of M7 and M9 are shorted	0.3675
f_{18}	In O5, gate and source of M9 are shorted	0.0862
f_{19}	In O5, M17 drain open	0.2741
f_{20}	In O5, M4 gate open	0.0222
f_{21}	In O6, gates of M7 and M9 are shorted	0.3657
f_{22}	In O6, gate and source of M9 are shorted	0.1978
f_{23}	In O6, M17 drain open	0.3563
f_{24}	In O6, M4 gate open	0.0231
f_{25}	V_{tn}- 30%	0.4197
f_{26}	V_{tn} + 30%	0.4788
f_{27}	R7 - 50%	0.1500
f_{28}	R7 + 50%	0.0300
f_{29}	C2 - 50%	0.0270
f_{30}	C2 + 50%	0.0093

Figure 6.14 shows the test pattern generated for faults f_2 and f_{19} and the signatures at the primary output node 13. Faults f_2 and f_{19} produced similar error response *E(F)* profiles however, their magnitudes differ. This implies that f_2 and f_{19} produced an error response with similar frequency components but different magnitudes (the detectabilities are different). Therefore, the same test pattern is generated to detect both faults. From table 8 we see that fault f_{19} has the lowest detectability and fault f_2 has the highest detectability therefore indicating that fault f_2 is more detectable then fault f_{19}. This is reflected in their respective signatures. The signature difference between the good circuit and the circuit with fault f_2 is larger than the signature difference between the good circuit and the circuit with fault f_{19}.

For the 30 faults listed in table 5, LATTE would generate 30 test vectors, one for each fault. However, the efficient test set produced by the test set compaction algorithm requires only 7 test vectors or an 76.67% reduction. This is expected if we study the target fault list (table 5) more closely. Each of the 30 faults may be separated into four categories: (i) op-amp input terminals shorted together, (ii) gate and source terminals of transistor M9 are shorted inside the op-amp, (iii) output stage disconnected (transistor M17's drain terminal open), (iv) bias voltage disconnected (transistor M4's gate is open), (v) 30%

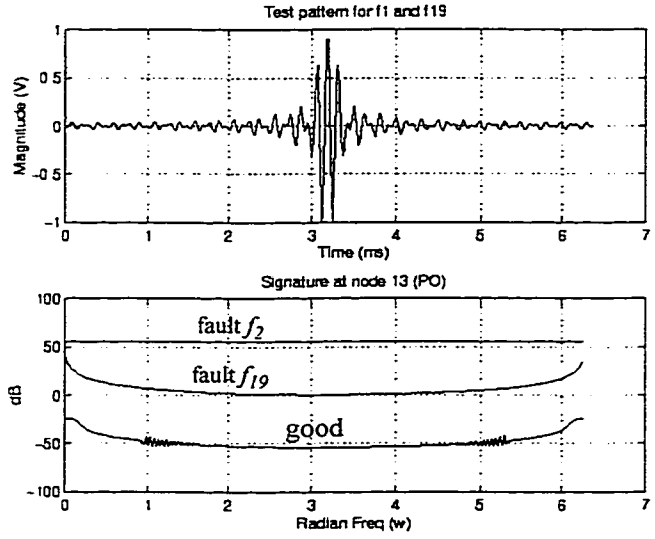


Figure 6.14: Test pattern and signatures for f_2 and f_{19}

variations in the value for V_{tn}, (vi) 50% variation in the resistance value of R7, and (vii) 50% variation in the capacitance value of C2. Instead of 30 vectors, only 7 vectors (one for each category) are required to test for these 7 types of faults.

If we study the target fault list for the state variable filter (table 4), we see a similar pattern. The only difference is that the state variable filter requires 8 vectors instead four even though the same types of faults are studied. This can be explained by the circuit topology. Figure 6.7 shows the state variable filter which has the op-amp O1's positive terminal not connected to ground like op-amps O2 and O3. Therefore, the effects of f_1 are not similar to the effects of faults f_5 and f_9, which implies that a separate test vector is required to test for f_1. This was exactly what had happened as the efficient test set $\boldsymbol{T}$ has 8 vectors.

6.6.2 Automated DFT for analog circuits and systems

Avionics power supply

The Boeing Company provided an AC-to-DC converter power supply to be used in this research. A diagram of the power supply is shown in figure 6.15. Since the power supply was provided by the Boeing Company, a detailed schematic is not available to the public due to proprietary reasons. However, the circuit under test shown in figure 6.15 includes resistors, inductors, capacitors, transformers, opto-couplers, buffers, flip-flops, a pulse width modulator (PWM), power FETs, diodes and op-amps, *etc.*

Since test points add area overhead and performance degradation, it is important to minimize the number of test points added to the circuit. Using REIGN, four test points were added to the circuit to increase its testability. Table 6 shows the improvement in controllability, observability and testability as a result of the added test points.

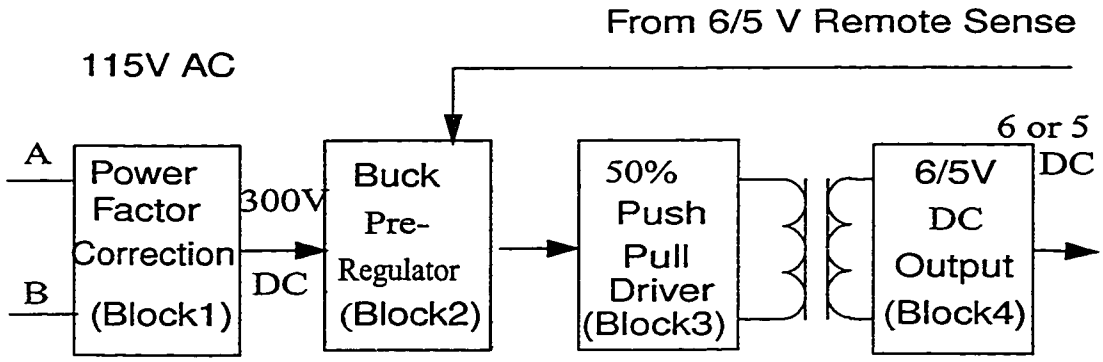


Figure 6.15: Power Supply

TABLE 6. Testability improvement

	0 test points	4 test points	improvement
controllability	0.3319	0.4571	36%
observability	0.4830	0.5888	20%
testability	0.3370	0.4995	48%

To validate our approach, the original power supply designer was asked to select a set of test points (4) without any knowledge of our selection nor our approach. The designer selected test points intuitively based on his knowledge of the design and testing. Three out of the four points selected by our technique were included in the set of points selected by the designer. This showed that our automated technique can produce results which are comparable with the results produced by an experienced designer with intimate knowledge of the circuit.

4-bit successive-approximation ADC

A 4-bit successive-approximation ADC is analyzed as another example. Its basic components are gates, D-flip-flops, CMOS switches, resistors, a comparator and an op-amp. Figure 6.16 is the diagram of this ADC. The *Digital Encoder* is mainly composed by

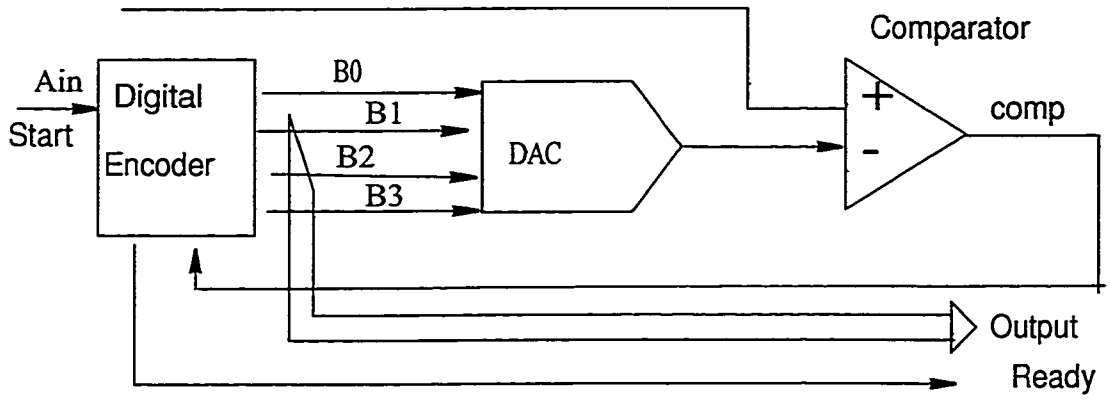


Figure 6.16: 4-bit successive approximation ADC

two groups of D-flip-flops. The *DAC* structure is a reduced-resistance-ratio ladder with an op-amp. A_{in} is the input analog voltage, *B0 to B3* are the outputs from MSB to LSB.

TABLE 7. Test Points and testability improvement

	before	after	improvement
4 control points controllability	0.506	0.732	31%
4 observation observability	0.125	0.541	331%
4 test points testability	0.181	0.439	142%

Three groups of test points are selected corresponding to the improvement in controllability, observability and testability. The results are shown in Table 7. *B0 to B3* control the DAC directly, an increase in their controllability will improve the overall controllability. The four control points are connected with the data and clock inputs of the output D-flip-flop group. They can increase the overall controllability by increasing the controllability of *B0 to B3*. As outputs, *B0 to B3* are totally observable. Because they are directly connected with the resistor-ladder, the observability of resistor-ladder in *DAC* is relatively high. Since the TTF of D-flip-flop is relatively low, the three observation points were inserted between the two D-flip-flop groups to improve the observability of digital part. To further improve the overall observability, the clock, a global control signal, is selected as the fourth observation point.

As described in these two examples, testability can be increased effectively by the

test point insertion procedure presented. These test points can help circuit designers obtain important information to test their circuits.

6.7 SYSTEM PERFORMANCE

This section analyzes the memory and run time requirements for REIGN. The largest usage of memory in REIGN is the connectivity matrix and the solution vector. For N nodes in a circuit, the memory requirement is $O(N^2)$.

The LU decomposition technique is used as the matrix solver. It can be shown that the run time of the LU decomposition technique is

$$Z = O\left(\frac{N^3}{3} + N^2 - \frac{N}{3}\right) \tag{6.1}$$

As N approaches infinity, equation (6.1) is reduced to just the N^3 term, equivalently

$$Z = O(N^3) \tag{6.2}$$

However, REIGN can be ran in three separate modes (testability analysis mode, test point selection and test generation) and each mode has a different run time. For the analysis mode, the run time is $O(SZ)$ where S is the number of steps in the analysis and Z is one LU decomposition as defined by equation (6.2). S depends on the range of values under consideration for analysis and is user defined.

In the test point selection mode, the run time (R_{tp}) is

$$R_{tp} = O(MNZ - MZ + Z) \tag{6.3}$$

Where M is the total number of test points to insert, N is the total number of nodes in the circuit and Z is equation (6.2). In the worst case, for each test point to insert, REIGN has to do an exhaustive search of every node in the circuit to find the optimum node to insert and this requires NZ LU decompositions. In the worst case where each test point is inserted iteratively (one at a time), for M points to insert, the inner loop is repeated M times by the

outer loop and the required run time is *MNZ*. However, once a node has been selected for insertion, that node is no longer included in the search and therefore the savings in run time is *MZ*. The final factor *Z* is due to the computation required to calculate the initial baseline (the objective functions evaluated without any test points selected. Substituting equation (6.2) into (6.3) and noting that as *N* approaches infinity, equation (6.3) reduces to

$$R_{tp} = O(MN^4) \tag{6.4}$$

For automatic test generation, we can break down the run time into three separate terms: the time required to calculate T_g (the good profile), the time required to calculate T_{fi} (profile in the presence of fault *fi*), and the time required to calculate D_{fi} and generate v_{fi}. The time required to calculate T_g is the same time required for analysis and is therefore *O(SZ)*. The time required to calculate T_{fi} is the same for T_g, but for n_f faults in the fault list, LATTE must calculate n_f T_{fi}'s. That results in a run time of $O(n_fSZ)$. Since calculating D_{fi} and generating v_{fi} does not directly depend on the size of the circuit, we can assign it have a run time of *K*. The complete run time (R_{tg}) is then

$$R_{tg} = O(SZ + n_fSZ + K) \tag{6.5}$$

From equation (6.5), note that the run time for test generation is determined by three factors: range of frequency for analysis, size of fault list and the time required to compute the inverse transform and convolution operations. The range of frequency for analysis determines *S* and that is defined by the user. The size of the fault list, n_f, is also user defined. *K* is the third factor which is the sum of the time required to calculate D_{fi} and perform the inverse transform and convolution operations. The time required to calculate D_{fi} is small compared to the time required for the inverse transform operation. Further, the convolution operation is performed with the delta function which is also small compared with the inverse transform operation. Therefore *K* is mainly determined by the inverse transform operation.

The inverse transform operation is implemented with the IFFT (Inverse Fast Fourier Transform) algorithm. For a P point IFFT, the run time for the IFFT algorithm is [52]:

$$K = O(P \bullet LOG_2 P) \tag{6.6}$$

Where P is the number of points in the IFFT. In REIGN, P is set to 512, however, it is also user define. Therefore, a user can select different P for various different applications.

Equation (6.5) clearly shows the accuracy versus speed trade-off as it shows that the run time is a user defined triplet variable (S, n_f, P). Automatic test pattern generation may be sped up using several different methods. First, S can be reduced by restricting the range of frequencies of interest. However, by restricting the range of frequency of interest, the test pattern generated may not contain all the necessary frequency components required to detect the faults specified in the fault list. Second, the fault list may be reduced (reduce n_f). However, with a smaller fault list, we run the danger of having a high escape rate. That is faults will escape detection. Finally, K may be reduced through the reduction of P. Again, this leads to less accurate test patterns.

6.8 SUMMARY

This chapter discussed the CAD implementation of the algorithms and methodologies presented in this dissertation. The result is a CAD software package called REIGN which is suite of tools for automated DFT and ATPG applications for mixed-signal and analog circuits. REIGN was applied to various benchmark circuits and circuits from the industry and results are very promising.

In a DFT case study involving an AC-to-DC converter (power supply) designed by the Boeing Company, REIGN produced results which are comparable if not identical to those of an experience designer. In particular, REIGN selected four nodes as test points and independently, the power supply designer was asked to do the same. Three our of the four nodes selected by REIGN were identical to those selected by the designer. With the four test points selected by REIGN, the testability of the power supply was improved by

48%. In a separate case study focused on a 4-bit successive approximation analog-to-digital converter, with four test points inserted, the testability metric was shown to be improved by 142% and observability was improved by over 330%.

These results are significant in light of the electronic industries high staff turn over rate. In the Boeing power supply case study, we showed that REIGN produced results which are comparable to those of an experience designer. This implies that experience is not a necessity in terms of DFT applications. Given a design, an entry level engineer may apply REIGN and produce results which are comparable to an experience designer. Another advantage of REIGN is for large systems, REIGN provides a systematic approach for DFT thus eliminating human errors. Even the most experience of designers might make inadvertent errors, however REIGN does not.

Other then DFT, REIGN can also automatically generate test patterns to detect parametric and catastrophic faults. We showed that the generated test set can be reduced through fault simulation. Results showed that 76.67% reduction is achievable. Smaller test sets implies shorter test times and length of test times has been identified as a major bottleneck in the field of mixed-signal and analog circuit testing. The presented algorithm shows promise as a realistic possible solution to solving this major bottleneck.

Chapter 7
Concluding Remarks

There are two primary factors motivating the development of advances in mixed-signal integrated circuits. First, tremendous cost advantages can be obtained by combining analog and digital functions in single-substrate manufacturing processes. Second, overall performance is generally enhanced by increasing the levels of circuit density, and reducing the frequency of off-chip data transfer. Unfortunately, mixed analog/digital circuit designers and test engineers are faced with challenging issues that have been largely irrelevant during the evolution of purely digital VLSI. Historically, design and test activities are done by separate groups of engineers at different phases of the design flow. Recent developments in mixed-signal circuit designs especially with high speed digital and high precision analog subsystems on the chip have made this separation an economic disadvantage.

As reviewed in chapter 1, it is imperative that design and test activities be done simultaneously. Test program development must be done with efficiency and the developed test must be of high quality to ensure a short time to market. With a short time to market, the product has a much higher probability of being an economic success. Our discussion has summarized research efforts aimed at aiding in test program development and

test pattern generation.

Test program development is made easier through the application of DFT and automatic test pattern generation methodologies. Testability analysis based on a set of novel testability models is the approach used to alleviate much of the difficulties previously encountered in these areas. Test points are optimally placed for DFT purposes. A new multi-frequency algorithm was developed to generated test patterns automatically. In addition, the algorithms and methodologies have been implemented in a stand alone testability analysis system called REIGN.

The techniques employed in REIGN have been developed specifically to surpass the capabilities of previously adopted testability strategies. At the same time, it would be incorrect to assert that this approach offers a complete solution to mixed-signal circuit testability. It seems reasonable at this point to present a forthright discussion on the limitations of the adopted methodology.

Although having a highly controllable and observable circuit does not guarantee that the circuit satisfies its specifications, it does help with test and diagnosis. Having access to internal nodes of a complex circuit may help in identifying problems and its causes much quicker. It also provides alternative ports for test vectors and allows for more flexibility during test.

Another consideration concerns the test set compaction routine, ESPRESSO. The optimal test set design algorithm described in chapter 6 requires that a fault simulation be performed for every fault and test in the generated test set. As of this writing, there were no widely available efficient analog fault simulator. We were forced, therefore, to use serial fault simulation techniques utilizing a circuit simulator (HSPICE). That is, each fault was modeled with an equivalent circuit model and serially simulated in HSPICE. This was an excruciatingly slow process. For circuits with a large fault list, this method is not very practical. If a concurrent fault simulator, such as CONCERT [26], [27], is avail-

able, this process can be significantly sped up.

The current TTF formulation does not include signal phase. Neglecting signal phase may have several interesting consequences. The edge weight on all the edges in the testability graph will always be positive. Consider two signals, one which is the inverse of the other. If the two copies are connected to a common node, the signals will both contribute positively to the testability measure, though in the real circuit, the signals will cancel each other completely. However, the TTF of a component, as it is currently formulated, is an indication of how well signals can propagate through a component. It is true that two signals, one the inverse of the other, connected to a common node will contribute positively to the testability measure when in reality, the signals cancel each other. That is because it is just as easy (or difficult) to propagate a signal with a certain phase as it is to propagate the same signal but the exact opposite phase. The TTF characterizes the component not the signal. Thus, neglecting phase may produce more optimistic results. As a direction for future research, one has to consider the effects of signal phase when formulating TTFs for components.

The current test point selection algorithm neglects the CUT's performance degradation due to the addition of test points into the CUT. The optimization is performed only on the objective functions discussed in chapter 5. However, a constraint based on performance degradation may be included in the optimization to further enhance the SEATTLE algorithm. Such a constraint may guarantee that testability is maximized while minimizing the performance degradation of the CUT due to the addition of test points.

The severity of the fault simulation bottleneck could also be attenuated by incorporating a form of concurrent fault simulation technique. A future version of REIGN might have a concurrent fault simulator as part of the system. The motivation here is to reduce the amount of time spent on fault simulation required to generate an optimal test set. With this addition, REIGN can provide an integrated test program development solution from testability analysis to test generation and finally to fault simulation.

One drawback of the test vectors generated by the algorithm presented is that the vectors lack structure form. Hence, the vectors are rather difficult to generate and would require a high performance arbitrary waveform generator which may potentially be very expensive. Therefore, it would be more cost effective to generate test vectors with structural form.

References

[1] M. Abramovici, M. A. Breuer, and A. D. Friedman, **Digital Systems Testing and Testable Design**, IEEE Press, Revised Printing, 1990, ISBN 0-7803-1062-4.

[2] V. D. Agrawal, K. T. Cheng, D. D. Johnson and T. Lin, "A Complete Solution to the Partial Scan Problem," *Proceedings of the International Test Conference*, pp. 44 - 51, 1987.

[3] A. P. Ambler, M. Abadir, S. Sastry, Ellis Horwood, Editors, **Economics of Design and Test for Electronic Circuits and Systems**, 1992, ISBN 0-13-224767-4.

[4] F. Azais, A. Ivanov, M. Renovell and Y. Bertrand, "A Methodology and Design for Efficient Testing of Voltage-Controlled Oscillators (VCOs)," *Proceedings of the Asian Test Symposium*, pp. 383 - 387, 1998.

[5] P. Bannerjee and J. A. Abraham, "Characterization and Testing of Physical Failures in MOS Logic Circuits," *IEEE Design & Test of Computers*, Vol. 1, No. 4, pp. 76 - 86, 1984.

[6] R. G. Bennets, C. M. Maunder, and G. D. Robinson, "CAMELOT: A Computer-Aided MEasure for LOgic Testability," *Proceedings of the IEEE International Conference on Circuits and Computers*, pp. 1162 - 1165, 1980.

[7] W. C. Berg and R. D. Hess, "COMET: A Testability Analysis and Design Modification Package," *IEEE Test Conference*, pp. 364 - 378, 1982.

[8] F. R. Bergseth and S. S. Venkata, **Introduction to Electric Energy Devices**, Prentice Hall, Englewood Cliffs, New Jersey, 1987, 0-13-481383-9.

[9] H. Bogard and C. Repasky, "Increasing Test Throughput Through the Implementation of Parallel Test on a 16-bit Multimedia Audio CODEC," *Proceedings of the International Test Conference*, pp. 370 - 376, 1995.

[10] C. H. Chen and C. Wu and D. G. Saab, "BETA: Behavioral Testability Analysis," *IEEE International Conference on Computer-Aided Design, Digest of Technical Paper*, pp. 202 - 205.

[11] S. DasGupta, P. Goel, R. G. Walther, and T. W. Williams, "A Variation of LSSD and Its Implications on Design and Test Pattern Generation in VLSI," *Proceedings of the Test Conference*, pp. 63 - 66, 1982.

[12] G. Devarayanadurg, P. Goteti, and M. Soma, "Hierarchy based Statistical Fault Simulation of Mixed-Signal ICs," *Proceedings of the International Test Conference*, pp. 521 - 527, 1996.

[13] G. Devarayanadurg and M. Soma, "Dynamic Test Signal Design for Analog ICs," *Proceedings of the International Conference on Computer-Aided Design*, pp. 627 - 629, 1995.

[14] G. Devarayanadurg and M. Soma, "Analytical Fault Modeling and Static Test Generation for Analog ICs," *Proceedings of the International Conference on Computer-Aided Design*, pp. 44 - 47, 1994.

[15] S. Franco, **Design with Operational Amplifiers and Analog Integrated Circuits**, McGraw-Hill, Chapter 6, "Stability and Frequency Compensation," 1988, 0-07-021799-8.

[16] G. Gielen, Z. Wang, and W. Sansen, "Fault Detection and Input Stimulus Determination for the Testing of Analog Integrated Circuits Based on Power Supply Current Monitoring," *Proceedings of the International Conference on Computer-Aided Design*, pp. 495 - 498, 1994.

[17] D. K. Goel and R. M. McDermontt, "An Interactive Testability Analysis Program - ITTAP," *Proceedings of the 19th IEEE Design Automation Conference*, pp. 581 - 586, 1982.

[18] L. H. Goldstein and E. L. Thigpen, "SCOAP: Sandia Controllability/ Observability Analysis Program," *Proceedings of the 17th Design Automation Conference*, pp. 190 - 196, 1980.

[19] L. H. Goldstein, "Controllability/Observability Analysis of Digital Circuits," *IEEE Transactions on Circuits and Systems*, Vol. CAS-26, pp. 685 - 693, 1979.

[20] P. Goteti, G. Devarayanadurg, and M. Soma, "DFT for Embedded Charge-Pump PLL Systems Incorporating IEEE 1149.1," *Proceedings of the Custom Integrated Circuits Conference*, pp. 210 - 213, 1997.

[21] J. Grason, "TMEAS, A Testability Measurement Program," *Proceedings of the 16th IEEE Design Automation Conference*, pp. 156 - 161, 1979.

[22] N. B. Hamida, K. Saab, D. Marche, B. Kaminska, G. Quesnel, "LIMSoft: Automated tool for design and test integration of analog circuits," *Proceedings of the International Test Conference*, pp. 571 - 580, 1996.

[23] N. Ben Hamida and B. Kaminska, "Analog circuit testing based on sensitivity computation and new circuit modeling," *Proceedings of the IEEE International Test Conference*, 1993.

[24] G. J. Hemink, B. W. Meijer, H. G. Kerkhoff, "Testability Analysis of Analog Systems," *IEEE Transactions on Computer-Aided Design*, Vol. 9, pp. 573 - 583, 1990.

[25] G. J. Hemink, B. W. Meijer, H. G. Kerhoff, "TASTE: A Tool for Analog System Testability Evaluation," *Proceedings of the International Test Conference*, pp. 829 - 838, 1988.

[26] J. Hou and A. Chatterjee, "CONCERT: A concurrent Transient Fault Simulator for Non-Linear Analog Circuits," *Proceedings of the International Conference on Computer-Aided Design*, pp. 384 - 391, 1998.

[27] J. Hou and A. Chatterjee, "CONCERT: A concurrent Fault Simulator for Analog Circuits," *Proceedings of the International Mixed-Signal Test Workshop*, pp. 3 - 8,

1998.

[28] J. B. Hughes, I. C. Macbeth, and D. M. Pattullo, "Second-Generation Switched-Current Signal Processing," *Proceedings of the IEEE International Symposium on Circuits and Systems*, 1990.

[29] S. D. Huss and R. S. Gyurcsik, "Optimal Ordering of Analog Integrated Circuits to minimize Test Time," *Proceedings of the Design Automation Conference*, pp. 494 - 499, 1991.

[30] **S. D. Huynh**, J. Zhang, S. Kim, G. Devarayanadurg, and M. Soma, "Hierarchical ATPG for Analog Circuits and Systems," *International Mixed-Signal Test Workshop*, 1999.

[31] **S. D. Huynh**, S. Kim, M. Soma, and J. Zhang, "Automatic Analog Test Signal Generation Using Multi-Frequency Analysis," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 46, No. 5, pp. 565 - 577, 1999.

[32] **S. D. Huynh**, S. Kim, M. Soma, and J. Zhang, "Dynamic Test Set Generation for Analog Circuits and Systems," *Proceedings of the Asian Test Symposium 98*, pp. 360 - 365, 1998.

[33] **S. D. Huynh**, S. Kim, M. Soma, and J. Zhang, "Testability Analysis and Multi-Frequency ATPG for Analog Circuits and Systems," *Proceedings of the International Conference on Computer-Aided Design*, pp. 376 - 383, 1998.

[34] **S. D. Huynh**, M. Soma, and J. Zhang, "Analytical Testability Models for Mixed-Signal Circuits," *1998 IEEE International Workshop on System Test and Diagnosis, Digest of Papers*, pp. 43 - 50, 1998.

[35] **S. D. Huynh**, A. S. Tea, and M. Soma, "Access Structures and Circuits to Improve Mixed-Signal Testability," *TECHCON 96*, Semiconductor Research Corporation, Z96012, Task ID 552.001, 1996.

[36] M. Ismail and T. Fiez, **Analog VLSI Signal and Information Processing**, McGraw-Hill, Chapter 12, "Design for Testability," 1994, ISBN 0-07-032386-0.

[37] G. Iuculano *et. al.*, "Multifrequency measurement of testability with application to

large linear analog systems," *IEEE Transactions on Circuits and Systems*, Vol. CAS-33, pp. 644 - 648, 1986.

[38] J. E. Jagodnik and M. V. Wolfson, "Systematic fault simulation in an analog circuit simulator," *IEEE Transactions on Circuits and Systems*, Vol. CAS-26, No. 7, pp. 549 - 554, 1979.

[39] B. Kaminska, K. Arabi, I. Bell, P. Goteti, J. L. Huertas, B. Kim, A. Rueda, and M. Soma, "Analog and Mixed-Signal Benchmark Circuits - First Release," *Proceedings of the International Test Conference*, pp. 183 - 190, 1997.

[40] M. Katoozi, H. Kutz, M. Soma, and **S. Huynh**, "Strategies and Structures for Test Access in Mixed-Signal MCMs," *Proceedings of the IEEE MCM Conference*, pp. 144 - 149, 1997.

[41] P. G. Kovijanic, "Single Testability Figure of Merit," *IEEE Test Conference*, pp. 521 - 529, 1981.

[42] P. G. Kovijanic, "Computer-Aided Testability Analysis," *AUTOTESTCON*, pp. 292 - 294, 1979.

[43] W. M. Lindermeir, H. E. Graeb, and K. J. Antreich, "Design based Analog Testing by Characteristic Observation Inference," *Proceedings of the International Conference on Computer-Aided Design*, pp. 620 - 626, 1995.

[44] E. Liu, W. Kao, E. Felt, and A. Sangiovanni-Vincentelli, "Analog testability analysis and fault diagnosis using behavioral modeling," *IEEE Custom Integrated Circuit Conference*, pp. 413 - 416, 1994.

[45] H. K. T. Ma, S. Devadas, A. R. Newton, and A. Sangiovanni-Vincentelli, "An Incomplete Scan Design Approach to Test Generation for Sequential Machines," *Proceedings of the International Test Conference*, pp. 730 - 734, 1988.

[46] W. Maly, "Realistic fault modeling for VLSI testing," *Proceedings of the IEEE Design Automation Conference*, pp. 173 - 180, 1987.

[47] M. Meixner and W. Maly, "Fault Modeling for the Testing of Mixed Integrated Circuits," *Proceedings of the International Test Conference*, pp. 564 - 572, 1991.

[48] META-SOFTWARE, **HSPICE User's Manual**, Vol. 2: Elements and Models.

[49] L. Milor and A. L. Sangiovanni-Vincentelli, "Minimizing Production Test Time to Detect Faults in Analog Circuits," *IEEE Transactions on Computer-Aided Design*, Vol. 13, No. 6, 1994.

[50] L. Milor and V. Visvanathan, "Detection of Catastrophic Faults in Analog Integrated Circuits," *IEEE Transactions on Computer-Aided Design*, Vol. CAD-8, No. 2, pp. 114 - 130, 1989.

[51] N. Nagi, A. Chatterjee, A. Balivada and J. A. Abraham, "Fault-based Automatic Test Generator for Linear Analog Circuits," *Proceedings of the International Conference on Computer-Aided Design*, pp. 88 - 91, 1993.

[52] A. V. Oppenheim and R. W. Schafer, **Discrete-time Signal Processing**, Prentice-Hall, 1989, ISBN 0-13-216292-X.

[53] C. Y. Pan and K. T. Cheng, "Test generation for analog LTI systems," *International Mixed-Signal Test Workshop*, pp. 93 - 100, 1997.

[54] A. Pahwa and R. A. Rohrer, "Band Faults: Efficient approximations to fault bands for the simulation before diagnosis of Linear Circuits," *IEEE Transactions on Circuits and Systems*, Vol. CAS-29, pp. 81 - 88, 1982.

[55] J. Partridge, "Testing for bipolar integrated circuit failure modes," *Digest of papers, 1980 Test Conference*, pp. 397 406, 1980.

[56] W. H. Press, S. A. Teukolskey, W. T. Vetterling, B. P. Flannery, **Numerical Recipes in C: The Art of Scientific Computing**, Second Edition, Cambridge University Press, 1992, ISBN 0-521-43108-5.

[57] **Proposed IEEE Standard for a Mixed-Signal Test Bus - D20**, IEEE P1149.4, IEEE Standards Office, 1998.

[58] I. M. Ratiu, A. Sangiovani-Vencentelli, and D. O. Pederson, "VICTOR: A fast VLSI testability analysis program," *IEEE Test Conference*, pp. 397 - 401, 1982.

[59] G. W. Roberts, "Re-Examining the Needs of the Mixed-Signal Test Community," *Proceedings of the International Test Conference*, pp. 298, 1995.

[60] N. Sen and R. Saeks, "Fault Diagnosis for Linear Systems Via Multifrequency Measurements," *IEEE Transactions on Circuits and Systems*, Vol. CAS-26, No. 7,

pp. 457 - 465, 1979.

[61] N. Sen and R. Saeks, "A measure of testability and its application to test point selection - Theory," *20th IEEE Midwest Symposium on Circuits and System*, 1977.

[62] N. Sen and R. Saeks, "A measure of testability and its applications to test point selection - Computation," *IEEE AUTOTESTCON*, 1977.

[63] J. P. Shen, W. Maly, and F. J. Ferguson, "Inductive Fault Analysis of MOS Integrated Circuits," *IEEE Design & Test of Computers*, Vol. 2, No. 6, pp. 13 - 26, 1985.

[64] G. Singer, "VTS 97 Keynote: The Future of Test and DFT," *IEEE Design & Test of Computers*, pp. 11 - 14, 1997.

[65] M. Slamani and B. Kaminska, "Fault observability analysis of analog circuits in frequency domain," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 43, pp. 134 - 139, 1996.

[66] M. Slamani, B. Kaminska, and G. Quesnel, "An Integrated Approach for Analog Circuit Testing with Minimum Number of Detected Parameters," *Proceedings of the International Test Conference*, pp. 631 - 640, 1994.

[67] M. Slamani and B. Kaminska, "Multifrequency Testability Analysis for Analog Circuit," *Proceedings of the IEEE VLSI Test Symposium*, pp. 54 - 59, 1994.

[68] M. Slamani and B. Kaminska, "Analog circuit fault diagnosis based on sensitivity computation and functional testing," *IEEE Design & Test of Computers*, pp. 30 - 39, 1992.

[69] M. Slamani and B. Kaminska, "Testing analog circuits by sensitivity computation," *Proceedings of the IEEE European Design Automation Conference*, pp. 532 - 537, 1992.

[70] M. Soma, "Challenges in Analog and Mixed-Signal Fault Models," *Circuits & Devices*, pp. 16 - 19, 1996.

[71] M. Soma, "Structures and Concepts for Current-based Analog Scan," *Proceedings of the IEEE Custom Integrated Circuit Conference*, pp. 517 - 520, 1995.

[72] M. Soma, "A Design for test methodology for active analog filters," *International*

Test Conference, pp. 183 - 192, 1990.

[73] M. Soma, T. M. Bocek, T. D. Vu, and J. D. Moffatt, "Experimental Results for Current-based Analog Scan," *Proceedings of IEEE International Test Conference*, 1997.

[74] *Standard 1149.1 1993*, "A Standard Test Access Port and Boundary Scan Architecture," IEEE 1993.

[75] G. N. Stenbakken, T. M. Souders, and G. W. Stewart, "Ambiguity Groups and testability," *IEEE Transactions on Instrumentation and Measurements*, Vol. 38, pp. 941 - 947, 1989.

[76] G. N. Stenbakken and T. M. Souders, "Test-point selection and testability measures via QR factorization of linear models," *IEEE Transactions on Instrumentation and Measurements*, Vol. IM-36, pp. 406 - 410, 1987.

[77] J. E. Stephenson and J. Grason," A Testability Measure for Register Transfer Level Digital Circuits," *IEEE Fault-Tolerant Computing*, pp. 101 - 107, 1976.

[78] M. W. Tian and C. J. R. Shi, "Rapid Frequency-Domain Analog Fault Simulation Under Parametric Tolerances," *Proceedings of the Design Automation Conference*, pp. 275 - 280, 1997.

[79] C. Timoc, M. Beuhler, T. Griswold, C. Pina, F. Scott, and L. Hess, "Logical Models of Physical Failures," *Proceedings of the International Test Conference*, pp. 546 - 553, 1983.

[80] J. Turino, "Test Economics in the 21st Century," *IEEE Design & Test of Computers*, pp. 41 - 44, 1997.

[81] Shen-Jen Tsai, "Test Vector Generation for Linear Analog Devices," *Proceedings of the International Test Conference*, pp. 592 - 597, 1991.

[82] W. Verhaegen, G. V. D. Plas, and G. Gielen, "Automated Test Pattern Generation for Analog Integrated Circuits," *Proceedings of the 15th IEEE VLSI Test Symposium*, pp. 296 - 301, 1997.

[83] V. Visvanathan and A. Sangiovanni-Vincentelli, "A computational approach for the diagnosability of circuits," *IEEE Transactions on Computer-Aided Design*, Vol.

CAD-3, pp. 165 - 171, 1984.

[84] R. Voorakaranam, S. Chakrabarti, J. Hou, A. Gomes, S. Cherubal and A. Chatterjee, "Hierarchical Specification-Driven Analog Fault Modeling for Efficient Fault Simulation and Diagnosis," *Proceedings of the International Test Conference*, pp. 903 - 912, 1997.

[85] J. Zhang, **S. D. Huynh**, and M. Soma, "A Test Point Insertion Algorithm of Mixed-Signal Circuits," To be published and presented at *VLSI Test Symposium 99*.

[86] M. Zwolinski, A. D. Brown, C. D. Chalk, "Concurrent Analogue Fault Simulation," *Proceedings of the International Mixed-Signal Testing Workshop*, pp. 42 - 47, 1997.

Appendix
TTF derivations MOSFETs

In this appendix, we will derive the transconductance and channel conductance of MOSFETs. The transconductance and channel conductance has different values depending on the region of operation for the MOSFET.

1. Cutoff: $V_{GS} < V_T$

   In the cutoff region, the MOSFET is basically off and there is only the leakage current from drain to source. Therefore, g_m is zero and g_{DS} is proportional to the leakage current.

   $$g_m = 0 \qquad \text{(A.1)}$$

   $$g_{DS} \propto I_{leakage} \qquad \text{(A.2)}$$

2. Linear: $V_{DS} < V_{GS} - V_T$

   In this region, the drain to source current is:

$$i_{DS} = \frac{KP}{2}\left(\frac{W}{L}\right)\left[2(v_{GS}-V_T)v_{DS}-v_{DS}^2\right] \quad \text{(A.3)}$$

Applying the respective derivatives (equations 4.16 and 4.17), the following results.

$$g_m = KP\left(\frac{W}{L}\right)v_{DS} \quad \text{(A.4)}$$

$$g_{DS} = KP\left(\frac{W}{L}\right)(v_{GS}-V_T-v_{DS}) \quad \text{(A.5)}$$

3. Saturation: $V_{DS} \geq V_{GS}-V_T$

In the saturation region, the drain to source current is:

$$i_{DS} = \frac{KP}{2}\left(\frac{W}{L}\right)(v_{GS}-V_T)^2 \quad \text{(A.6)}$$

Applying the respective derivatives (equations 4.16 and 4.17), the following results.

$$g_m = KP\left(\frac{W}{L}\right)(v_{GS}-V_T) \quad \text{(A.7)}$$

$$g_{DS} = \frac{\lambda KP}{2}\left(\frac{W}{L}\right)(v_{GS}-V_T) \quad \text{(A.8)}$$

Table 8 shows the list of parameters and their definitions.

Table 8: Parameters for MOSFET

Parameters	Units	Definitions
$I_{leakage}$	A	Leakage current when the MOSFET is off
λ	1/V	Channel modulation
KP	A/V^2	Transconductance

Table 8: Parameters for MOSFET

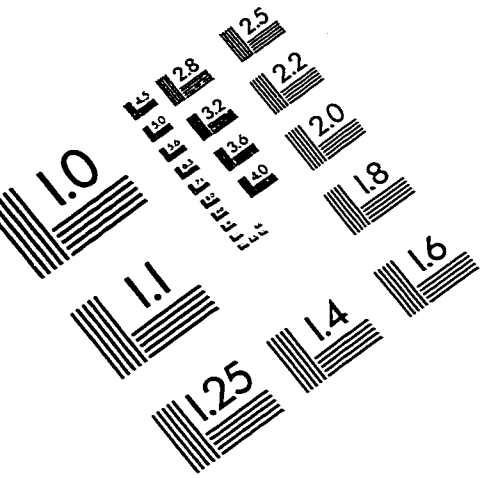
Parameters	Units	Definitions
W,L	m	Channel width and channel length
V_T	V	Threshold voltage
v_{GS}, v_{DS}	V	Gate to Source and Drain to Source voltage

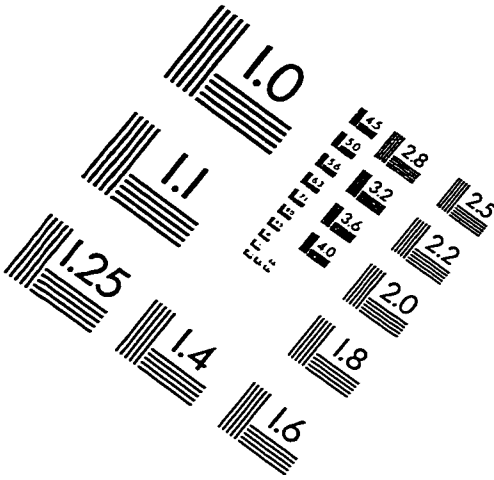
Vita

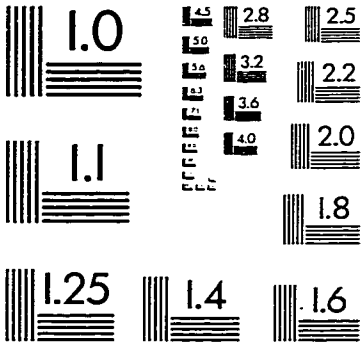
Sam D. Huynh was born in Ho Chi Minh City (formerly Saigon), Vietnam on June 20, 1972. He received the B.S.E.E. and M.S.E.E. degrees from the University of Washington, Seattle, Washington in 1994 and 1996 respectively. During the summers from 1991 through 1996, he interned at IBM (Rochester, '91), Siemens ROLM (Santa Clara, '93 and '94), Advanced Micro Devices (Austin, '95) and Intel (Portland, '96). At the end of 1996 and beginning of 1997, he held a temporary position with Cascade Design Automation, Seattle, Washington. During the academic years from 1996 through 1998, he taught electronics classes for North Seattle Community College and Western Washington University Extension. From April to June 1998, he was an adjunct lecturer at Seattle University, Seattle, Washington.

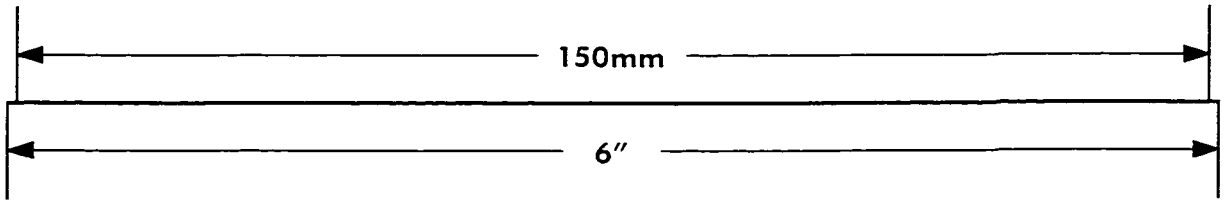
He is currently completing Ph.D. degree requirements in the Department of Electrical Engineering at the University of Washington, Seattle, Washington. His graduate research has focused primarily on testability modeling and analysis methodologies for mixed analog/digital ICs. Additional research interests include design-for-test methodologies, mixed analog/digital circuit and systems design, and automatic test pattern generation for mixed analog/digital VLSI. He has accepted a position with Silicon Graphics, Inc., Mountain View, California, in the High Performance Graphics Group.

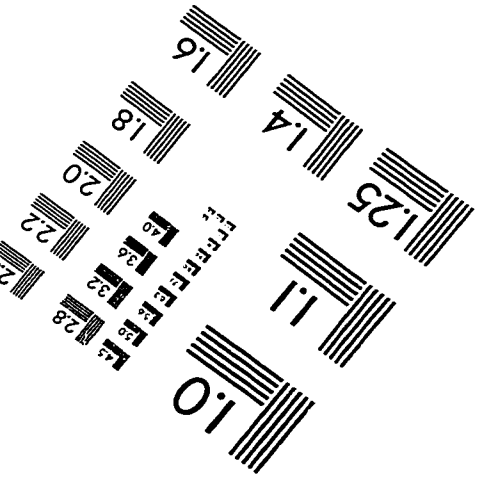
IMAGE EVALUATION
TEST TARGET (QA-3)

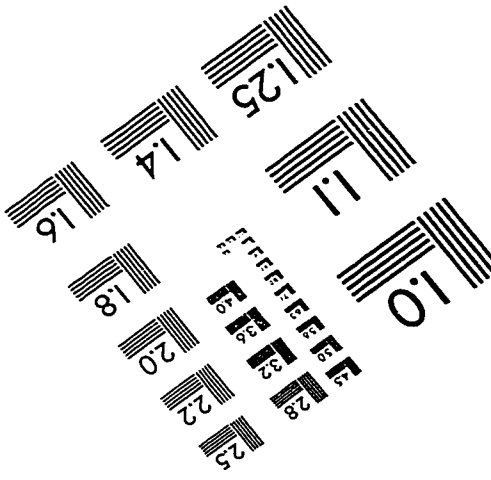












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989