PERFORMANCE MEASURES OF DATA COMPRESSION TECHNIQUES IN FAULT DETECTION

by

ROBERT CAMPBELL AITKEN
B.Sc., University of Victoria, 1985

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in the Department
of
Computer Science

ACCEPTED
FACULTY OF GRADUATE STUDIES
[redacted]
DEAN
DATE May 20, 86

We accept this thesis as conforming
to the required standard

[redacted]
Dr. Jon C. Muzio

[redacted]
Dr. Frank Ruskey

[redacted]
Dr. George A. Beer

[redacted]
Dr. Fayez H. El Guibaly

University of Victoria

April 1986

Supervisor: Professor Jon C. Muzio

ABSTRACT

The testing of digital circuits comprises a significant and growing percentage of their total production cost. Data compression is a set of testing techniques which can be shown to substantially reduce the expense of fault detection. Further, data compression is amenable to design for testability methodologies, as well as built-in self-test designs.

New performance measures are derived for two major data compression schemes: LFSR signature analysis and testing based on Rademacher-Walsh spectral coefficients. Results are based on both a standard theoretical framework and empirical values from a set of sample circuits. The empirical evidence obtained allows analysis of both the techniques and the validity of the theoretical framework.

The relative merits of various methods are given, based on the theoretical and empirical evidence presented. It is shown that those which include the syndrome of a function are particularly useful because of the deterministic bounds on fault coverage which can be obtained for them.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank Jon Muzio for his invaluable help and guidance during the preparation of this thesis. In addition, I am indebted to George Beer for his assistance throughout my university career, Frank Ruskey for his help with the combinatorial portion of this work, as well as NSERC for funding my studies. Finally, I would like to thank Denise Maloney for her continued and much-appreciated support.

Chapter 1: Introduction

One of the most important and expensive tasks in digital circuit production is testing. This thesis investigates methods which can substantially reduce testing costs. Initially, data compression, which is a class of testing techniques, is defined and its methods reviewed. The major work which is presented herein is a detailed analysis, both theoretical and empirical, of the performance of these methods. Initially, we present an outline of the testing problem and introduce some of the terminology used.

Many digital circuits are situated entirely on a single silicon chip (integrated circuits). The testing problem assumes that a chip which has been manufactured correctly will implement the function specified by the circuit designer. This makes testing distinct from design verification, where the function implemented by a circuit and that intended by its designer are compared.

Because of the intricasies of the integrated circuit manufacturing process, many chips produced are faulty. Testing must be performed at manufacturing time in order to ensure that the number of faulty chips released for sale is minimized. Testing is also performed periodically during a circuit's lifetime in order to detect problems that have occurred during operation. This periodic testing allows faulty hardware to be detected and replaced. Entire boards can be tested in this fashion. Without some

form of checking, system reliability cannot be ensured.

All testing methods can be greatly assisted if the circuit is designed with testability in mind. Design for testability (DFT) is now an important methodology, and if a testing technique is to be successful it should be amenable to DFT.

In this work we are concerned with fault detection, as distinct from fault location: i.e., a circuit either passes or fails a test, and no attempt is made to locate the actual fault in a circuit which fails. This technique is often called "go/no-go" testing. Built-in self-testing (BIST), in which the tester is located on the chip, is an important example. In BIST, the circuit is given a signal to begin testing, and after a prescribed length of time it reports the success or failure of the test. Naturally, in this situation the tester must be able to test itself as well as the rest of the circuit.

As integrated circuits have increased in complexity, so has the problem of testing these circuits. With today's VLSI (very large scale integration) taxing the limits of classical testing schemes, it seems unlikely that these classical methods will continue to be practical as circuit density increases further. For this reason, testing methodologies such as data compression, design for testability, and built-in self-testing are being aggressively pursued. It is expected that they will substantially reduce the cost of testing chips, which at the present time represents a significant and

growing percentage of total circuit cost.

Throughout the thesis only combinational circuits are considered. These circuits have no internal memory, so their output is completely determined by their present inputs. When DFT techniques such as level-sensitive scan design (LSSD) (see [16] or [8] for more information) are used, sequential circuits can be thought of as combinational for test purposes. Without these DFT methods, testing of sequential circuitry is such a complex problem that some manufacturers, such as IBM, require them for all new circuit designs.

Chapter 2 provides background information about classical testing methods, and introduces the concept of fault models. It discusses various methods proposed in the literature to lower the expense (in both time and resources) of the classical test set techniques. The ideas examined are random test generation and data compression techniques, which include the pseudo-random shift register "signature analysis" and various counting methods. The counting methods are all based on the notion of Rademacher-Walsh spectral coefficients, which in turn are derived from the Hadamard matrix. In addition, the theoretical framework for comparing the effectiveness of testing schemes is presented, together with a review of some previous work in probabilistic analysis of testing methods.

Chapter 3 outlines the previous work done in terms of measuring the effectiveness of shift register signature analysis techniques. The results obtained are used later in comparing the performance of counting methods to that of the "standard" for data compression, the shift register. Both this chapter and the new results of chapter 4 are based on a particular assumption regarding the behaviour of faulty circuits. The validity of this assumption is examined in detail in chapter 5.

In chapter 4, the average and worst-case behaviour in terms of error coverage are derived for several counting methods. The work for syndrome testing is presented as a "standard". Effectiveness measures are developed for two other spectral-based techniques. The first of these, the use of the syndrome and an additional spectral coefficient, has been proposed by others but not analyzed in this way. The second method, the so-called 4-valued syndrome, is entirely new. A final section tabulates numerical results for the earlier formulae and makes comparisons between the various testing methods based on these results.

Chapter 5 includes two sections. In the first section an experiment is given to examine the predictions of the standard theoretical framework in a particular instance, and measure its validity. In addition, arguments are presented to support the claim that the empirical results are consistent with typical circuit behaviour. In the final section, empirical measures of

the performance of each of the testing methods introduced are given. The empirical values facilitate comparisons of the actual effectiveness of the assorted test methods and allow some statements to be made regarding the "best" method to use for a given circuit.

Finally, some concluding remarks are given and areas where further research is necessary are outlined.

Chapter 2: Background

This chapter gives some background into fault detection as well as some of the theoretical framework required by subsequent chapters. It is divided into sections on classical testing, random test generation, data compression, and probabilistic analysis of testing methods. Before proceeding, the reader should be aware of the distinction between a fault in a circuit, and its corresponding fault effect. A *fault* is a hardware failure, where something in the circuit is not performing correctly. A *fault effect* is the change in the output function of a circuit which results from a fault. A fault with no fault effect is called a *redundant fault*. Redundant faults are not considered in this thesis.

2.1 Classical Testing Methods

The most obvious of all testing methods is simply to cycle through all possible input patterns and compare the outputs with the known correct values. This technique is known as a *full functional test*. This was a reasonable method before the advent of medium or large scale integration. The problem with this scheme is that each correct output value must be stored, and with each additional input there are twice as many output values in an exhaustive test. For example, with a 20 input circuit, there are 1,048,576 different output values to be stored. This doubles for a 21

input circuit. Clearly, a more efficient testing scheme must be used for large circuits.

One variation on the intuitive method of a full functional test is to cycle through the input values on two circuits, one which is suspect and one which is known to be fault-free. This eliminates the problem of storing the correct result, but begs the question: How is the "good" circuit known to be fault-free? Even a circuit which performs an entire test flawlessly may contain faults which are intermittent and did not affect the first test. Intermittent faults may depend on factors such as temperature and the speed at which the inputs are presented. This comparison method, then, can show only that one of the circuits is faulty (though it cannot determine the faulty one), or it can show that either both are fault-free or both are faulty. Again, this method is unsuitable for general circuit testing.

Full functional testing of a circuit is typically not necessary, since a subset of the inputs can usually detect all faults. This is the basis for the classical method of circuit testing, which consists of generating a *test set* and a set of correct output values for that test set. The test set consists of a set of *test vectors*, where each test vector assigns a fixed value to each of the circuit inputs. A test vector is said to *cover* a fault in a circuit if the circuit's output (with the test vector as input) would differ under the

presence of the fault. Test sets are chosen, classically, as a minimal set to cover all possible faults. Typically the faults can be covered with substantially fewer test vectors than the 2^n required for an exhaustive test.

When it is said that a set of faults is to be covered, it is necessary to determine precisely what that set is. There are numerous methods by which a circuit can fail. These have been divided into three classes by McCluskey in [19]. These are: open interconnect, where a connection should exist but does not; shorted connection, where a connection exists that should not; and finally parameter out of specification, where some physical component of the circuit cannot function correctly because of a manufacturing defect. Furthermore, failures may not be catastrophic: that is, observable immediately after manufacture and forever after that. They may develop over time: i.e. the circuit may degrade, or, as mentioned above, the failures may be transient.

Failure may occur for a variety of physical reasons, many dependent on the physical layout of the circuit on the chip, the manufacturing process, and the materials used. Because of the difficulty of anticipating such failures, it is necessary to use a *fault model* to simulate the possible faults. Of course, it is only necessary to simulate the fault effects, rather than the actual faults.

A fault model must, if it is to be of any use, correspond to a high degree with the actual fault effects possible in a circuit. A variety of models exist, with the simplest, and still the most popular, being the *single stuck-at fault* model. In this model, one line of a circuit is assumed to be either shorted to ground (stuck at 0, or s-a-0), or shorted to the supply voltage (stuck at 1, or s-a-1). These stuck-at faults cause the circuit to behave as though the given line is a constant, rather than a function of other circuit elements. If a line "fans out" into two or more gates, the model permits these "fanout branches" to become stuck independently. Therefore, the total number of possible stuck-at faults is twice the total number of lines (connections between circuit elements, including fanout branches) in the circuit.

The number of lines in a circuit depends on whether the circuit is modelled at the gate level or the transistor level. In this thesis, all the circuits are modelled at the gate level: i.e. they are considered to be collections of logic gates (see [20] for more information). At the transistor level, the gates are replaced by their transistor equivalents. In this way, the gate model is an abstraction of the transistor model for circuits.

It turns out that the single stuck model does cover a large number of actual faults, but there are some that it does not. For example, the fault effect of many bridging faults does not correspond to any one possible in

the stuck-at fault model. In a bridging fault, two lines of a circuit are shorted together, and depending on the technology used, the resulting value on both lines may be either the logical AND or OR of the original values on each line. The number of possible bridging faults is of the order of the number of lines in the circuit squared (fanout lines are not included in this case), so in general there are considerably more possible bridging faults than single stuck-at faults. The physical layout of a circuit will make many bridging faults unlikely, but since the actual chip layout may bear little resemblance to a circuit diagram, the designer must be prepared for all bridging faults.

Another class of fault not covered by the single stuck fault model is the sequential fault. In this case, a strictly combinational circuit is made into a sequential circuit by the presence of a fault. One obvious way for this to occur is through a feedback bridging fault, where one of the shorted lines in a bridging fault is on a path from an input to the other line. This causes a feedback loop in the circuit, which, as with a standard flip-flop, may introduce an additional state (or memory element) into the circuit.

A final form of fault which cannot be thought of as a stuck-at fault is the stuck-open fault. This is peculiar to CMOS implementations, where it is an extremely important problem. The stuck-open fault causes the circuit

to "remember" its previous output. This dynamic memory effect means the stuck-open fault must be considered as a sequential fault. Various methods have been proposed to detect these stuck-open faults. One example is that of Zasio [40], where power current consumption, rather than logical behaviour, is measured. Zasio admits that the technique is slow, and offers little justification for it other than "experience shows", but the paper does help demonstrate that not all testing methods involve logic checks. Many additional works on stuck-open faults exist, two of the most recent being [3] and [5]. In the former, Bate and Miller emphasize the requirement of additional test vectors, while in the latter, Brzozowski investigates the use of additional hardware to detect the faults.

It is the task of the test set designer, in a classical scheme, to choose a fault model which will cover the majority of the faults which may occur in the circuit, and then to choose a test set to cover these. This is an expensive task, since it can involve simulating the circuit both in its fault-free state, as well as in each of the possible faulty states. From these simulations, test vectors which cover the faults are chosen. Algorithms exist to find these test sets, particularly the "D-algorithm". A good overview of these algorithms can be found in any standard reference on logic design and testing, such as [16] or [8]. The test set is usually chosen to cover all faults in the given fault model, which covers a large percentage of actual possible faults. Interestingly enough, there appear to be no

statistics on the number of actual faults covered by any fault model. It is hoped that many of the faults not covered by the model will still be covered by the test set. In practice, many test sets do not try to cover all faults in a model, but rather a high percentage (e.g. 98%) of them.

The high design-time cost of test set generation is not its only disadvantage, however. There is a similarly high run-time cost associated with loading each test vector, running it through the circuit, and comparing the output with the correct value, also loaded from an external file. Because of this requirement for loading external data, even the most advanced and expensive testers cannot simulate some circuits at maximum speed, thus missing many dynamic faults (those which appear only when the circuit is operated above some particular speed) as well as transient faults. The most expensive testers (which cost more than 1 million dollars) can be run at speeds of up to 12MHz on pre-loaded test sets of up to 4000 vectors, so the operating speed of much circuitry can be attained.

Because of the high costs associated with the classical test set scheme, testing has become a significant expense in the manufacture of chips. Many schemes to reduce this cost have been proposed. These fall mainly into two categories: random test generation and data compression.

2.2 Random Test Generation

Random test generation avoids much of the design time cost of test pattern generation by choosing the test patterns randomly. Two notable works in this area are Savir et al. [31] and Brglez [4]. The gist of these methods, as noted in [31], is to obtain random test patterns such that "$100-\epsilon$% of the population's single stuck faults would be detected with probability $1-\delta$." Savir would like to have $\epsilon < 2$ and $\delta < 0.001$. The principal property of the arguments for random test generation is that if a fault has probability p of being detected by a particular pattern, then p^{-1} random test patterns should detect it.

Savir notes the relationship between signal probability, the probability that a randomly-selected input vector will yield a value of 1 on the line in question, and the detection probability, which as implied above is the probability that a given input vector will cause an incorrect output value. In addition, he notes that previous attempts to examine the signal probability have all possessed the undesirable property of exponential space complexity. He introduces an approximate method, the "cutting algorithm", which gives bounds on the signal probability rather than exact values.

If the upper bound for a given fault is less than the acceptable value, δ above, then the fault is classified as an "easy" fault, otherwise it is a

"hard" fault. If there are too many hard faults, the design for the circuit should be modified in order to eliminate them. Problems in the algorithm result with reconvergent fanout branches, so several heuristics are provided to first eliminate them, and later reintroduce them. Although it is claimed that both time and space complexity are "almost linear", no proof of the claims is provided.

Brglez [4] proposes a similar method, where he measures the "1-controllability" of a line (the signal probability). He claims that his method produces testability estimates (detection probabilities) that are closer to the true values than those of Savir. He further claims that his simple method will work on "many typical networks containing reconvergent fanout", although he gives no clue as to which networks either do or do not fall into this class. As with other random test methods, Brglez does not attempt to find actual test patterns, but rather the number of such patterns that exist. Little mention is made of the complexity of his algorithm, although he implies that it is better than the exponential case decried by Savir. A final statement Brglez makes is that his definitions of controllability and observability of a fault differ enough from Savir's so that the problems of [32] (see below) are avoided.

In [32], Savir outlines some limitations he sees with existing testability measuring methods. Specifically, he points out that the standard

definitions of controllability and observability do not necessarily guarantee good testability in the practical sense, as opposed to the theoretical sense. In other words, a fault which comes out of a random test algorithm with a high value for "testability" may not, in fact, be easy to test. The reason for this claim is that while the set of vectors which "control" a fault (force the line on which it occurs to be set to a particular value) and the set which "observe" a fault (force the effect of the fault to be propagated to the output) may both be large, they may fail to overlap significantly. It is the overlap between the two sets which provides the actual test vectors, and thus while good controllability and good observability usually imply good testability, they cannot be guaranteed to do so. He claims that computing the testability is "in general NP-complete", but offers no supporting evidence for the claim. A better statement would be that the problem does not appear to be tractable in polynomial time. However, such distinctions belong more in works on computational complexity.

Aside from the disadvantages pointed out above, random pattern testability does provide a significant savings in design-time costs from the standard test set generation methods. This saving is only realized if exact knowledge of the coverage is not desired. If it is required, the circuit must be simulated in both the fault-free and faulty cases, eliminating the savings in design-time costs of random test generation. The run-time costs can always be decreased, since the test vectors no longer need to be loaded

from an external file, but may be generated pseudo-randomly, from a feedback shift register, for example. A large cost still remains, however, since the correct values for the pseudo-random inputs must be loaded in each case and compared for each input vector provided. Depending on comparator speed and buffering of values to be compared, the circuit may be run at close to its operating speed, thus detecting more transient faults than with the classical method. It is possible to reduce this run-time cost even more, and at the same time obtain additional advantages over classical testing methods, through another testing idea – data compression.

2.3 Data Compression

Data compression (also known as data compaction) is the label given to a set of techniques which use encoding schemes to reduce the length of the output vector to a reasonable sized entity called a signature. Typically the length of the signature is orders of magnitude less than the length of the output vector. A formal definition is given below:

Definition 2.1: A signature consists of a set of attributes of the output vector. A signature is said to cover a fault if some element of the signature changes in the presence of that fault.

Clearly, in an ideal situation, one would like the signatures of the fault-free circuit and any observably faulty circuit to be different. In prac-

tice, ideal signatures are not always easy to find. With any genuine data compression technique, several possible output patterns from the circuit will compress to the same signature. When the output vector of the fault-free circuit and that of one or more observably faulty circuits compress to the same signature, it is said that "aliasing" has occurred. The object of data compression techniques is to minimize aliasing.

Aside from the savings in design and run-time costs, as compared with the conventional test set methods, data compression testing methods have an additional advantage, this being the possibility of built-in self test. A built-in self test allows a chip to test itself, with a minimum of additional hardware and a high degree of accuracy. For example, suppose for a 20-input circuit a signature can be devised which is on the order of 20 bits. Assuming that the hardware to calculate the signature is simple (all methods investigated in this thesis have this property), then with minimal hardware (signature evaluator and 20-bit comparator) a self-tester can be inserted on the chip. If test sets were used, there could be hundreds of test vectors required, necessitating perhaps several kilobytes of storage, or in the case of random test vectors, hundreds of bits. This large storage requirement is unreasonable, whereas the small area required for the signature tester may well be worth the investment in chip area because of the savings gained in test expense.

The discussion of built-in testing, of course, is meaningless without some indication of the performance of actual signature methods. In the majority of works which investigate the performance of data compression methods, one assumption is used in evaluating the methods. This is the "all equally likely" assumption, which is discussed below. First note that the defining vector of a function of n variables has length 2^n.

Definition 2.2: (The "all equally likely" assumption) Let F be the output function of a circuit with n inputs in the fault-free state, and F′ be its output in some faulty state. The "all equally likely" assumption is that F′ may be any of the 2^{2^n} possible output patterns with equal probability.

Under this assumption all possible output patterns are assumed to be equally likely in the presence of a fault. That is, output patterns in the presence of faults are assumed to bear no relation to the correct value.

The above assumption differs from the standard coding theory assumption of a Poisson distribution of erroneous bits. In coding theory, an error is defined as an incorrect bit in an output stream. Typically, in a digital circuit, the fault effect of one fault contains many errors. Since the errors in the output come from logic faults, rather than line noise, most faults will affect many bits in the output stream, so it cannot be assumed that errors are independent events. Thus the Poisson distribution is clearly an invalid model. The "all equally likely" assumption is the simplest

description of possible error sequences. It states, in effect, that logic faults alter the output of a circuit in a manner that cannot be determined without knowledge of the circuit.

At first glance, this might seem a perfectly reasonable assumption, but it becomes clear that this is not so. When a given circuit is simulated under various fault conditions, the faulty output sequences appear to have patterns to them, but no general description of the patterns for different circuits seems to be possible. Smith [36] hypothesized burst errors, and erroneous bits separated by powers of 2 in the output sequence. Saxena [34] has generalized the latter assumption to a model where the erroneous bits are spaced by either some constant number of positions, or a multiple of that constant. Of course, many logic faults will affect two successive bits, in which case the previous models (with the exception of burst errors) simply degenerate into the "all equally likely" assumption.

It seems that a general description of the errors in functional output caused by logic faults remains elusive. For this reason, the "all equally likely" assumption is used, even though for actual circuits it appears that some errors are more equal than others.

The "all equally likely" assumption has historical justification for its use. It has been applied numerous times, especially in the area of shift registers, starting with the initial development of data compression [7] and

continuing to the present day [33]. A particularly good defense of the method can be found in [17]. For this reason, the theoretical results of this thesis are based upon this assumption, even though empirical data of subsequent chapters raise some questions as to its validity. In spite of its inadequacies, the "all equally likely" assumption provides a useful basis for comparison between various signature methods.

The most obvious flaw with the "all equally likely" assumption is that for any given circuit, all output patterns are not equally likely in the presence of a fault. Only a few patterns can actually be generated by faults. Unfortunately, however, there is at present no method of characterizing the patterns generated by logic faults. Burst errors do not suffice, nor do equally spaced errors. The patterns produced by logic faults in an exhaustive test have characteristics different from those in a purely random test. For example, a fault which causes a circuit to become vacuous in some input will cause repeated patterns in an exhaustive test, while its effect in a random test is unclear.

The "all equally likely" assumption is valid, though, if actual faults are distributed randomly throughout the space of possible output patterns. At present, the most generally applicable, accurate characterization of output patterns caused by logic faults allows for the possibility of any output pattern, all with equal likelihood. Therefore, the "all equally likely"

assumption continues, despite possible flaws, to be the most suitable framework on which to base the theoretical analysis of data compression.

There are two primary methods of data compression described in the literature. The first, and possibly the more well-known, is based on a Linear Feedback Shift Register (LFSR), while the second uses various counting schemes.

2.3.1 LFSR Testing

The original work on the linear feedback shift register as a means for circuit testing is that of Frohwerk [7]. Shift register sequences have been studied extensively in coding theory, and indeed Frohwerk references two such works ([24] and [9]) in his paper. Essentially, Frohwerk adapts the coding practices used for CRC checks of data transmission to fault detection in circuits. The approach was chosen because of the similarity in the two ideas, although there are substantial differences. For instance, in data transmission an error may occur in any bit with equal probability. Hence, any two patterns with equal Hamming distance from the correct transmission have equal probability of occurring, assuming that errors are independent events (Poisson distribution). In actuality, errors tend to come in bursts (see [37] for a more detailed discussion), so the independence assumption may not be entirely valid. Nonetheless, the point remains that

in data transmission virtually all possible erroneous arrangements can occur in the presence of line noise. In fault detection, on the other hand, there are only a relatively small number of faults. Therefore, on an output stream of say 1000 bits, only several hundred of the 2^{1000} possible output arrangements will actually occur in the presence of faults. It is this distinction that separates fault detection from coding theory and causes many of the latter's arguments to fail when applied to the former. This is discussed in detail in later chapters.

The LFSR's used in testing nonetheless attempt to use the error detecting abilities of the shift register for fault detection. In practice they have been successful – witness the popularity of Hewlett-Packard's Signature Analyzer. The random nature of the LFSR does not restrict its use to particular types of circuit. In addition, the circuitry required for such a tester is compact and can easily be included on a chip. The biggest disadvantage of the LFSR is also its greatest advantage: the randomization of output functions makes design for testability difficult, if not impossible.

Further analysis of shift register behaviour, including mathematical values for aliasing probabilities and the justification for their use, is provided in the next chapter.

2.3.2 Counting Testing Methods

Counting schemes are another form of data compression. Rather than using a shift register to obtain a signature, some attribute of the output is counted. For example, one of the earliest such methods to be proposed was transition counting [11]. This technique counts the number of 0-1 and 1-0 transitions of the output vector for some input sequence. Another set of counting methods has overshadowed transition counting, though, in recent literature. These are spectral-based techniques, developed originally as syndrome, or one's, counting by Savir in [29] (although there are previous references to the idea of one's counting, see [17] for example).

Transition counting depends on a set of inputs being provided in a particular order. For a given set it is possible to arrange the input vectors so that only a single transition occurs, which then makes the probability of aliasing small. Setting the inputs so that the maximum possible number of transitions occurs also reduces the aliasing. Unfortunately, it may be difficult to generate the test patterns in the sequence required for the minimal aliasing, and sequences which are easier to generate may be less effective for testing. In general, a transition counter has higher aliasing probability than a comparably-sized LFSR. Furthermore, transition counting is not amenable to design for testability.

Because of the deficiencies of transition counting, it will not be considered further in this thesis. Syndrome counting, on the other hand, will be analyzed extensively in subsequent chapters. It is possible to design circuits which are guaranteed to be syndrome-testable, something not easily attainable with other data compression techniques. The biggest disadvantage to syndrome counting is that values for the syndrome tend to cluster around the midrange: i.e., most functions are about half 1's and half 0's. For this reason, under the "all equally likely" assumption, there is a large amount of aliasing for syndromes near $\frac{1}{2}$. The syndrome is the number of 1's (also known as the "weight") in the output vector normalized so that its value lies between 0 and 1.

Spectral coefficients, derived from the Rademacher-Walsh spectrum of a boolean function, are related to syndrome counting and have similar aliasing probabilities. These coefficients, in fact, include the weight of the function, as well as the "height" – a coefficient that relates a function to the sum modulo 2 of all its input variables. The transformation used to obtain the coefficients will be given, together with a more detailed explanation, in chapter 4. Suffice to say here that many authors have investigated and continue to investigate the use of spectral coefficients for testing purposes.

2.4 Probabilistic Analysis of Data Compression

This section introduces probabilistic analysis of data compression techniques and provides an overview of some previous work in the area. Probabilistic analysis of signature techniques represents the thrust of the work of this thesis. The purpose of such analysis is twofold: first, compression techniques with a high probability of success must be developed; and second, the probability of their success must be demonstrated to be high.

Several data compression techniques, LFSR signature analysis, syndrome testing and use of spectral coefficients, have already been mentioned. These and others have success probabilities derived and analyzed in the remainder of this thesis. The probability of a signature being successful in detecting a fault is by definition one minus the aliasing probability of the signature. Thus, if a signature has a 1% probability of aliasing, it has a 99% probability of success. Aliasing and success probabilities must be derived in terms of a probabilistic model of possible errors.

There are two ways of deriving such a model: theoretical and empirical. In a theoretical examination, actual circuits are ignored in favour of their output functions. A framework is used to describe the erroneous output functions which can result from a correct one, permitting expressions to be derived for the aliasing probability. The most commonly used frame-

work is that provided by the "all equally likely" assumption. In this case, the results obtained are valid for any circuit implementing any output function, within the context of the theoretical model. So, the accuracy of the results depends on the validity of the model. The theoretical method is as follows: The signature records some attribute of an n-variable output function. It is possible to calculate the number of n-variable functions which have the same value for that attribute as the original. Those functions with the same attribute value as the original are those which will alias if the signature is applied to them. Since all output functions are considered to be equally likely in the presence of a fault, the aliasing probability can be determined by dividing the number of functions with the same attribute value by the total number of functions.

The second method of determining the effectiveness of a signature is an empirical model. In this case, some circuits are simulated and the aliasing of a given signature in the presence of some faults is recorded. The faults simulated, of course, must be based on some fault model, e.g. single stuck-at. Since the set of circuits simulated can never include all circuits under all possible fault conditions, the empirical set can contain only a sample of all possible data. The probabilistic effect in this case is experimental error. The empirical set chosen is said to be "typical" of all circuits. Because of the large variety of digital circuits in existence, it is difficult to produce any reasonably-sized set that can be claimed as

unquestionably representative. This is not to say that any empirical analysis is futile – empirical evidence is what will eventually determine the usefulness of any signature technique. Nevertheless, it is important to recognize the existence of an error factor in any empirical data.

In this thesis, both empirical and theoretical methods are used to examine various types of signature. In chapters 3 and 4, the "all equally likely" assumption is used to develop aliasing probabilities for signature techniques: LFSR's in chapter 3 and spectral coefficients in chapter 4. In chapter 5 these probabilities are compared to the actual values found for a set of sample circuits.

Data compression techniques are generally very amenable to probabilistic analysis. Most authors include some empirical evidence as to the behaviour of a signature they are introducing, but the number who introduce a theoretical analysis of the signature seems to vary with the technique used. The first work in the probabilistic analysis of data compression was in connection with the first LFSR paper [7]. This was extended in [36] and [6]. Perhaps because of its association with coding theory, LFSR testing has been carefully studied in terms of aliasing effects. The results of these determinations are presented in chapter 3.

Probabilistic analysis has also been carried out for syndrome testing. Savir's original paper [29], though, does not include any such analysis

because he is more concerned with design for testability. He implicitly requires all circuits to have 0 aliasing probability because they are designed specifically to be syndrome-testable. This work, as well as that of [21], [22], and [23], differs from probabilistic analysis in that the problem is stated as "how can 100% coverage for a circuit be achieved?" rather than "how close to 100% coverage is this signature likely to provide?" If 100% covering signatures were in general easy to find and easy to apply, then probabilistic analysis would be unnecessary. Because the problem of finding such signatures seems to be difficult for complex circuits, developing signatures with low aliasing probabilities is a reasonable alternative.

In the first section of chapter 4, an analysis of syndrome testing is performed. This repeats a well-known result, that the distribution of weights is heavily concentrated towards the midrange value. It is this observation that has lead some authors to propose methods of changing the weight for test purposes. This will place the syndrome on the edges of its distribution and hence decrease substantially the aliasing probability. Agarwal and Zorian, in particular, ([1], [41], and [42]) have developed such methods of altering a function's weight during a test. They desire the placing of the fault-free value for the syndrome closer to the edges of the distribution, assuming that the faulty values will remain clustered around $\frac{1}{2}$.

The technique proposed is interesting. It is unfortunate that none of [1], [41], or [42] offers any empirical justification to support their approach. Perhaps future works will contain examples to demonstrate empirical improvement with their technique. Once again, the most serious flaw in the solely theoretical analysis is that only a minute fraction of actual error patterns can ever be produced by logic faults. In fact, even with the 2^{-60359} improvement in aliasing probability theoretically predicted in [1], it is still possible for every single fault in the circuit to have precisely the same syndrome value after the functional transformation. Furthermore, these transformations require complex additional circuitry, as well as additional time to perform the test, reducing the ability of the syndrome counter to test a circuit "at speed". These last two problems appear to have been alleviated in the most recent work [42], although the method of designing the modifier for the circuit has become quite complex.

Agarwal and Zorian used the "all equally likely" assumption to arrive at their conclusions. This assumption was also used by Hsiao and Seth, [12] and [13], with regard to Rademacher-Walsh coefficients as well as the syndrome. They divide their work into two sections: deterministic compact testing and random compact testing. It is in the latter that the "all equally likely" assumption is used. (They define compact testing to be data compression). The first paper, [12] deals with both types of compact

testing, while the second is concerned solely with random compact testing.

In [12] it is shown that any two-level circuit is 100% covered for all single stuck-at faults by the signature consisting of the syndrome and the height (see chapter 5, section 5.3.5), provided that the height is non-zero. (It is shown later in this thesis that such a signature has empirically very high coverage regardless of the nature of the circuit or the value of the height). The other results they obtain for deterministic testing are not particularly noteworthy. For instance, they state that an exclusive-or function can be tested by a signature consisting solely of the non-zero spectral coefficient determined by the function, and they give an expression for the number of functions with a particular spectral coefficient of a given value.

For random compact testing, the papers make the distinction between type I and type II errors (the probability of rejecting a good unit, and accepting a bad one respectively). The first paper [12] skims very briefly over the entire subject; it is only in [13] that some detailed analysis is made. Using this scheme, it is possible to obtain a type I error, since the test inputs are generated randomly, in a manner unknown before the test. In most data compression methods there is no possibility of such an error because of the existence of a reference signature. Their proposed method allows testing of larger circuits, but does introduce the possibility of a

good unit failing the test. It would be interesting to modify their random compact testing method to include a reference signature, and then carefully analyze the probability of type II errors. Performing such an evaluation could reduce the problem of test time for exhaustive testing methods.

With regard to type II errors, the authors obtain some probabilistic results for a signature consisting of a largest magnitude spectral coefficient. These are also applicable to a syndrome signature. In addition, they present some graphs which demonstrate that aliasing probability is significantly lower (using the "all equally likely" assumption) for coefficients with values towards the extremes of the distribution. They do not show, however, how an "average" circuit might behave, or indeed what the average value of a coefficient might be. Their results suggest that the largest coefficient in the spectrum should be chosen as the signature, but they present no empirical evidence to support this claim.

As an aside here, observation has confirmed (see [15] for a list of function classes and the magnitude of their coefficients) that every circuit of $n \leqslant 5$ inputs has a spectral coefficient of absolute value $\geqslant 2^{n-3}$, but a proof for larger values of n is elusive, and therefore not presented in this thesis. The value actually listed in [15] is 2^{n-2}, because the coefficients were computed using {+1,-1} encoding, rather than the conventional {0,1}.

It is probable that should the result be correct, the {+1,-1} encoding will be used in the proof, since it eliminates "special case" coefficients. If this result were correct, then every circuit would have a coefficient which, by Hsiao and Seth's argument, would have very low aliasing probability.

One of the more recent works on aliasing probability is that by Savir in [33]. He compares the average aliasing probability of LFSR's, syndrome counters and transition counters and arrives at the conclusion that while all are good, LFSR's are the best. Savir's results are for the average case, and thus differ from those previously considered. Agarwal considered modifying the function in order to obtain a syndrome with a value at the edge of the distribution, thus having aliasing probability orders of magnitude lower than that of an LFSR. Hsiao and Seth, on the other hand, advocate choosing the largest value coefficient available, again pushing the counting techniques to the edge of the distribution, where their performance is theoretically best. Nonetheless, the expression Savir gives is useful for comparison of the methods as applied to general functions with no modification, using the "all equally likely" assumption, and concur with results obtained in this thesis.

Savir's paper contains no empirical results and no numerical values for the expressions he gives. Some of these are supplied by Saxena [34]. This paper does not use the "all equally likely" assumption, but rather a

view of errors as dependent and evenly spaced by multiples of some constant value. The circuits investigated by Saxena are "random logic", although it is unclear whether he means randomly constructed or randomly chosen. Only four circuits are analyzed, but their average complexity is about 200 gates. He shows that the predicted coverage based on his error analysis and that practically observed are approximately equal. He concludes that the algorithm designed to predict the coverage (which is not included in the paper) gives a lower bound to the performance of LFSR testing and is essentially linear in the number of gates in the circuit. The claim of a lower bound is not entirely true, even considering his own results. It appears that the algorithm gives an approximate value, which may be either higher or lower than the actual coverage. This is distinct from a guaranteed lower bound, such as that obtainable for syndrome testing.

A review of the literature thus indicates a lack of a comprehensive analysis of the various data compression methods. This thesis attempts to alleviate the problem by encapsulating previous results for the theoretical aliasing probabilities of LFSR and syndrome testing, deriving some new theoretical probabilities for spectral and spectral-related signatures, and finally performing an empirical analysis of the various methods. The results of the empirical analysis allow the evaluation of both the testing schemes and the theoretical framework with which they have previously

been compared.

We define now a notation used throughout the remainder of this thesis. Given some data compression technique z applied to a function of n variables, $P_w(z,n)$ denotes the worst-case aliasing probability, while $P_{av}(z,n)$ gives the aliasing probability in the average case.

Chapter 3: LFSR Testing

This chapter gives some results for aliasing probability of linear feedback shift registers. None of the material in this chapter is new. The results derived here were originally developed by Frohwerk in [7], and extended by Smith in [36] and Carter in [6].

Linear feedback shift registers perform polynomial division over the Galois Field GF(2). There are two ways to construct an LFSR (see figures 3.1 and 3.2, taken from [36]), which have different behaviour in terms of their contents after division. Both LFSR's divide by the same polynomial, in this case $x^5 + x^4 + x^2 + 1$. Note that any sequence of k bits can be considered to be a polynomial of degree k-1. Thus a bit sequence b(1),b(2),...,b(k) can be considered to be the polynomial $b(1)*x^{k-1} + b(2)*x^{k-2} + \cdots + b(k)*x^0$. So, the sequence 10110001 corresponds to the polynomial $x^7 + x^5 + x^4 + 1$.

The LFSR of figure 3.1 has a feedback connection from the output which is then exclusive or'ed with the bits as they are shifted through the register. The input polynomial is passed, one bit at a time, to an initially all-zero shift register. The bits which leave the output are the quotient, and those remaining in the register after the last input bit has been passed in are the remainder.

The LFSR of figure 3.2, has feedback taps at various points, rather than an exclusive-or connection between blocks of the shift register. These taps are exclusive-or'ed together, and the result exclusive-or'ed with the input polynomial. The output of this shift register is still the quotient of the polynomial division, but in this case the contents of the shift register after the division are not the remainder. This signature, left after the polynomial division, while not the remainder, nonetheless has similar properties to that of the LFSR in figure 3.1.

In both cases, the signatures of a faulty circuit and its fault-free counterpart will be the same only if the erroneous input sequence differs from the correct input sequence by a multiple of the polynomial used by the LFSR in the polynomial division. In short, the two techniques are isomorphic, although the nature of the isomorphism is difficult to determine [36].

The feedback points of the shift register determine the polynomial used in the division in different ways, depending on the shift register used. Suppose an m-bit shift register of the type in Figure 3.1 has feedback points at positions $p_1, p_2, \cdots, p_k$. (That is, exclusive-or gates between stages $p_i - 1$ and p_i). The polynomial used in the division is then $x^m + x^{p_1 - 1} + x^{p_2 - 1} + \cdots + x^{p_k - 1}$. If an m-bit shift register of the type in Figure 3.2 has feedback taps at positions $p_1, p_2, \cdots, p_k$, then the polyno-

mial used in the division is then $x^m + x^{r_1} + x^{r_2} + \cdots + x^{r_k}$, where each $r_i = m - p_i$.

Frohwerk's LFSR tester, [7], is of the type of figure 3.2. These shift registers are easier to implement on a single chip than those of figure 3.1, so are more useful for built-in self-test. The LFSR performs the polynomial division over the Galois field GF(2) of the input sequence by a fixed primitive polynomial. The LFSR design does not require a primitive polynomial, but clearly it makes sense to use one in order to obtain the largest possible number of signatures. The design proposed, and in fact implemented by Hewlett-Packard, consists of a 16-bit shift register with taps at positions 7, 9, 12 and 16. These taps are summed modulo 2, using exclusive or gates, and the result is exclusive or'ed with each successive input bit (see figure 3.3). The resulting signature is determined solely by the input stream, but appears almost random in that two sequences separated by a very small Hamming distance can produce vastly different signatures.

The polynomial used by Hewlett-Packard, $x^{16} + x^9 + x^7 + x^4 + 1$ is primitive, so the shift sequence is of maximal length (2^{16}-1, all patterns except 0). There are 2048 polynomials which produce this maximal length sequence. Some are not irreducible, so they tend to cluster undetectable patterns together, but there are many other irreducible ones. The HP poly-

nomial appears to behave very well as a signature. As to whether or not it is the best selection, no evidence exists.

Frohwerk derived the aliasing probability of the LFSR as follows. Consider an m-bit LFSR. There are 2^m possible signatures. Now, given an exhaustive test of an n-input function, there are 2^n output values. This means that there are 2^{2^n} possible functions of n inputs. Through the use of a primitive polynomial of degree m in the shift register, Frohwerk claims that the patterns will be scattered essentially randomly (this claim is supported by Savir in [30]). Therefore, one can assume that the 2^{2^n} patterns will settle evenly into the 2^m signatures, giving:

Result 3.1: The total number of functions having a given signature is:

$$2^{2^n-m} \tag{3.1}$$

This allows the immediate determination, using the "all equally likely" assumption, of the aliasing probability of an LFSR as total number of functions with a given signature, not including the correct one, divided by the total number of functions of n variables, or:

Result 3.2:

$$\frac{2^{2^n-m}-1}{2^{2^n}} \tag{3.2}$$

As 2^n becomes large in comparison to m, expression (3.2) asymptotically approaches:

Result 3.3:

$$2^{-m} \tag{3.3}$$

Note that for any m bit LFSR, the aliasing probability could not be, on average, less than given by expression (3.3).

Much additional analysis of the LFSR testing technique has been done, with perhaps the most important results being those of Smith [36] and Carter [6]. Smith was the first to carefully analyze shift register behaviour, and concludes that the "all equally likely" assumption is not entirely valid. He provides an example of an extreme application of the assumption, in which he concludes that feedback is unnecessary and that the last 16 bits of the output stream are adequate as a signature. If the "all equally likely" assumption was in fact universally applicable, then this conclusion would be correct. The suggested signature would never be used in practice, however, since it would not cover many faults in an actual circuit.

Smith then measures the performance of the shift register signature technique in terms of capture of two kinds of errors: burst errors, and repeated use faults. In the former, d erroneous bits occur within n consecutive bit positions, while in the latter, erroneous bits are spaced at

intervals equal to a power of 2. He finds that even with these additions, the aliasing probability of an LFSR is unchanged.

Carter [6] "gives theoretical justification to the use of several signature testing techniques". In effect, he proves several theorems which give the aliasing probability under certain conditions. The requirements he sets for the signature test are that the inputs must be presented in random order, the feedback connections made randomly, and finally that (in the case of multiple output circuits) connections to the shift register must be random. He obtains two different results, depending on the nature of the shift register, one of which uses the "all equally likely" assumption, while the other does not.

With regard to his assumption that inputs be presented in random order, Carter suggests that the pseudo-random order generated by an LFSR may not be adequate for input combinations. In an IBM research report, however, Savir [30] claims otherwise. Carter's worry is primarily based on the possibility of both input and output LFSR's having the same feedback connections, which would lead to correlation between the two and destroy the randomness of the approach. Savir, on the other hand, concentrates only on the random nature of a single LFSR output and concludes that under a test he has developed it is "close to a perfect random integer generator".

Carter's first result assumes that a single error is present in the output stream of the faulty function, and that the input patterns are presented in random order. When an exhaustive test is assumed, he shows that the aliasing probability of an LFSR for a circuit of n inputs is less than

$$2^{-n+2}$$

assuming that the shift register is of length $> n$.

Carter further states that it is his personal belief that in fact the aliasing probability is less than

Result 3.4:

$$2^{-n} \tag{3.4}$$

but that he is unable to show that this is the case.

This result is interesting, because the length of the shift register is not considered. A longer shift register, though, can produce aliasing probabilities considerably lower than expression (3.4). For instance, if the length of the shift register is 2^n, then clearly there will be no aliasing whatsoever if no feedback taps are taken. In fact, for a 16-bit shift register with the standard HP connections (see Figure 3.3), there is no aliasing for functions of either 3 or 4 inputs. If the shift register were only 5 bits long, there would be aliasing of at least 2^{-5} for each type of function.

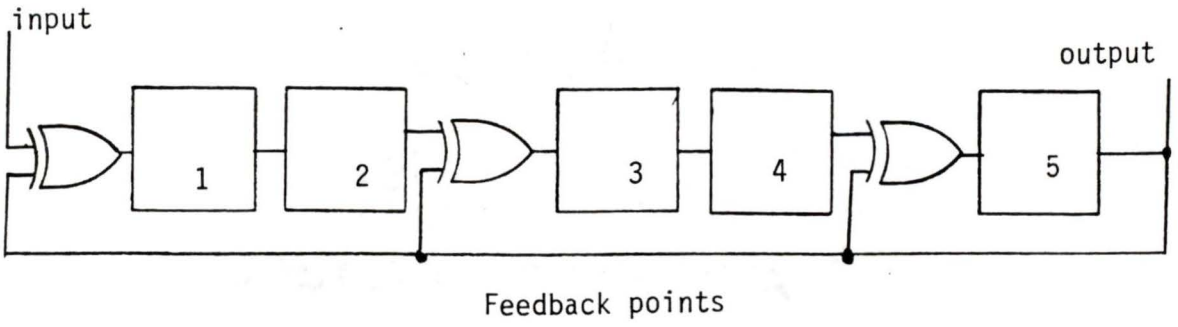
Carter's second result arises from a randomly selected primitive polynomial for use in the LFSR. In this case, using the "all equally likely" assumption, he shows that expression (3.3) gives the aliasing probability, for a shift register of m bits and any number of input patterns. He neglects in this case to point out directly that $m \leqslant \log_2$(number of input patterns), or $m \leqslant n$ in the case of an exhaustive test, for his results to apply, although he implies it by his definition of a primitive polynomial.

For the remainder of this thesis, expression (3.4) will be used for the aliasing probability of a shift register of length equal to that of a syndrome counter for a given function (n bits), while the following expression (a combination of expression (3.3) and expression (3.4)) will be used for the standard 16 bit LFSR. Thus, for an LFSR we have:

Result 3.5:

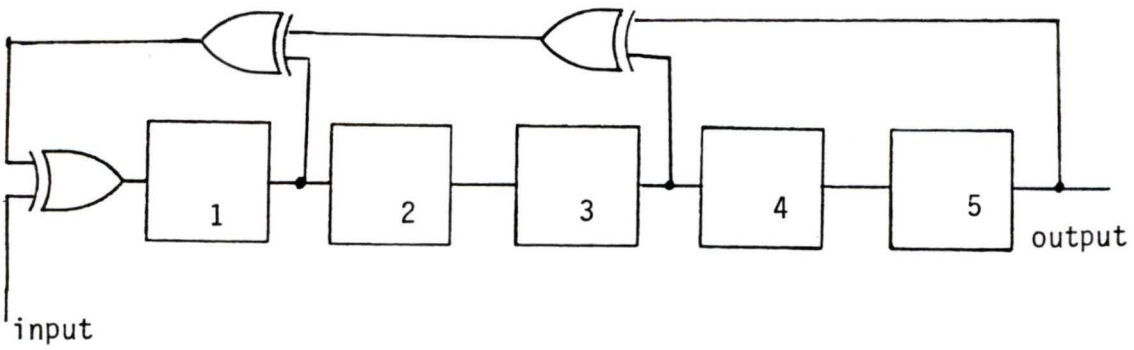
$$P_w(LFSR,n) = P_{av}(LFSR,n) = \max(2^{-16},2^{-n}) \qquad (3.5)$$

Note that for any LFSR which uses a primitive polynomial, the pseudo-random distribution of the signatures implies that the average and worst-case aliasing probabilities are essentially the same. This means that any of the expressions listed in this chapter can be considered as representing either case.



LFSR for dividing by $x^5+x^4+x^2+1$, with
correct remainder

Figure 3.1



LFSR for dividing by $x^5+x^4+x^2+1$, with
random contents after division

Figure 3.2

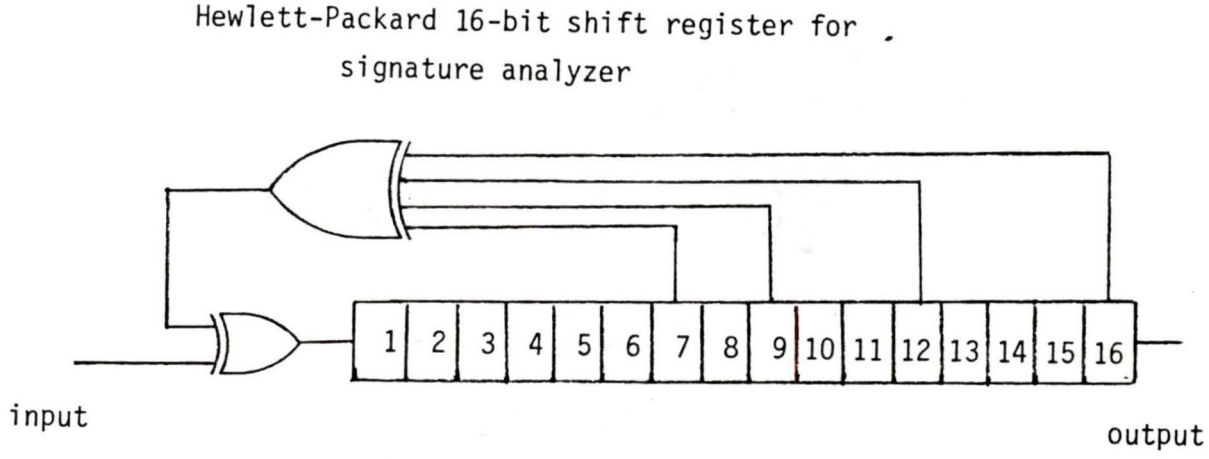

Hewlett-Packard 16-bit shift register for signature analyzer
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
input
output

Figure 3.3

Chapter 4: Spectral Testing

In the previous chapter, the aliasing probability of LFSR testing was reviewed. Here we present similar results for some testing techniques based on spectral coefficients. The expressions derived here permit comparison between the two data compression methods in terms of their error coverage.

As mentioned in chapter 2, spectral testing includes both syndrome testing and the use of other Rademacher-Walsh coefficients for testing. Here we investigate both of these, as well as a new related method, 4-valued syndrome testing. First, some definitions and background information are required.

Savir [29] defines the syndrome as the signal probability of the output line. In other words, the syndrome is defined, for a given function F of n inputs as: $S(F) = \frac{W(F)}{2^n}$, where W(F) is the weight of the function; that is, the number of 1's in the output vector. In his original paper Savir presents the definition of the syndrome, outlines some methods for design for syndrome testability (including examples), and gives a means of partitioning large circuits into subcircuits. The last issue is important, since the syndrome technique requires that every input combination be cycled through (in order to obtain the weight). Since the typical upper limit on

test time for a given chip is under 10 seconds, this means, for example, that with the inputs presented at a rate of 1MHz, only 2^{23} combinations can be presented in less than 10 seconds, putting the upper limit of testable combinational circuits at 23 inputs. If sequential circuitry is tested combinationally, through scan techniques such as LSSD, the number of inputs possible is further reduced. A curious footnote is that by 1983 [31], Savir had changed his mind about partitioning, and decided that "good partitioning programs are virtually nonexistent". This may not be the case, however, given works such as [25], where Roberts and Lala describe a partitioning algorithm which is able to deal with reconvergent fanout, and even basic sequential elements such as flip-flops.

In any case, syndrome testing, as with other data compression methods, allows the circuit to be tested at maximum operating speed, thus detecting many of the transient faults missed by test-set methods. The work of Miller and Muzio [22], shows that syndrome-untestable faults have definite characteristics, unlike those undetectable by an LFSR. This permits design for testability, something not readily possible with many other techniques. The major disadvantage of syndrome testing is that the distribution of syndrome values centers around $\frac{1}{2}$, with very few functions at the edges of the distribution (near 0 or 1). Thus, syndromes for fault-free and faulty circuits tend to be clustered around the midrange

syndrome value. This problem is discussed later in this chapter.

Now, while the range of a syndrome is (0,1) (neglecting the constant functions), an actual tester will have a range of 0 to 2^n, for a function of n inputs. This is because the simplest implementation of a syndrome tester is just a weight counter (see Figure 4.1). It is clear that for functions of n inputs, the syndrome and weight are isomorphic, and indeed it is common to refer to the weights as the "syndrome" in this case.

Of course, there is no need to use an actual counter to evaluate the syndrome. An autonomous LFSR of n bits, using a primitive polynomial for the feedback taps, and having the function output gate the clock (see Figure 4.2, or [42] for a full description) will use considerably less silicon area than a counter, while obtaining signatures isomorphic to a weight count.

The advantage, then, of syndrome testing is that it is amenable to a deterministic analysis of fault coverage. A recent paper by Robinson and Saxena [27] has indicated that syndrome coverage is orthogonal to that of LFSR testing. This result suggests the combination of the two methods will produce a signature which may combine the best assets of both types of testing. This signature is analyzed in more detail in chapter 5.

Another counting scheme related to syndrome testing is the use of Rademacher-Walsh spectral coefficients. These coefficients arise from the

Hadamard matrix, which is recursively defined for a given n as shown below:

$$T_0 = [1]$$

$$T_n = \begin{vmatrix} T_{n-1} & T_{n-1} \\ T_{n-1} & -T_{n-1} \end{vmatrix}$$

This matrix is used as a transform from function values to spectral coefficients. The spectrum obtained from this transform is isomorphic to the original function, since the matrix is invertible. In fact, the inverse of the Hadamard matrix is itself, with the exception of a constant factor. A more complete discussion of the transform may be found in [14].

As an example of the Hadamard matrix, consider T_3, applied to some function f (f_i denotes the i^{th} output value).

$$\begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{vmatrix} \begin{vmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{vmatrix} = \begin{vmatrix} r_0 \\ r_1 \\ r_2 \\ r_{12} \\ r_3 \\ r_{13} \\ r_{23} \\ r_{123} \end{vmatrix}$$

Notice that r_0 is simply the weight of the function, hence syndrome testing and spectral testing are related. The labeling of the other coefficients is derived from their relationship to certain exclusive-or functions. For example, r_{12} will have the value -2^{n-1} (its minimum value) iff

the function being transformed is x_1 exclusive-or x_2. Similarly, the last coefficient $r_{12...n}$, sometimes called the "height" [18], will take its minimum value only when the function consists of all n variables exclusive- or'ed together. Only the complement of such a "minimum" function produces the maximum value for a given coefficient, 2^{n-1}.

The other spectral coefficients have the same distribution as the weight, only they range from -2^{n-1} to 2^{n-1}. To build a device to test for a given coefficient r_α, each of the variables referred to by alpha must be exclusive-or'ed together: i.e. if $\alpha = 13$ then inputs x_1 and x_3 must be exclusive-or'ed together. The output of this exclusive-or is then exclusive-or'ed with the function output, and the weight of the resulting function is taken (see Figure 4.3).

Much work has been done on producing signatures based on the Rademacher-Walsh spectrum, see for instance [22], [13]. This thesis is not primarily concerned with finding signatures for various circuits, but rather with analyzing the effectiveness of these signatures, based on various probability models.

All expressions derived are based on the "all equally likely" assumption, and hence are an accurate reflection of error detection, though not necessarily of fault detection. In the next chapter, the results obtained here are compared with some empirical data from actual circuits. The

experimental evidence aids in evaluating the validity of the "all equally likely" assumption. Here it is used as our base and this allows us to obtain formulae which can be directly compared with those previously listed for the LFSR.

4.1 Syndrome Testing

We now investigate the aliasing probabilities of a syndrome counter. An earlier version of some results in this section was contained in [2]. In addition, expression (4.3) was independently derived by Savir and McAnney in [33].

We will consider single-output functions of n inputs. (The restriction to single-output functions is for simplicity only. Methods of extending these concepts to multiple-output circuits can be found in, for example, [26] and [35]). For such functions, the distribution of weights is clearly binomial. The total number of functions with weight w ($0 \leqslant w \leqslant 2^n$) is given by:

$$\binom{2^n}{w}$$

Note that $\binom{x}{r}$ denotes a binomial coefficient, and is commonly pronounced "x choose r". Hence we must "choose" where to place w ones among 2^n locations.

The binomial distribution is symmetric. Therefore, to determine the worst case aliasing, $P_w(synd,n)$, we should consider the mid-range value, where the weight is 2^{n-1}. Consequently the number of n-variable functions with the same weight is, in the worst case,

$$\begin{pmatrix} 2^n \\ 2^{n-1} \end{pmatrix}$$

Since the number of possible n-variable functions is 2^{2^n}, the following theorem may be stated:

Theorem 4.1: The worst-case aliasing probability, $P_w(synd,n)$, for a syndrome test of a function with n inputs is, under the "all equally likely" assumption:

$$P_w(synd,n) = \frac{\begin{pmatrix} 2^n \\ 2^{n-1} \end{pmatrix} - 1}{2^{2^n}} \tag{4.1}$$

Proof: Obvious.

The complexity of calculating expression (4.1) increases exponentially. Because of this, it is helpful to use Stirling's approximation in order to evaluate the factorials. Recall, from function theory (see, for instance, [38]) that Stirling's approximation is:

$$k! \sim (2\pi k)^{-\frac{1}{2}} e^{-k} k^k$$

Note that $\sim$ refers to "asymptotically equal to", while $\approx$ means a very

good approximation for the values in question and makes no statement about asymptotic behaviour.

Applying Stirling's approximation to expression (4.1) yields expression (4.2):

Result 4.2:

$$P_w(\text{synd},n) \sim (\pi 2^{n-1})^{-\frac{1}{2}} \tag{4.2}$$

These expressions may be compared with $P_w(\text{LFSR},n)$, expression (3.5). Full results are tabulated in section 4. 4.44, but some statements may be made here. Both methods can give better than 99% error coverage even for all $n > 13$. In fact, a 7-variable function is covered by an LFSR with 99.2% probability, and a 13-variable function is covered by a syndrome count with probability of at least 99.1%. These are acceptable results in many cases, but the question of exactly what level of coverage is required in general remains open. For a manufacturing process with low yield, fault coverage of 99% is inadequate. On the other hand, a built-in self-test for a non-critical component with 99% fault coverage is likely "good enough".

Another point to note is that while both syndrome counting and LFSR signatures have acceptably high error coverage even for functions of only a few inputs, $P_w(\text{LFSR},n)$ decreases faster than $P_w(\text{synd},n)$ for $n \leqslant 16$. If

the shift register is always longer than n bits (comparable hardware size to the syndrome counter) then expression (3.4) may be used for P_w(LFSR,n). In this case,

$$P_w(\text{LFSR},n+1) = \frac{1}{2}P_w(\text{LFSR},n)$$

while

$$P_w(\text{synd},n+1) \sim \frac{1}{\sqrt{2}}P_w(\text{synd},n)$$

So far we have discussed only the worst-case error coverage of the two methods. The nature of the distribution implies that $P_{av}(\text{synd},n) < P_w(\text{synd},n)$, so it is worth investigating whether or not significant improvement in aliasing results from considering the average case, rather than the worst case, for error coverage.

Theorem 4.3: (proof due to [28]) The average probability of aliasing for a syndrome counter under the "all equally likely" assumption is asymptotically given by

$$P_{av}(\text{synd},n) \sim (\pi 2^n)^{-\frac{1}{2}} \tag{4.3}$$

Proof: There are $\left\lfloor \binom{2^n}{j} \right\rfloor$ functions of n variables with weight equal to j. Hence, the probability of a given function having weight j is

$$\frac{\binom{2^n}{j}}{2^{2^n}}$$

and the probability of aliasing for given weight j is

$$\frac{\binom{2^n}{j}-1}{2^{2^n}}$$

To compute $P_{av}(synd,n)$, simply multiply the probability of a given function having weight j by the probability of aliasing for that weight j, and sum over all j. This gives average aliasing probability of:

$$P_{av}(synd,n) = \frac{1}{2^{2^{n+1}}} \sum_{j=0}^{2^n} \binom{2^n}{j} \left[\binom{2^n}{j} - 1 \right]$$

The following identities are standard combinatorial results [10]:

$$\binom{2^n}{j} = \binom{2^n}{2^n-j} \quad \text{and} \quad \sum_{j=0}^{2^n} \binom{2^n}{j} = 2^{2^n}$$

So we may write:

$$P_{av}(synd,n) = \frac{1}{2^{2^{n+1}}} \left[\sum_{j=0}^{2^n} \binom{2^n}{j} \binom{2^n}{2^n-j} - 2^{2^n} \right]$$

The Vandermonde convolution, [10], gives the following result:

$$\sum_{j=0}^{N} \binom{N}{j} \binom{N}{N-j} = \binom{2N}{N}$$

Applying this convolution gives:

$$P_{av}(\text{synd},n) = \frac{1}{2^{2^{n+1}}}\left[\binom{2^{n+1}}{2^n} - 2^{2^n}\right]$$

For $n \geq 4$, the $-\frac{2^{2^n}}{2^{2^{n+1}}}$ term becomes insignificant, and we may again use Stirling's approximation of the first term to approximate the entire expression.

Thus for $2^n \to \infty$ $P_{av}(\text{synd},n)$ reduces to $(\pi 2^n)^{-\frac{1}{2}}$. Notice that $P_{av}(\text{synd},n) \sim \frac{1}{\sqrt{2}} P_w(\text{synd},n)$.

4.2 Signatures of Size Two

This section analyzes the aliasing probabilities of a signature consisting of the syndrome plus another coefficient from the Rademacher-Walsh spectrum. Intuitively, one might assume that adding a second coefficient would simply square the result obtained for a single coefficient (the syndrome). This, in fact, would be true if the coefficients were independent. They are not, however, so the actual aliasing probabilities differ from this simplistic view.

Recall that all coefficients in the Rademacher-Walsh spectrum of a given function have the same parity. This immediately eliminates the independent coefficient hypothesis, since one cannot have two coefficients of opposite parity simultaneously. Actually, restricting the syndrome to a

given value puts additional restrictions on the value of other coefficients.

We shall now show the distribution of spectral coefficients resulting from a fixed syndrome and that of the syndrome resulting from a fixed spectral coefficient. These distributions allow the aliasing probabilities for a signature consisting of the syndrome and an additional spectral coefficient to be determined.

Suppose the syndrome value is fixed as r_0. Now, define ρ_0 such that

$$\rho_0 = \min(r_0, 2^n - r_0)$$

Recall that each row of the Hadamard matrix (besides the one for the syndrome) consists of an equal number of +1's and -1's. In a string of length 2^n with weight r_0, there are precisely ρ_0 disjoint pairs of the form {0,1}. For any coefficient r_α there exists a function with weight r_0 such that $r_\alpha = \rho_0$ and a function with weight r_0 such that $r_\alpha = -\rho_0$. These are chosen by arranging the function such that for each pair {0,1} the 1 corresponds to a 1 in the r_α row of the matrix, and the 0 corresponds to a -1, then a coefficient value of ρ_0 for r_α will result. Exchanging the 1's and 0's produces a coefficient value of $-\rho_0$. There is no way to obtain a value $> \rho_0$ or $< -\rho_0$ for r_α, since there are not enough pairs. By exchanging the pairs of 0's and 1's one at a time, however, any value of the same parity as r_0 between $-\rho_0$ and ρ_0 can be obtained.

In other words, for a given r_0, the other coefficients in the Rademacher-Walsh spectrum are fixed in the range:

$$r_\alpha \in A = \left\{ -\rho_0, -\rho_0+2, \ldots, \rho_0-2, \rho_0 \right\}$$

Some points to note:

1) The number of functions with a fixed $r_\alpha \in A$ is: $\left\lfloor \frac{2^{n-1}}{\frac{r_0+r_\alpha}{2}} \right\rfloor \left\lfloor \frac{2^{n-1}}{\frac{r_0-r_\alpha}{2}} \right\rfloor$.

   (Detailed exposition later in this section).

2) All spectral coefficients will lie in the above range for a fixed r_0, therefore it makes no difference which coefficient is being considered.

3) Fixing one of these will have an effect on the others, so while the range applies initially to all coefficients, it will be further restricted when actual values are assigned to particular coefficients.

Relationships between other coefficients and the syndrome may be deduced as well. For instance, if some coefficient r_α has absolute value ρ_α, then we may reverse the previous arguments to obtain a range for the syndrome. There must be at least ρ_α {0,1} pairs in the function, so a syndrome of at least ρ_α and at most $2^n-\rho_\alpha$ is possible, but there could be up to 2^{n-1} such pairs, hence any value of the same parity between the two can

occur.

That is, for functions with r_α fixed as $\pm\rho_\alpha$, for any spectral coefficient, the range of values for r_0 is fixed as:

$$r_0 \in B = \left\{ \rho_\alpha, \rho_\alpha+2, \ldots, 2^n-\rho_\alpha-2, 2^n-\rho_\alpha \right\}$$

Points to note:

1) This restriction on the syndrome guarantees that for any value within the range a function can be chosen with that syndrome.

2) Once a particular function is chosen, however, with a particular coefficient of absolute value ρ_α, the only statement that can be made is that the syndrome of that particular function takes on a value somewhere within the range.

Consider a coefficient r_α with value $\pm\rho_\alpha$. We can establish a range for the syndrome from this, and from that establish a value for r_β. If this is done it becomes clear that no restriction (besides parity) can be placed on r_β. The arguments above cannot ensure that r_α is not in fact the same coefficient as r_β, so even if one is of maximal value, nothing can be said about the other. Because of the additional complexities that arise with extra coefficients, the signatures considered in this thesis will consist only of the syndrome and one other coefficient.

If any spectral coefficient is considered on its own, then the distribution of its values is similar to that for the syndrome. There are $\left| \begin{matrix} 2^n \\ 2^{n-1}-\rho_\alpha \end{matrix} \right|$ functions with a given $r_\alpha = \pm\rho_\alpha$. The worst case value occurs for $\rho_\alpha = 0$, so expression (4.1) holds for any spectral coefficient, not just the syndrome, as far as error detection is concerned.

A new distribution of values for the other coefficients results when the syndrome is fixed. Now given a fixed syndrome, and defining ρ_0 as above, we must determine the number of functions which have a given value of some r_α within the set A defined by ρ_0. Given that $r_\alpha = \rho_\alpha$, where $\rho_\alpha \epsilon A$, and assuming, without loss of generality, that $r_0 = \rho_0$ and ρ_α is non-negative, then it is possible to determine the number of functions which satisfy this criterion.

When calculating spectral coefficients for a function of weight ρ_0, only ρ_0 entries in the Hadamard matrix are summed for each coefficient. To obtain a value of ρ_α for r_α, then there must be ρ_α more +1's than -1's in the ρ_0 values being summed. Hence we have:

Number of +1's : $\dfrac{\rho_0+\rho_\alpha}{2}$

Number of -1's : $\dfrac{\rho_0-\rho_\alpha}{2}$

Total number of +1's and -1's : ρ_0

Value for coefficient : $\frac{\rho_0+\rho_\alpha}{2} - \frac{\rho_0-\rho_\alpha}{2} = \rho_\alpha$

There are 2^{n-1} each of +1's and -1's in the Hadamard matrix for rows which do not correspond with the syndrome. The ρ_0 1's of the function may then be distributed such that $\frac{\rho_0+\rho_\alpha}{2}$ go to one set of 2^{n-1} positions, while the remaining $\frac{\rho_0-\rho_\alpha}{2}$ go to the other set. Hence, the total number of functions with fixed syndrome ρ_0, and a given $r_\alpha = \rho_\alpha$ is equal to:

$$\left(\begin{array}{c} 2^{n-1} \\ \frac{\rho_0+\rho_\alpha}{2} \end{array}\right)\left(\begin{array}{c} 2^{n-1} \\ \frac{\rho_0-\rho_\alpha}{2} \end{array}\right) = \left(\begin{array}{c} 2^{n-1} \\ \frac{r_0+r_\alpha}{2} \end{array}\right)\left(\begin{array}{c} 2^{n-1} \\ \frac{r_0-r_\alpha}{2} \end{array}\right)$$

This then leads immediately to the following statement:

Result 4.4 The aliasing probability for functions with fixed syndrome r_0, and a given fixed r_α is equal to:

$$\frac{\left(\begin{array}{c} 2^{n-1} \\ \frac{r_0+r_\alpha}{2} \end{array}\right)\left(\begin{array}{c} 2^{n-1} \\ \frac{r_0-r_\alpha}{2} \end{array}\right) - 1}{2^{2^n}} \qquad (4.4)$$

The worst case for the distribution implied by expression (4.4) occurs when both the syndrome, r_0, and the other coefficient, r_α, take their worst-case values, hence when $\rho_0 = 2^{n-1}$ and $\rho_\alpha = 0$. Substituting into expression (4.4), this yields:

$$P_w(2coef,n) = \frac{\left|\binom{2^{n-1}}{2^{n-2}}\right|\left|\binom{2^{n-1}}{2^{n-2}}\right| - 1}{2^{2^n}}$$

The $-\frac{1}{2^{2^n}}$ rapidly becomes insignificant, and $2^{2^n} = \left|2^{2^{n-1}}\right|^2$ so we have

Result 4.5: (This result was included in [2], but no general proof was given).

$$P_w(2coef,n) \sim \left|\frac{\left|\binom{2^{n-1}}{2^{n-2}}\right|}{2^{2^{n-1}}}\right|^2 \tag{4.5}$$

Notice that $P_w(2coef,n) \approx (P_w(synd,n-1))^2$. There is an additional factor of $-\frac{1}{2^{2^{n-1}}}$, but this becomes insignificant for $n>4$. When Stirling's approximation is used, the result becomes even clearer:

$$P_w(2coef,n) \sim \left|\left(\pi 2^{n-2}\right)^{-\frac{1}{2}}\right|^2$$

which reduces to

Result 4.6:

$$P_w(2coef,n) \sim \frac{2^{-n+2}}{\pi} = \frac{4}{\pi}2^{-n} \tag{4.6}$$

Thus this signature, consisting of the syndrome and one other spectral coefficient, has aliasing in the worst-case of exactly the same order as the

LFSR, 2^{-n}. In fact, the value reached is nearly the same as that derived by Carter [6], except for the additional discount of $\frac{1}{\pi}$.

This result is the worst case for the aliasing probability of the signature. In the average case, the aliasing probability might be expected to parallel that of the syndrome alone and be lower by some approximately constant factor. Proceeding, then, in the same fashion as was done for the syndrome the following results are obtained:

The total number of functions with a fixed weight r_0 and another fixed coefficient r_α of the same parity as r_0 is:

$$\left\| \frac{2^{n-1}}{\frac{r_0+r_\alpha}{2}} \right\| \left\| \frac{2^{n-1}}{\frac{r_0-r_\alpha}{2}} \right\|$$

Hence, the probability of such a function occurring is:

$$\frac{\left\| \frac{2^{n-1}}{\frac{r_0+r_\alpha}{2}} \right\| \left\| \frac{2^{n-1}}{\frac{r_0-r_\alpha}{2}} \right\|}{2^{2^n}}$$

The probability of such a function aliasing is consequently, from expression (4.4),

$$\frac{\left\| \frac{2^{n-1}}{\frac{r_0+r_\alpha}{2}} \right\| \left\| \frac{2^{n-1}}{\frac{r_0-r_\alpha}{2}} \right\| - 1}{2^{2^n}}$$

Of course, since only values of r_α which have the same parity as r_0 are permitted, it is necessary to adapt the expression somewhat, to allow only those values which can actually occur to be summed. By performing this summation over all possible weights and values for the other coefficient we obtain:

$$P_{av}(2coef,n) = \frac{1}{2^{2^{n+1}}} \sum_{i=0}^{2^n} \sum_{j=0}^{a} \left| \binom{2^{n-1}}{j} \binom{2^{n-1}}{a-j} \right| \left| \binom{2^{n-1}}{j} \binom{2^{n-1}}{a-j} - 1 \right|$$

where $a = \min(i,2^n-i)$. Notice that the value a may be eliminated by performing the sum over i up to 2^{n-1}, doubling the result, and subtracting the value that occurs at 2^{n-1}. Thus, we have

Result 4.7:

$$P_{av}(2coef,n) = \frac{2}{2^{2^{n+1}}} \sum_{i=0}^{2^{n-1}} \sum_{j=0}^{i} \left| \binom{2^{n-1}}{j} \binom{2^{n-1}}{i-j} \right| \left| \binom{2^{n-1}}{j} \binom{2^{n-1}}{i-j} - 1 \right| - \frac{1}{2^{2^{n+1}}} \sum_{j=0}^{2^{n-1}} \left| \binom{2^{n-1}}{j} \binom{2^{n-1}}{2^{n-1}-j} \right| \left| \binom{2^{n-1}}{j} \binom{2^{n-1}}{2^{n-1}-j} - 1 \right| \quad (4.7)$$

Of course, expression (4.7) is too complex to be exactly calculated for large n. In section 4.4 an approximate expression is derived which facilitates the calculation of average case aliasing for a signature consisting of the syndrome and an additional spectral coefficient.

4.3 Pseudo 4-Valued Testing

We now introduce another testing method, pseudo 4-valued testing, which is based on syndrome testing, and examine its aliasing probabilities. In this method, the output stream is not considered to be a binary stream of length 2^n, but rather a 4-valued stream of length 2^{n-1}. In other words, every second bit in the output stream is given weight 2. These values are then summed to produce a pseudo 4-valued syndrome.

For example, consider the following function output:

1001101110100100

As a binary function, it has weight 8. Now consider the same function as being 4-valued:

10 01 10 11 10 10 01 00

which is equivalent to

2 1 2 3 2 2 1 0

which has a 4-valued weight of 13.

The 4-valued syndrome can also be considered to be an example of a linear combination of spectral coefficients, since the value obtained for it is identical to

$$\frac{1}{2}(3r_0 + r_1)$$

There are many such linear combinations possible, but the 4-valued syndrome has the advantage of being easily understood. Since only the single linear combination is considered here, (although the signature of the previous section can be thought of as $2^n r_\alpha + r_0$) the following problem remains.

Open Problem: What is the "best" linear combination of coefficients to use, in terms of fault coverage versus tester cost?

The range of the 4-valued syndrome is clearly larger than that of its binary equivalent. The smallest possible syndrome is still 0, while the largest becomes $2*2^{n-1}+2^{n-1} = 3*2^{n-1}$, rather than 2^n for the binary syndrome. Thus the 4-valued syndrome requires an additional bit of storage, as well as the extra hardware it needs to perform the double weighting.

The distribution of the 4-valued syndrome is symmetric, and since its range is larger than that of the binary syndrome, one might expect that the aliasing probability is lower. In fact, this turns out to be the case. In order to derive the aliasing probability, some facts must be noted.

First of all, there are often numerous ways to obtain the same 4-valued syndrome, with different binary weights. For instance, both the following have 4-valued syndrome of 6: 11110000, 11010101, while their binary weights are 4 and 5 respectively.

A second point is that, while the range extends to $3*2^{n-1}$, any 4-valued syndrome with a value greater than 2^{n-1} will require some doubly-weighted 1's. Hence, to obtain a 4-valued syndrome of, say, s_4, the minimum number of doubly-weighted 1's is:

$$w_1 = \max\left(0, \left|\frac{s_4 - 2^{n-1}}{2}\right|\right)$$

Similarly, the maximum number of doubly weighted 1's is:

$$w_2 = \min\left(2^{n-1}, \left|\frac{s_4}{2}\right|\right)$$

Now, if there are k doubly-weighted 1's (out of a maximum possible number of 2^{n-1}), then there must be $s_4 - 2k$ singly weighted 1's (again out of 2^{n-1}). This allows us to write the following expression:

Result 4.8: The total number of functions with 4-valued syndrome s_4.

$$\sum_{k=w_1}^{w_2} \binom{2^{n-1}}{k}\binom{2^{n-1}}{s_4-2k} = \sum_{k=0}^{2^{n-1}} \binom{2^{n-1}}{k}\binom{2^{n-1}}{s_4-2k} \qquad (4.8)$$

The worst-case aliasing probability for a 4-valued syndrome occurs with the midrange value of $3*2^{n-2}$, because of the symmetric nature of the distribution. Substituting $s_4 = 3*2^{n-2}$ into expression (4.8) yields the total number of functions with this syndrome.

$$\sum_{k=0}^{2^{n-1}} \binom{2^{n-1}}{k} \binom{2^{n-1}}{3*2^{n-1}-2k}$$

or, equivalently

$$\sum_{k=0}^{2^{n-2}} \binom{2^{n-1}}{k+2^{n-3}} \binom{2^{n-1}}{2k}$$

The worst-case aliasing probability then follows immediately as:

Result 4.9:

$$P_w(4val,n) = \frac{1}{2^{2^n}} \left| \sum_{k=0}^{2^{n-2}} \binom{2^{n-1}}{k+2^{n-3}} \binom{2^{n-1}}{2k} - 1 \right| \qquad (4.9)$$

The average case probability for the four-valued syndrome may be computed in essentially the same manner as expression (4.7) was derived from expression (4.6). Consider expression (4.8), the number of functions with a given four-valued weight s_4. The total number of functions of n inputs is 2^{2^n}, so the probability of a function of weight s_4 occurring is

$$\frac{\sum_{k=0}^{2^{n-1}} \binom{2^{n-1}}{k} \binom{2^{n-1}}{s_4-2k}}{2^{2^n}}$$

This allows us to compute the probability of the function aliasing, which is:

$$\frac{\left| \sum_{k=0}^{2^{n-1}} \binom{2^{n-1}}{k} \binom{2^{n-1}}{s_4-2k} \right| \left| \sum_{k=0}^{2^{n-1}} \binom{2^{n-1}}{k} \binom{2^{n-1}}{s_4-2k} - 1 \right|}{2^{2^{n+1}}}$$

Finally, by summing over all possible values for the syndrome, we obtain

Result 4.10: The average case aliasing for the four-valued syndrome under the "all equally likely" assumption is:

$$P_{av}(4val,n) = \frac{1}{2^{2^{n+1}}} \sum_{j=0}^{3*2^{n-1}} \left\{ \left| \sum_{k=0}^{2^{n-1}} \binom{2^{n-1}}{k} \binom{2^{n-1}}{j-2k} \right| \left| \sum_{k=0}^{2^{n-1}} \binom{2^{n-1}}{k} \binom{2^{n-1}}{j-2k} - 1 \right| \right\} \tag{4.10}$$

Again, exact calculation of expression (4.10) is infeasible for large n. For this reason, an approximation is made in the next section to enable comparison between the average case aliasing of the four-valued syndrome and that of other signature methods.

4.4 Numerical Values for Previous Expressions

The expressions derived earlier in the chapter for various aliasing probabilities are now tabulated for some values of n, in order to show more concretely how they compare with one another. It was mentioned earlier that exhaustive testing is infeasible for circuits with more than about 20 inputs, because of the time required to cycle through all input

combinations. Improvements in technology could increase the speed at which devices can operate, although improvement much beyond 10GHz (10^{10} operations per second) seems very unlikely due to restrictions imposed by the speed of light. At a rate of 1GHz, approximately 2^{30} test vectors can be applied in 1 second, so the absolute upper bound on exhaustive testing is probably just over 30 inputs. At present, exhaustive testing for circuits of more than about 25 inputs is impractical. The range of interest, then, for the various testing methods is circuits of fewer than 35 inputs. Hence, the tables in this section include only values within this range.

When calculating aliasing probabilities, the exact value for a particular expression is used whenever possible, but when its computation becomes unreasonable, an approximation to it is made. (For instance, expression (4.2) approximates expression (4.1) to a high degree for $n > 10$). In cases where no approximation can be derived theoretically, numerical values are extrapolated from those which can be calculated exactly. For example, P_{w}(4val,n), has no readily discernible approximation, but a clear pattern exists to the exact values computed.

4.4.1 Syndrome Testing

The first set of numerical values calculated are those for syndrome counting. Table 4.1 outlines the aliasing probability for both the syndrome and the linear feedback shift register. The expressions used for calculating the worst case aliasing were:

$$P_w(\text{synd},n) = \frac{\left\lfloor \frac{2^n}{2^{n-1}} \right\rfloor - 1}{2^{2^n}} \tag{4.1}$$

for $n \leqslant 18$, and

$$P_w(\text{synd},n) \approx (\pi 2^{n-1})^{-\frac{1}{2}} \tag{4.2}$$

for $n > 18$. In the average case:

$$P_{av}(\text{synd},n) \approx (\pi 2^n)^{-\frac{1}{2}} \tag{4.3}$$

For the LFSR:

$$P_w(\text{LFSR},n) = P_{av}(\text{LFSR},n) = \max(2^{-n}, 2^{-16}) \tag{3.5}$$

This reflects the aliasing predicted by Carter [6] for a 16-bit LFSR. Note that for a shift register there is no difference between average-case and worst-case results.

From Table 4.1 it can be seen that $P_w(\text{synd},n) > P_w(\text{LFSR},n)$ for $n \leqslant 31$. As stated earlier, exhaustive testing is not practical on circuits

with many more than 20 inputs. Thus, it can be seen that theoretically, in the worst case, (using the "all equally likely" assumption) syndrome counting is not as good as shift register testing for circuits which can reasonably be tested exhaustively. This stems from the supposition that lower aliasing probability implies a better testing technique. In a later section some actual circuits will be examined in order to determine how well the theory stands up in practice.

In the average case as well, syndrome testing is seen to have higher aliasing probability for all $n \leqslant 30$. So once again LFSR testing is theoretically better than syndrome testing in terms of error coverage, (again, not necessarily fault coverage), for circuits testable exhaustively.

There are two items worth noting at this point, however. First, syndrome testing gains lower aliasing probability than LFSR testing at about n=31 because the size of the shift register is fixed at 16. If the shift register was permitted to be at least n bits long, then Carter's formula would give aliasing probability of:

$$P_{av}(\text{LFSR},n) = 2^{-n}$$

in an exhaustive test. In this case, $P_{av}(\text{LFSR},n) < P_{av}(\text{synd},n)$ for all n, while the testing circuit sizes would remain approximately equal. The 16-bit value for the LFSR was chosen because that is the size implemented by Hewlett-Packard.

The second point to note is that even though the LFSR has smaller aliasing probability, after a certain point, both have probability that may be considered "small enough" for many applications. For instance, with n=20, the probability of a syndrome counter detecting an error is 99.94%, while the probability of an LFSR detecting an error is 99.998%. Both of these values are considerably higher than the typical percentage of faults detected by a test set. (Recall Savir's claim in [31] that a test set should cover 98% of the faults with 99.9% probability). So, if the error coverage rates claimed theoretically transfer into actual fault coverage, then both syndrome testing and LFSR testing are a significant improvement over the classical test set method.

4.4.2 Signatures of Size Two

There are numerous spectral signatures which consist of two coefficients. Here we consider only those which contain the syndrome, because of its coverage of unate circuits. Similar analysis could be performed on other signatures, but is not included here.

The next table, Table 4.2, compares the aliasing probability of a signature consisting of the syndrome and an additional spectral coefficient with that for an LFSR. Expression (3.5) is used for P_w(LFSR,n) and P_{av}(LFSR,n), while expression (4.6) gives P_w(2coef,n),

and expression (4.7) gives $P_{av}(2coef,n)$ for $n \leq 8$. For the remaining values of n in the table, it was noted that $P_{av}(2coef,n)$ was progressing asymptotically towards one half of $P_{w}(2coef,n)$. This progression was extrapolated to obtain the average-case aliasing for the other values of n because of the computational infeasibility of evaluating expression (4.7) for $n > 8$. The values obtained, then, are not necessarily entirely accurate, but are within the correct range and in all likelihood, accurate within ±10%.

It is important to note that the other spectral coefficient used in these expressions must be chosen at random. If any heuristic, such as largest first-order coefficient, is used to choose the coefficient, the resulting aliasing probabilities may be different. Another way to say this is that the coefficient to be used must be chosen *a priori* for the aliasing probabilities stated here to apply. This relates to the fact, mentioned in chapter 2, that there is at present no method of classifying binary functions for $n > 5$ so that one may determine, for instance, what the largest magnitude coefficient will be.

The results of Table 4.2 show that theoretically, based on the "all equally likely" assumption, $P_{w}(2coef,n) \approx 1.27 * P_{w}(LFSR,n)$. In the average case, the aliasing probability for the new signature is actually less than that for the LFSR, $P_{av}(2coef,n) \approx 0.64 * P_{av}(LFSR,n)$. This is true for all $n \leq 16$, assuming a 16-bit shift register, but if an n-bit LFSR was used,

the results would hold for all n. The conclusion that can be drawn from these data is not that the new signature is better than the LFSR in the average case, but rather that the magnitude of their aliasing probabilities under the "all equally likely" assumption is essentially equal. As an example, for n=11, the LFSR has error coverage of 99.95%, while the new signature has worst case error coverage of 99.94%, and average case coverage of 99.97%.

The only disadvantage of the new signature is the additional hardware needed to compute it. To compute a first order coefficient (without loss of generality we may assume the coefficient is r_n), an n-1 bit counter is required. Two successive syndrome tests are performed on a partition of the input about x_n, giving weights m1 and m2. These two weights uniquely determine both r_0 and r_n: i.e.

$$r_0 = m1 + m2, \text{ and } r_n = m1 - m2$$

Hence, the signature can be determined by m1 and m2, so reference values for these two must be stored. It is the additional storage required for the two reference values (2n-2 bits versus n bits) that makes the tester for the new signature more complex than a syndrome tester, and hence an LFSR tester. Nonetheless, the testing hardware for the new signature is still compact enough to be included on a chip and retains the design for testability characteristics of the syndrome tester.

In the next chapter, the new signature is tested in comparison with the LFSR. This assists in deciding whether or not the two methods are as comparable in fault coverage as they are in error coverage.

4.4.3 Four-Valued Syndrome Testing

Table 4.3 gives the aliasing probability of the four-valued syndrome and compares it with that for an LFSR for $n < 30$. Expression (3.5) is again used for the aliasing probability of the LFSR in all cases, while expression (4.9) gives $P_w(4val,n)$ and expression(4.10) gives $P_{av}(4val,n)$ (for small n).

For $n > 10$, it was computationally infeasible to evaluate expression (4.9). It was noted, however, that the values obtained for $n \leqslant 10$ followed a pattern, enabling the approximation below to be used for the remainder of the values:

Result 4.11: (approximate aliasing, 4-valued syndrome, worst case)

$$P_w(4val,n) \approx 0.505 \left| 2^{-\frac{n}{2}} \right| \qquad (4.11)$$

Expression (4.10) is also too complex to be evaluated for large n. Again, it was noted that the values obtained for $n < 9$ had a pattern to them. In this case, expression (4.10) gave a value of approximately 0.443 times that of expression (4.9). So the second approximation used is:

Result 4.12: (approximate aliasing, 4-valued syndrome, average case)

$$P_{av}(4val,n) \approx 0.443 * P_{w}(4val,n) \qquad (4.12)$$

Once again, these approximations mean that the values in Table 4.3 may not be accurate for $n \geq 9$. It is assumed, however, because of the good approximation expression (4.11) gives for the known values that it continues to apply for the unknown values. Similarly, the extrapolated average case (4.12) is assumed correct within some margin of error, probably less than an order of magnitude. In any case, Table 4.3 permits the observation of a general trend, not immediately obvious from the expressions of chapter 4.

This trend duplicates the pattern observed for syndrome testing: that is,

$$P_{w}(4val,n+1) \approx \frac{1}{\sqrt{2}} P_{w}(4val,n)$$

and

$$P_{av}(4val,n+1) \approx \frac{1}{\sqrt{2}} P_{av}(4val,n)$$

while

$$P_{w}(LFSR,n+1) = \frac{1}{2} P_{w}(LFSR,n)$$

As a result, the four valued-syndrome's aliasing probability (under the "all equally likely" assumption) decreases more slowly than that of

the LFSR. The LFSR has substantially lower aliasing probability than the 4-valued syndrome in both average and worst cases, but both methods are likely good enough. For example, given n=12, the 4-valued syndrome has worst-case error coverage of 99.2% (average case 99.6%) while the LFSR has coverage of 99.99%. It is likely that these coverage values are adequate for most applications (again recall Savir's 98% figure for test sets [31]). On the other hand, 99% coverage may not be enough for low-yield processes, so 99.99% may be more suitable. In any case, higher error coverages are achieved by larger n with both methods.

The hardware required to calculate a four-valued syndrome is more complex than that needed for a standard syndrome. Consequently, it is useful to determine whether the benefits of additional coverage outweigh the costs of the larger tester. Comparing the first column of Tables 4.1 and 4.3 shows that

$$P_w(4val,n) \approx 0.63 * P_w(synd,n)$$

In the average case, the ratio is

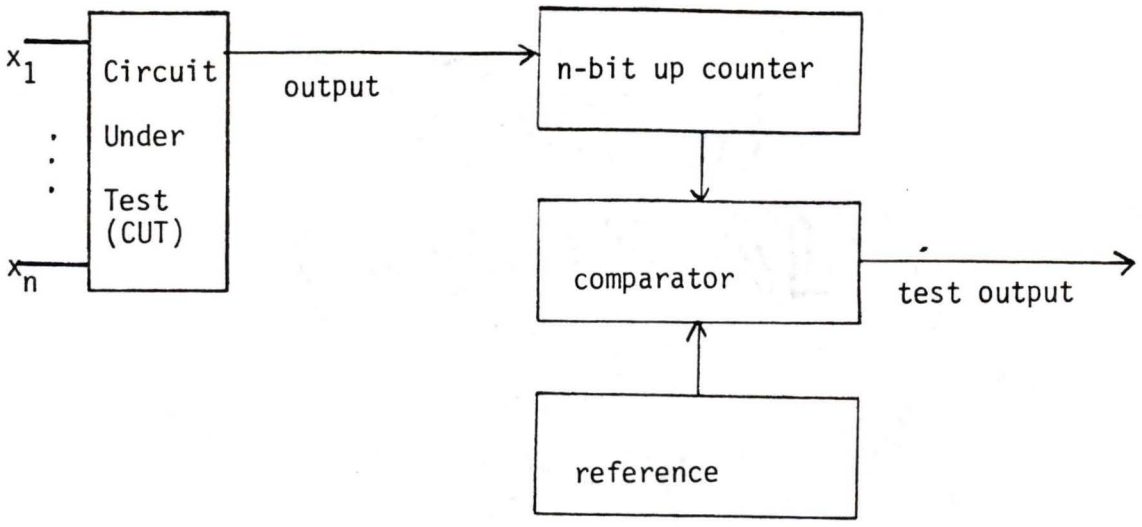
$$P_{av}(4val,n) \approx 0.40 * P_{av}(synd,n)$$

(comparing the second column of the two tables). So it can be seen that while the 4-valued syndrome has better error coverage than the binary syndrome, the difference between the two is a constant factor, rather than the squaring figure observed for the two coefficient signature of the

previous section.

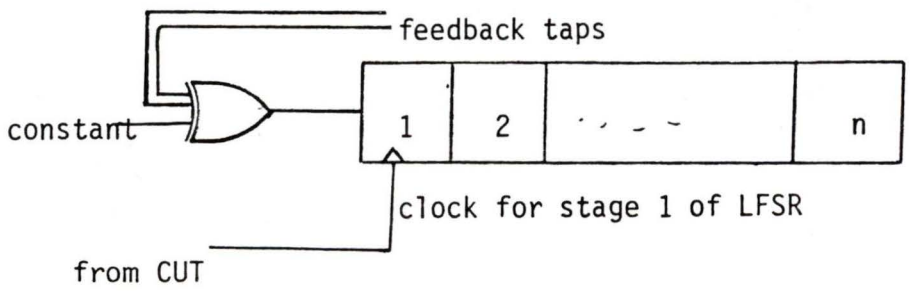
The additional hardware required for a 4-valued syndrome test is some weighting circuitry and an additional bit of storage, hence a constant amount. It is unclear whether or not the circuitry is simple enough to warrant its use. Very probably this will depend on the method used to implement the double weighting.

In the following chapter, the various signatures analyzed so far are used in actual circuits, and the coverages for these empirical tests are compared. In chapter 6, conclusions are drawn regarding the relative merits of the signatures examined here.



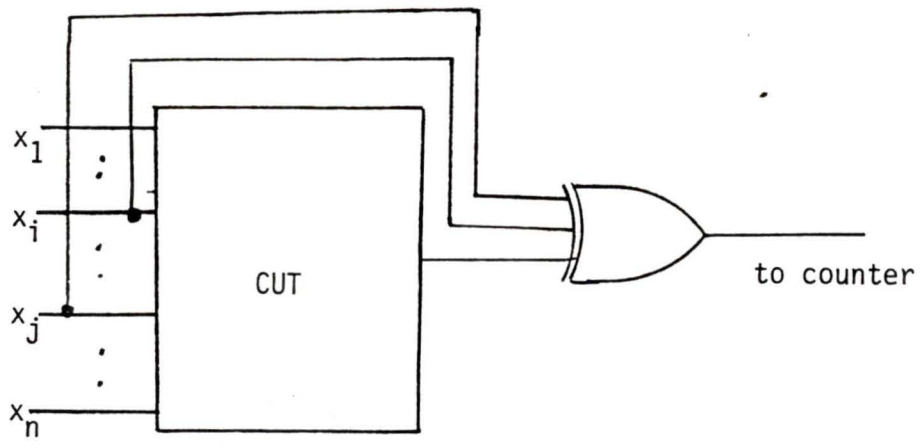
Syndrome Tester

Figure 4.1



Use of Autonomous LFSR as counter

Figure 4.2



Tester for R_{ij}

Figure 4.3

TABLE 4.1

Syndrome and LFSR Aliasing Probability

	Syndrome		LFSR
n	Worst Case	Average Case	All Cases
1	0.25	0.39894228	0.5
2	0.3125	0.28209479	0.25
3	0.26953125	0.19947114	0.125
4	0.19636536	0.14104740	0.0625
5	0.13994993	0.09973557	0.03125
6	0.09934675	0.07052370	0.015625
7	0.07038609	0.04986779	0.0078125
8	0.04981911	0.03526185	0.00390625
9	0.03524464	0.02493389	0.00195313
10	0.02492781	0.01763092	0.00097656
11	0.01762877	0.01246695	0.00048828
12	0.01246619	0.00881546	0.00024414
13	0.00881519	0.00623347	0.00012207
14	0.00623338	0.00440773	0.00006104
15	0.00440770	0.00311674	0.00003052
16	0.00311672	0.00220387	0.00001526
17	0.00220386	0.00155837	0.00001526
18	0.00155837	0.00110193	0.00001526
19	0.00110193	0.00077918	0.00001526
20	0.00077918	0.00055097	0.00001526
21	0.00055097	0.00038959	0.00001526
22	0.00038959	0.00027548	0.00001526
23	0.00027548	0.00019480	0.00001526
24	0.00019480	0.00013774	0.00001526
25	0.00013774	0.00009740	0.00001526
26	0.00009740	0.00006887	0.00001526
27	0.00006887	0.00004870	0.00001526
28	0.00004870	0.00003444	0.00001526
29	0.00003444	0.00002435	0.00001526
30	0.00002435	0.00001722	0.00001526
31	0.00001722	0.00001218	0.00001526
32	0.00001218	0.00000861	0.00001526
33	0.00000861	0.00000609	0.00001526
34	0.00000609	0.00000430	0.00001526
35	0.00000430	0.00000304	0.00001526

TABLE 4.2

Signature of 2 Coefficients and LFSR Aliasing Probabilities

	2 Coefficient Signature		LFSR
n	Worst Case	Average Case	All Cases
1	0.0	0.0	0.5
2	0.25	0.03125	0.25
3	0.140625	0.04504395	0.125
4	0.07476807	0.02810416	0.0625
5	0.03856534	0.01575515	0.03125
6	0.01958598	0.00849124	0.015625
7	0.00986978	0.00446250	0.0078125
8	0.00495419	0.00230734	0.00390625
9	0.00248193	0.00116650	0.00195313
10	0.00124214	0.00059001	0.00097656
11	0.00062135	0.00032982	0.00048828
12	0.00031077	0.00015540	0.00024414
13	0.00015540	0.00007771	0.00012207
14	0.00007771	0.00003886	0.00006104
15	0.00003886	0.00001943	0.00003052
16	0.00001943	0.00000971	0.00001526
17	0.00000971	0.00000486	0.00001526
18	0.00000486	0.00000243	0.00001526
19	0.00000243	0.00000121	0.00001526
20	0.00000121	0.00000060	0.00001526

TABLE 4.3

4-Valued Syndrome and LFSR Aliasing Probabilities

	4-Valued Syndrome		LFSR
n	Worst Case	Average Case	All Cases
3	0.15625	0.07476807	0.125
4	0.12304688	0.05631289	0.0625
5	0.08825659	0.03856535	0.03125
6	0.06274294	0.02760012	0.015625
7	0.04448457	0.01958598	0.0078125
8	0.03149726	0.01390935	0.00309625
9	0.02228674	0.00986978	0.00195313
10	0.01576434	0.06983603	0.00097656
11	0.01114892	0.00495420	0.00024414
12	0.00789065	0.00349556	0.00012207
13	0.00557953	0.00247173	0.00006104
14	0.00394532	0.00174778	0.00003052
15	0.00278977	0.00123587	0.00001526
16	0.00197266	0.00087388	0.00001526
17	0.00139488	0.00061704	0.00001526
18	0.00098633	0.00043694	0.00001526
19	0.00069744	0.00030897	0.00001526
20	0.00049317	0.00021847	0.00001526
21	0.00034872	0.00015448	0.00001526
22	0.00024658	0.00010923	0.00001526
23	0.00017436	0.00007724	0.00001526
24	0.00012329	0.00005462	0.00001526
25	0.00008718	0.00003862	0.00001526
26	0.00006165	0.00002731	0.00001526
27	0.00004359	0.00001931	0.00001526
28	0.00003082	0.00001365	0.00001526
29	0.00002180	0.00000966	0.00001526

Chapter 5

This chapter consists of two parts. In the first part, an experiment is conducted based on a set of 37 circuits to examine the validity of the "all equally likely" assumption in a particular instance. In the second part, we investigate the actual aliasing of the various signatures discussed in the earlier chapters for a sample set of 44 circuits.

5.1 Analysis of Parity of Fault Effects

When examining the functions which result from stuck-at faults within actual circuits, it becomes clear that functions of even weight occur much more often than those of odd weight. This appears to contradict the predictions of the "all equally likely" assumption, which has been used throughout the theoretical portion of this thesis to obtain aliasing probabilities. The following experiment examines the validity of the assumption in the particular instance of predicting the parity of fault effects.

It is surmised that the assumption fails in this aspect on particular classes of faults. The purpose of the experiment is to identify these classes, as well as others where the assumption predicts the distribution correctly. In all, eleven classes are defined, and the parity of their fault effects is determined. The classes are briefly outlined in Table 5.1 (more detailed

definitions are provided later).

Table 5.1

Description of Fault Classes

Class	Description
0	entire sample
1	faults on odd-weighted circuits
2	faults on even-weighted circuits
3	non-input faults
4	faulty functions dependent on all inputs
5	faults on "positive" lines
6	faults on "negative" lines
7	faults on "candidate" lines
8	non-equivalent faults
9	class 8 $\cap$ class 4
10	class 9 $\cap$ class 1

An initial analysis leads to the conjecture that classes 4,9, and 10 should follow the parity predictions of the "all equally likely" assumption, while classes 0 through 3 and 8 should not. No hypotheses regarding the behaviour of classes 5 through 7, which relate to the syndrome-testability of lines, could be formed prior to the experiment.

Before any results are presented, it is necessary to give the following definitions:

Definition 5.1: A fault is said to be an *odd fault* if its fault effect causes the resulting function to have odd weight.

Similarly,

Definition 5.2: A fault is said to be an *even fault* if its fault effect causes the resulting function to have even weight.

Note that functions with odd weight are often referred to as odd functions, and functions with even weight are referred to as even functions.

Lemma 5.3: The proportion of functions of n variables with odd weight is given by:

$$\frac{\sum_{j=0}^{2^{n-1}-1} \binom{2^n}{2j+1}}{2^{2^n}} = \frac{1}{2} \tag{5.3}$$

This value sums the total number of functions with each odd weight.

Lemma 5.4: The proportion of even functions of n variables is given by subtracting expression (5.3) from one, and is thus also $\frac{1}{2}$.

Therefore, the "all equally likely" assumption suggests that one half of the faults will produce odd-weighted outputs. We now examine a set of faults. If the assumption is valid in its prediction of fault parity, we would expect the proportions of odd and even faults to be roughly equal.

Thirty-seven "typical" circuits, ranging in number of inputs from 3 to 12, were examined. Sixteen implemented odd functions, while the

remaining twenty-one implemented even functions. These circuits were all combinational and implemented everything from full adders to parity trees to two-out-of-five checkers. These circuits are numbers 1-19, 24-27, and 31-44 in Table 5.7. The other circuits in this table were generated by the author and not included in the sample because of possible bias. The circuits were simulated in the fault-free state, as well as for each single stuck-at fault. No fault collapsing (where several faults in a given circuit with identical fault effects are considered to be the same fault) or other simplification was performed (except for Classes 8,9, and 10). In all, 1310 faults were analyzed. These faults are subsequently referred to as Class 0, or the set which includes all faults.

The distribution of faults within Class 0 is as follows:

- 217 odd faults
- 1093 even faults

That is, 16.6% of the fault effects produced odd functions, while 83.4% were even. These percentages differ substantially from a 50-50 split, using the "all equally likely" assumption. They confirm the preponderance of even faults. It was decided to analyze the faults further in order to classify them as to the tendency toward even or odd parity. One possible such categorization is those faults which result from initially odd and initially even functions. It was conjectured that this division

would not affect fault parity distribution.

Definition 5.5: Class 1 is the set of faults which occur in circuits that have odd-weighted output functions in the fault-free state.

Definition 5.6: Class 2 is the set of faults which occur in circuits that have even-weighted output functions in the fault-free state.

Clearly the classes partition Class 0: i.e., Class 1 $\bigcup$ Class 2 = Class 0, and Class 1 $\bigcap$ Class 2 = ϕ. This partition divides the odd and even faults as follows:

Table 5.2

Distribution of Odd/Even Faults
Classes 1 and 2

class	number		percentage	
	odd	even	odd	even
1	148	597	19.9	80.1
2	69	496	12.2	87.8
total	217	1093	16.6	83.4

It can be seen from the above table that classifying faults by the parity of the fault-free function does not significantly alter the distribution of even and odd faults. It is interesting that the proportion of odd faults in Class 2 is lower than the average (Class 0), although this may simply be due to experimental error.

It was hypothesized that the classes defined below would lean less heavily towards even faults. It was suspected that the proportions of even and odd faults in class 4 would be roughly equal, although class 3 was expected to have more even faults. These conjectures were made because it can be shown that faults excluded from each class can have only even weight. This is demonstrated below.

Definition 5.7: Class 3 is the set of stuck-at faults which occur on lines x, such that x is not an input line to the circuit.

Definition 5.8: Class 4 is the set of stuck-at faults which do not cause the output function of the circuit to become vacuous in terms of one or more inputs.

Some initial observations on these classes are:

1) Class 4 is a proper subset of Class 3. Class 3 always includes the output line to a circuit, while Class 4 never does. Further, any fault in Class 4 is also in Class 3, since an input fault makes the circuit vacuous in terms of that input.

2) No relationship can be specified between Classes 3 and 4, and Classes 1 and 2.

3) In a fanout-free circuit, Class 4 will be empty.

The faults in Classes 3 and 4 were examined, and the resulting parities are documented in Table 5.3 below.

Table 5.3

Distribution of Odd/Even Faults
For Classes 3 and 4

Fault Class	number		percentage	
	odd	even	odd	even
3	217	718	23.2	76.8
4	217	423	33.9	66.1

From the above table, we note the following:

1) As expected, the complement of Class 3 (or co-Class 3), input faults, contains only even faults. This can be seen by comparing the number of odd faults in Class 3 to the number in Class 0.

2) Similarly, the complement of Class 4 (co-Class 4), those faults which make the circuit vacuous in one or more inputs, contains only even faults.

3) In both classes even faults greatly outnumber odd faults, although the percentages for class 4 are considerably closer to the 50-50 ratio predicted by the "all equally likely" assumption.

4) Because of the number of faults involved, it is hypothesized that the results are beyond statistical error.

Points 1 and 2 are easily explained. In an exhaustive test, any fault which does not depend on all inputs (co-Class 4, which includes co-Class 3) results in a function of even weight, because the input sequence to the circuit is repeated. Hence, if a circuit is vacuous in input x_1, say, the input set $\{0,x_2, \ldots, x_n\}$ is identical to the set $\{1,x_2, \ldots, x_n\}$ as far as the circuit's output function is concerned. This notion can be extended. If the circuit becomes vacuous in two inputs, then the resulting function has weight that is 0 mod 4, and so on. One direct consequence is that any fault in a fanout-free circuit will always have even weight. A corollary is that any odd-weighted fanout-free circuit is testable by any spectral coefficient, including the syndrome.

Because the faults missing from Class 4 and included in Class 3 are all even, it is reasonable that Class 3 leans heavily towards even faults. It is unclear why such a large percentage of Class 4 is even. One possible explanation is that the portions of the circuit which can produce odd functions may have been made vacuous in an input while the circuit as a whole is not, due to reconvergent fanout. This conjecture was not investigated, since it involves checking values occurring on internal lines in a circuit.

This second classification scheme still fails to closely mimic the results predicted by the "all equally likely" assumption. Additional schemes to analyze the faults were sought. One such method is that used

to classify lines for syndrome testing [22]. In this method, lines are marked as to their inversion parity, or the parity of the number of inverters on every path between the line and the output. In the absence of fanout each line can be marked as either positive or negative, depending on whether the number of inverters between the line and the output is even or odd. In the case of reconvergent fanout, where branches have different inversion parity, and for exclusive-or gates, the lines are marked as "candidate" lines (see figure 5.1 for an example). If a path between a line and an output includes a candidate line, the line is also a candidate line. The only lines which are not guaranteed to be syndrome testable are the candidate lines. It was thought that perhaps a relationship existed between syndrome testability and the odd or evenness of the function resulting from a stuck-fault on a line.

Definition 5.9: Class 5 is the set of faults on positive lines, Class 6 is the set of faults on negative lines, and Class 7 is the set of faults on candidate syndrome-untestable lines.

Classes 5,6, and 7 partition Class 0. The sample set of faults was partitioned into the three new classes, and the following results were obtained:

Table 5.4

Distribution of Odd/Even Faults
Based on Syndrome Testability

class	number		percentage	
	odd	even	odd	even
5	125	583	17.7	82.3
6	85	196	30.2	69.8
7	7	314	2.2	97.8
total	217	1093	16.6	83.4

We may make some observations from the above table.

1) The percentages for positive faults are about the same as the average.

2) Those for negative faults are above average.

3) Those for candidate lines are well below average.

The very small number of odd faults in the candidate untestable lines may perhaps be due to the nature of faults on such lines. This possibility is explored below.

Many candidate untestable lines eventually either reach an exclusive-or gate, or some circuitry which mimics such a gate. Suppose lines g_1 and g_2 meet at an exclusive-or gate, whose output is g_3. The weight of g_3 is given by:

$$W(g_3) = W(g_1) + W(g_2) - 2W(g_1g_2)$$

where W(F) is the weight of F, and g_1g_2 is g_1 AND g_2. It is clear from the above expression that g_3 can have odd weight only when exactly one of its

input functions has odd weight. As seen before, a function has odd weight only when it depends on all the inputs. Therefore, in order for g_3 to be odd, it must modify a function of all inputs by taking its exclusive-or with some even function. Exclusive-or gates are expensive, in terms of the silicon area required to implement them, so using one as a modifier may not be the most cost-efficient realization of a function. In practice, exclusive-or constructions are almost always used to achieve functions of even weight. In many cases, the capacity to produce functions of odd weight is nonexistent, because the output is the only line which depends on all inputs, so odd faults are almost impossible.

The higher proportion of negative functions with faults of odd weight arises because several of the circuits checked were NAND or NOR implementations. A large number of the lines were close to the output, depended on all inputs, and of negative parity. It seems to be these lines which were responsible for the relatively large number of odd-weighted faults. This is demonstrated below.

The weight of AND and OR functions can be defined in a similar manner to that previously given for exclusive or:

$$W(g_1 + g_2)+W(g_1g_2) = W(g_1) + W(g_2)$$

where $g_1 + g_2$ refers to the logical OR of the two functions.

Note that the parity of a function's complement is the same as that of the original function, so the inclusion of inverters in a circuit does not affect its parity.

It can be seen that as long as the output of an AND or an OR gate depends on all inputs, its weight can be odd. This differs from the exclusive-or gate, where the inputs to the gate had to be functions of all inputs. So while a circuit constructed entirely of exclusive-or gates can only have even weight, one constructed of AND and OR gates can have either even or odd weight.

The expressions above seem to indicate that when all faults which cause a circuit to become vacuous in an input are eliminated the distribution of odd and even faults should be about even. Table 5.3 has shown that this is not so. A possible explanation is that equivalent faults were not considered in the results of Table 5.3. The results are now re-analyzed, with fault collapsing performed. Two new classes are defined:

Definition 5.10: Class 8 is the set of faults remaining after fault collapsing has been performed on Class 0.

Definition 5.11: Class9 = Class8 $\bigcap$ Class4.

Class 9 is the set of all faulty functions that depend on all inputs, after fault collapsing. The faults in these classes were examined as to

their parity. The results are reported below in Table 5.5.

Table 5.5

Distribution of Odd/Even Faults
After Fault Collapsing

class	number		percentage	
	odd	even	odd	even
8	142	717	16.5	83.5
9	142	243	36.8	63.2

We may observe the following from the table:

1) There is no significant change in the proportion of even and odd faults between faults in Class 8, and those in Class 0.

2) The percentage of odd faults in Class 9 is slightly higher than that in Class 4 (36.8% to 33.9%).

3) Both classes consist primarily of even faults.

There is no immediately obvious reason why the faults in Class 9 do not conform to the expected 50-50 ratio predicted by the "all equally likely" assumption. An examination of the data shows that 90 of the even faults in Class 9 result from just 3 circuits. Without these circuits, the proportions of even and odd faults in the sample become much closer to the expected ratio:

- 142 odd faults (48.1%)
- 153 even faults (51.9%)

If these circuits could be shown to be some sort of aberration, then it could be concluded that Class 9 contains faults which conform to the parity predictions of the "all equally likely" assumption.

The circuits removed are numbers 27, 34 and 35 (see Table 5.7). These circuits do not seem to have any characteristics which separate them from the others, except for the fact that 27 and 34 are almost entirely candidate syndrome-untestable. Circuit 35 differs from 34 by only a single gate, so together they may provide some skewing. The three circuits do have one thing in common, however – they are all even.

Since parity appears to be the only common factor among the three circuits, it was decided to investigate a final class:

Definition 5.12: Class 10 $\doteq$ Class 9 $\bigcap$ Class 1.

This class contains non-equivalent faults which do not result in a function vacuous in one or more inputs, and which occur in circuits which are odd in the fault-free state. In this class, the odd/even faults were distributed as:

- 49 odd faults (52.7%)
- 44 even faults (47.3%)

In this class we see the expected proportions of even and odd faults. There are no aberrations in the data for this class. There are no circuits with 38 even faults and no odds (such as 35), nor are there any with a large number of odd faults and few evens. However, the sample size is now less than one tenth of the original (93 faults, as compared with 1310).

Class 10 is the largest class of those examined which can conclusively be shown to be representative of the distribution implied by the "all equally likely" assumption. For all the other classes, with the possible exception of 4 and 9, a high tendency towards even faults is apparent. The complements of the classes, too, consist almost entirely of even faults.

If the "all equally likely" assumption were accurate in predicting parity of fault effects, those classes which follow its prediction would contain the majority of faults. Suppose that Class 9 included even and odd faults in roughly equal proportions. It can be seen that most actual faults cause the circuit to be vacuous in terms of some input. Consider the total number of non-equivalent faults in the circuits observed, which is 959, or 25.9 for each of the 37 circuits. Of these, 385, or 10.4 per circuit, did not cause the circuit to become vacuous in one or more of its inputs. Thus, for

the sample set observed, 60% of the faults were members of a class which violates the "all equally likely" assumption simply on the basis of parity of the erroneous function.

Of course, the only class which was definitely shown to support the "all equally likely" assumption was Class 10. Only 93 of the 959 non-equivalent faults belong to this class, hence only 9.7% of actual faults would seem to follow the assumption's parity predictions.

Even if Class 10 is the largest such class, however, this alone does not entirely invalidate the assumption. If the fault effects of a circuit are randomly distributed among the even functions, with only a small percentage being odd, then some of the aliasing probabilities will have to be re-evaluated, but they probably will not change by more than a constant value. For spectral testing, the probabilities will decrease for odd functions and increase for even functions. For this reason, the higher proportion of even faults observed tends to support the argument advanced by others (e.g. [22]) that spectral techniques are more suited to odd functions.

5.2 Empirical Results

The results of the previous chapters give the performance of various signatures based upon the "all equally likely" assumption. This section gives the performance of the signatures in actual circuits. The faults are modeled at the gate level, and only single stuck-at faults are considered. In practice, most testers model the faults at the transistor level, but they do use the single stuck-at fault model. The gate model will suffice in this case, since what is desired is not numerical proof or disproof of the theoretical results, but rather a comparison of the effectiveness of the signatures.

The sample consists of 44 circuits, from 3 to 12 inputs, culled from various locations including papers, books, and actual chips. Because of the large amount of computing resources required for simulation, circuits of more than 12 inputs were not tested. The distribution of circuits and number of inputs is given below, while a description of the circuits themselves may be found in Table 5.7:

Table 5.6

Distribution of Circuits

Num. of Inputs	Num. of Circuits
3	8
4	9
5	6
6	7
7	3
8	5
9	3
10	1
12	2

Table 5.6 shows that the circuits tended to be small. The median circuit has 5 inputs, and the average number of inputs is 5.75, or approximately 6. Because of the small number of circuits, as well as their small size, the aliasing results given may not be numerically valid as "average cases" for "typical" circuits. It is claimed that most of the circuits given are typical practical circuits (with the exception of a few chosen specifically because they are syndrome-untestable).

One result which was immediately clear is that expression (3.5) does not appear to accurately reflect the aliasing probability of the 16-bit shift register used (the standard H-P model, with taps at positions 7,9,12, and 16). In fact, with all of the circuits modeled, the H-P shift register did not alias even once. This leads to the suspicion that the 2^{-16} given by most authors is in fact more accurate than Carter's value of 2^{-n}. Because of this,

additional shift registers were also simulated. These registers had n bits, with the feedback tap on the last bit (and also at positions 7 and 9, if n was large enough). No analysis of the polynomials used by the shift registers was performed, so they may or may not be irreducible. It seems reasonable enough that a shift register's aliasing probability should depend on its length, and for this reason, expression (3.3) may provide the most accurate reflection of the true aliasing probability.

To enable comparison, the signatures of a syndrome tester, the n-bit LFSR described above, and a 4-valued syndrome tester were computed for all possible faults on the 44 circuits. Equivalent faults were eliminated (this technique is commonly referred to as fault collapsing), and the aliasing was computed by dividing the number of faults which aliased by the total number of faults (over the set of non-equivalent faults). This gives an aliasing of between 0 and 1 which is reported in Table 5.8 for each of the 44 circuits. The circuits are grouped by number of inputs, and for each set of circuits with a given number of inputs, average values are provided.

It is important to note when examining the empirical results that the worst-case values encountered for actual circuits can be considerably lower than theoretical "worst-case" values for aliasing probability. This phenomenon will always result when random samples are taken from a distribution, as any statistics text will indicate (see for example [39]). For

this reason, "worst-case" refers to the worst-case distribution which can result from fixing some parameter, rather than actual samples taken from that distribution. The mean of worst-case values from sample circuits will approach the theoretical worst-case result, provided that sample circuits represent a random sample from the set of all possible functions.

5.2.1 Syndrome Testing

The worst syndrome aliasing occurs for circuit 42, closely followed by that for circuit 34. Circuit 42 is the 74LS630 parity generator, and consists entirely of exclusive-or gates. Every line but the final output is a candidate syndrome-untestable line, and all of these alias. It is well-known that this sort of circuit is very unsuitable for syndrome testing (see for instance, [22]). The only spectral coefficient which may be used to test this circuit is the "height", $r_{1,2,\ldots,10}$. Of course, the circuitry required to calculate the height is equivalent to the circuit under test, since in both cases an exclusive-or tree is necessary. Notice, though, that the 10-bit LFSR did not alias at all on the circuit. Because shift register techniques are pseudo-random, there does not appear to be any method of predicting which circuits will have higher LFSR aliasing than others. For this reason, an LFSR is more suitable than counting methods on circuits with many candidate syndrome-untestable lines.

The second circuit with a large amount of syndrome aliasing is circuit 34. This circuit is one output of an ALU from page 85 of [53]. The circuit output comes from an exclusive-or gate, so all lines but the output in the circuit are candidate syndrome-untestable. An interesting point to note is that the 74LS181 ALU (which was investigated by Savir in his original paper on syndrome testing [29]) is largely syndrome testable. With the addition of two gates and a single control input, Savir was able to make the circuit entirely syndrome testable. So we have two ALU's, implementing essentially similar functions, with vastly different syndrome testabilities. In both cases, however, virtually all lines in the circuit are candidate syndrome-untestable. In this case, though, no design for syndrome testability method was used, and the syndrome-testability of the 74LS181 appears to be more a matter of luck than anything else. However, it does raise the possibility of good design for testability methods.

Circuit number 34 contains two exclusive-or gates. When the one at the output is replaced by an OR gate, circuit 35 is obtained. The circuit is no longer functionally the same, but the syndrome aliasing probability is drastically reduced (by a factor of 10). It is still too large to permit the syndrome to be used as a signature for the circuit, however. If the final exclusive or gate is replaced by an OR gate, the aliasing probability for the syndrome counter is reduced to 0 (this circuit was not included in the set tested, though). This shows that large differences in syndrome aliasing can

occur with very minor changes in circuitry.

The other functions with large syndrome aliasing probabilities (for example, circuits 21 and 29) contain exclusive-or sections. Not all such circuits alias, though. Circuit number 27 (from [48]), for example, consists entirely of exclusive-or and exclusive-nor gates, yet has no syndrome aliasing at all.

The conclusion which may be drawn from the results is that the syndrome appears to either not alias at all, or alias too much to be of use, for circuits with more than four inputs. For this reason, if a circuit is to be tested with a syndrome counter, it is undoubtedly useful to apply the marking algorithm of [22]. This deterministic analysis will give (in an exhaustive check) a precise indication of the maximum syndrome aliasing. One may then check (either exhaustively or by random sample) the candidate syndrome-untestable lines before using the syndrome as a signature for a given circuit. If the syndrome aliasing determined in this fashion is too high, another signature should be chosen. The fact that this deterministic analysis is possible, without the large amount of simulation required by other methods, is a distinct advantage of the technique.

It would appear that for syndrome testing, the "all equally likely" assumption does not hold, except perhaps as an average. That is, actual circuits tend to have error patterns which come from those patterns which

alias or those which do not, but not both. This means the "all equally likely" assumption may be true on average, but not in specific cases.

5.2.2 Four-valued syndrome

The only statement that may be made with certainty about the results obtained for the 4-valued syndrome is that the results are uncertain. Consider, for instance, the following statistics:

- The 4-valued syndrome outperformed the binary syndrome on 9 circuits.
- The binary syndrome outperformed its 4-valued counterpart on 1 circuit.
- The two methods performed equally well on 34 circuits.

These would indicate that the 4-valued syndrome is a significant improvement on the binary syndrome. Similarly,

- The binary syndrome did not alias at all on 27 of the 44 circuits.
- The 4-valued syndrome did not alias on any of the above 27, plus 3 others.

On the other hand, the results may be rearranged so that

- The 4-valued syndrome outperformed its binary equivalent for all $n \leqslant 5$.

- The binary syndrome outperformed the 4-valued one for all $n \geq 6$.

These results indicate that the binary syndrome is more appropriate, since it provides better coverage for larger circuits. They are somewhat misleading, however, since the last statement results from a single circuit, number 34, where both methods have aliasing of over 90%. It seems unreasonable to evaluate a technique based on a single circuit, but for larger n, the 4-valued syndrome provides no improvement whatsoever in the cases observed, despite its larger range.

The expected decrease in aliasing is easily seen for the smaller circuits. For instance, consider circuit 9, where the aliasing drops from 0.3333 to 0.0 (even the LFSR aliasing was 0.25). Circuits 11 and 13 are also good examples. It is unfortunate that the data for larger n is inconclusive. Clearly more data are required.

Interestingly enough, the 4-valued syndrome always aliased on candidate binary syndrome-untestable lines, and in all cases but circuit 34, aliased on lines where the binary syndrome also aliased. This leads to the conclusion that the 4-valued syndrome is in practice even more closely tied to the binary syndrome than the theoretical results of chapter 4 indicate. This suggests that the 4-valued syndrome will probably result in a small amount of improvement in aliasing, possibly enough to justify the additional silicon area involved in its calculation.

The aliasing observed for circuits 29, 34, and 42 suggests that in cases where the syndrome aliasing is too high to permit its use as a signature, the 4-valued syndrome will not offer significant improvement. If additional data confirms this hypothesis, the 4-valued syndrome is probably of use only for circuits with binary syndrome aliasing just marginally above an acceptable value. The 4-valued syndrome appears significantly better than its binary counterpart for $n \leq 5$, but beyond that the two are essentially the same. In fact, the 4-valued syndrome aliases on one line of the ALU in circuit 34 that the binary syndrome does not (again, probably a matter of chance). Because the two syndrome methods are so closely related (in all cases the 4-valued syndrome aliased on candidate binary syndrome-untestable lines) it may be that the theoretical gain in coverage of the 4-valued method is not in fact realized in larger circuits. The counting hardware of a 4-valued syndrome is more complex than that for the binary version, and thus may not be worth the cost if little additional coverage is obtained. Clearly, additional data are required to draw firm conclusions.

5.2.3 Spectral Coefficients

Table 5.9 gives the empirical aliasing observed for the 44 test circuits when various spectral coefficients are used as signatures. The coefficients considered are the syndrome, the height, and the first-order coefficients. The meaning of the columns is given below:

Column Definitions for Table 5.9

Column Name	Definition
circ	circuit number
n	number of inputs
a	aliasing of syndrome
b	aliasing of height
c	lowest aliasing of any first order coeff.
d	highest aliasing of any first order coeff.
e	lowest aliasing of largest first order coeff.
f	highest aliasing of largest first order coeff.
g	lowest aliasing of smallest first order coeff.
h	highest aliasing of smallest first order coeff.
i	aliasing of a random first order coeff.
j	average aliasing of first order coeff.

As with Table 5.8, the averages for circuits with the same number of inputs are tabulated along with the values for each circuit. In the definitions above, largest first order coefficient refers to the largest absolute value, and smallest refers to the smallest absolute value. Since many circuits have more than one coefficient of this maximal value, the highest and lowest aliasing found within each set is also listed. The random coefficients were chosen using a table of random numbers on page 605 of [39]. This

method was used to avoid some of the problems associated with pseudo-random number generators (such as those present in many computers).

Under the "all equally likely" assumption, each spectral coefficient has an identical distribution (including the syndrome). Therefore, when considering two distinct coefficients, in this case the height and the syndrome, one would expect that their aliasing values would be essentially equal. Even in the small sample of Table 5.9, it can be seen that this is not so. With the exception of the pathologically syndrome-untestable functions mentioned earlier, the height has consistently higher aliasing for circuits of more than 3 inputs. The reason for this behaviour seems to be that the height tends to remain very close to the midrange value. In fact, 38 of the 44 circuits have a height that is within the range [-2,2]. Compare this to the 23 of 44 circuits which have a syndrome within 2 of the midrange. It seems that the "all equally likely" assumption's prediction that values close to the midrange will alias more often than those further away is supported by the fact that both circuit 42 and 34 (those with the highest syndrome aliasing) have a syndrome of precisely the midrange value.

Another interesting observation is that the syndrome and the height appear almost orthogonal in their fault coverage. In the circuits of more than 4 inputs tested, a high value for the syndrome aliasing implies a low value for the height aliasing and vice versa. Some extractions from Table

5.9 are shown below to emphasize the point.

Comparison of Syndrome and Height Aliasing

Circuit	Syndrome Aliasing	Height Aliasing
21	0.3636	0.0
22	0.0625	0.9375
27	0.0	0.8889
30	0.0	1.0000
34	0.9130	0.0217
42	0.9474	0.0

In many cases, an unreasonably high value for either syndrome or height aliasing implies zero aliasing for the other coefficient. This observation could be exploited to produce a signature with very high fault coverage, as is seen in a subsequent section.

The "all equally likely" assumption suggests that those values furthest from the midrange will alias the least. Therefore, one might expect that the largest absolute value first order coefficient will have the lowest aliasing of the first order coefficients. This appears to be true in most cases, but possibly not to the extent expected. For example, there are several circuits (14, 23, and 25) where even the lowest aliasing for the first order coefficient of largest absolute value (column e) is higher than the average aliasing of first order coefficients (column j). In general, though, the largest first order coefficient has aliasing that is lower than that of the smallest first order coefficient. In some cases, though, there is wide disparity between aliasing of coefficients of the same magnitude.

Compare column e to column f and column g to column h to see this disparity.

Occasionally the smallest absolute value coefficient has the least aliasing, as in circuit 31. It cannot be asserted without more data, of course, but it certainly appears from Table 5.9 that the largest absolute value coefficients do provide better fault coverage than average, usually better than the smallest coefficients. They do not provide the best coverage, though, in all cases, and sometimes picking a coefficient at random seems to cover just about as well. Of course, a truly random selection process will eventually pick coefficients such that aliasing is average, so column j should be taken as a more accurate reflection of random coefficient aliasing than column i.

The conclusions which may be drawn from Table 5.9 are that the syndrome and the height are almost orthogonal in their fault coverage, and that the heuristic of choosing the largest absolute value coefficient is generally a good choice, although as with any heuristic, it is not foolproof. A final interesting sideline is the difference in aliasing of first order coefficients between circuits 39 and 40. Both are carry circuits for a full adder, and yet the fault coverage of the smallest coefficients is substantially different (in both cases the smallest coefficient has absolute value 1). This again shows the usefulness of design for testability as a design stra-

tegy in that different realizations of the same circuit can be vastly different in their testability. It is also interesting that the more testable of the two, circuit 40, has 20% fewer lines than circuit 39.

5.2.4 Syndrome and Additional Spectral Coefficient

Table 5.10 presents the aliasing observed for the 44 test circuits when signatures consisting of the syndrome plus an additional coefficient are considered. The coefficients used in addition to the syndrome are again the height and first order coefficients.

Column Definitions for Table 5.10

Column Name	Definition
circ	circuit number
n	number of inputs
a	aliasing of syndrome
b	lowest aliasing of synd., any first order coeff.
c	highest aliasing of synd., any first order coeff.
d	lowest aliasing of synd., largest first order coeff.
e	highest aliasing of synd., largest first order coeff.
f	lowest aliasing of synd., smallest first order coeff.
g	highest aliasing of synd., smallest first order coeff.
h	aliasing of syndrome and height
i	aliasing of synd., random first order coeff.
j	average aliasing of synd., first order coeff.

The most obvious change from Table 5.9 to Table 5.10 is the great preponderance of zero entries. These stem primarily from the obvious fact that if the syndrome does not alias, a signature consisting of the syndrome

and something else will also not alias. In chapter 4 it was shown that under the "all equally likely" assumption, the signatures of Table 5.10 should have aliasing probability approximately equal to those of an n-bit LFSR. (Table 4.2 tabulates the values of aliasing probability of the two types of signature).

Once again, large discrepancies between the theory and reality of fault coverage by the signatures are apparent. For instance, the signature consisting of the syndrome and the height should have approximately the same fault coverage as that of the syndrome and a first order coefficient. It can be seen, though, that the syndrome and the height (column h) is consistently lower in aliasing than either the average first order coefficient and syndrome aliasing (column j), or that of a randomly chosen first order coefficient and the syndrome (column i). In fact, column h contains aliasing frequently lower than that of column b, the best value for a first order coefficient and the syndrome. Notice that the worst value for syndrome and some first order coefficient, column c, is for all circuits of more than 4 inputs, the same as the syndrome aliasing alone.

What is apparent, based on the results of Table 5.10, is that first order coefficients are very closely tied to the syndrome. As a result, the significant gains predicted by chapter 4 with regard to the aliasing probability of these signatures do not materialize in actuality. The circuitry

required to calculate the syndrome and a first order coefficient is somewhat more complicated than that needed for the syndrome alone, and the gains realized in the average case are small. That is, extra storage is required for the two signatures, as well as additional routing circuitry (see Figure 5.2), but the gains realized even in the best case (column b) are typically not more than 30% in terms of reduced aliasing. Certainly, the squaring effect predicted by chapter 4 is not apparent.

The two circuits with the highest syndrome aliasing, numbers 34 and 42, do not have their aliasing reduced at all with the addition of any first order coefficient. So it seems that the pathologically syndrome-untestable circuits are not made any more testable with the addition of a first-order coefficient as part of the testing signature. Because of this situation, it is probably better to simply use the syndrome alone as a signature on those circuits which are clearly indicated as testable by the marking algorithm of Miller and Muzio, and to use some other signature on pathologically syndrome-untestable circuits. The signature consisting of the syndrome and some first order coefficient, regardless of the heuristic used to choose that coefficient, seems to be uneconomical in terms of increased coverage versus increased tester cost.

5.2.5 Syndrome and Height

The counting signature which does show some promise is that which consists of the syndrome and the height. Coverage comparable to an LFSR is obtained by this signature, as indicated by Tables 5.8 and 5.10. In fact, this signature outperforms the LFSR in the average values for all $n < 7$, and performs as least as well in the average values for all cases except n=8. The tabulation below indicates this performance:

Aliasing Comparison
Syndrome/Height Signature and LFSR

Circuits where LFSR has lower aliasing	4
Circuits where Syndrome/Height has lower aliasing	14
Circuits where coverage equal (both 0)	26
Total	44

It seems that the syndrome/height signature lives up to the performance expectations predicted by the previous chapter. The only drawback of this signature is the large amount of circuitry needed to calculate it. A parity tree is required at some point in the tester (either in the input generation, or the output compactor, depending on the design chosen) and thus considerably more silicon area is required than that used by an LFSR if the tester is to be included on-chip. The signature has the advantage though that a deterministic analysis exists which will indicate those lines on which faults may be untestable. This of course is the syndrome marking algorithm [22], which further research might extend to predict those

faults which may be both syndrome and height untestable. Deterministic methods are suited to design for testability and so the syndrome/height signature may be ultimately more useful than the purely random shift register techniques.

5.2.6 Best Signatures

Table 5.11 gives the aliasing probability of the four "best" signatures examined. These are the combination of the syndrome and the height, the combination of the syndrome and the n-bit LFSR, the n-bit LFSR alone, and the 16-bit LFSR. The signature of the syndrome and the n-bit LFSR was included because of Robinson's claim [27] that the two techniques are almost orthogonal: i.e. using them together gives significant reduction in aliasing. Our results are consistent with this view.

It seems that the signatures which depend on either counting techniques or pseudo-randomization techniques are prone to occasional "glitches" of high aliasing. Circuits 21 and 28, for instance, have large n-bit LFSR aliasing (in comparison to the other circuits of the same number of inputs). Similarly, circuit 34 only has 97.83% single stuck-at fault coverage by the syndrome/height signature. Because the LFSR/syndrome signature combines two methods, its coverage does not appear to be as effected by occasional instances of high aliasing. This signature still

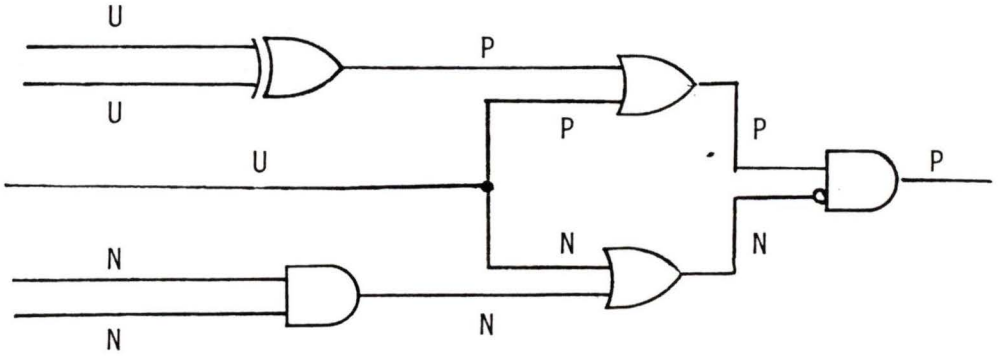
allows the deterministic design for testability possible with syndrome testing, with the added benefit that candidate syndrome-untestable lines are very probably covered with 2^{-m} certainty by an m-bit LFSR.

As an example of this method, suppose a 10 input circuit is designed where 1% of the lines are candidate syndrome-untestable. A syndrome count alone assures 99% fault coverage, even without simulation of the circuit. A 10-bit LFSR likely has 99.99% fault coverage, but there is no guaranteed coverage without simulation. If the two signatures are combined, there is still guaranteed 99% fault coverage, and there is almost 99.99% probability that the candidate syndrome-untestable lines are covered. This means that the signature probably covers 99.9999% of all the faults, and is guaranteed at least 99% coverage, with no simulation of the circuit whatsoever. It is these guaranteed lower bounds possible with syndrome testing that make it a useful technique. Other signature methods, although they may have theoretically higher coverage probabilistically, do not have the lower bound on coverage of the syndrome count.

All of the signatures used in Table 5.11 are good, and each has its benefits. The syndrome/height signature may be completely amenable to a deterministic analysis of its behaviour. The syndrome/LFSR combination has a guaranteed lower bound of coverage, and because of the orthogonality of its components appears to greatly reduce the incidence of "worst-

case" circuits. The LFSR alone is a good signature because of its low cost in terms of tester size. The n-bit LFSR tends to fail on some circuits, although it appears that by making the shift register long enough (16 bits for the circuits tested here) aliasing can be virtually eliminated. The tester chosen, then, can depend on the design methodology used, or the amount of chip area available for a built-in self tester. In many cases, however, any of the four signatures mentioned above will cover all single stuck-at faults of the circuit.

P=positive inversion parity
N=negative inversion parity
C=candidate syndrome-untestable



Inversion Parity of Circuit Lines

Figure 5.1

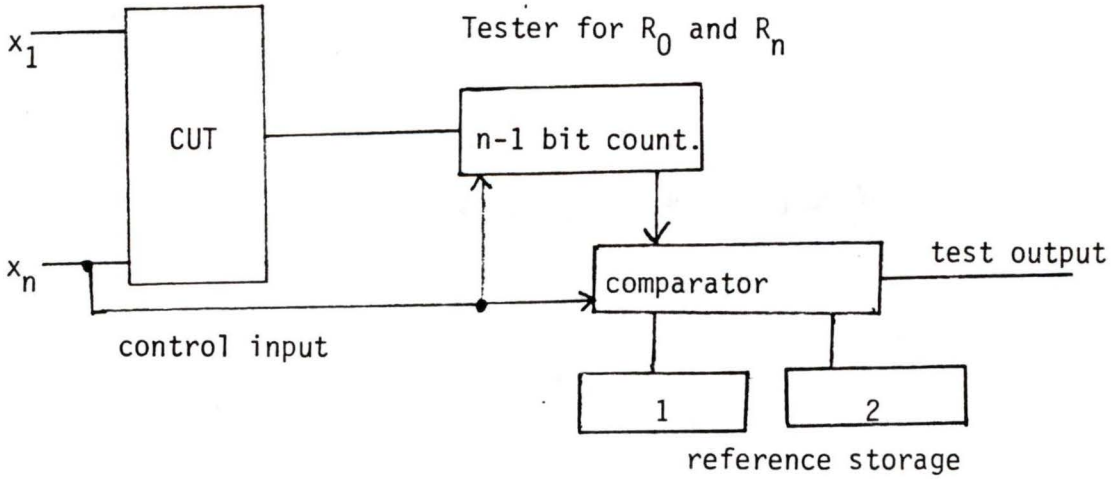


Figure 5.2

Table 5.7

Description of Circuits

circuit	number of inputs	number of lines	parity	description	source
1	3	11	E	carry stage for an adder	
2	3	17	O	random logic circuit	[21]
3	3	27	E	sum stage of full adder	[54]
4	3	13	E	BCD to excess 3 convert.	[20]
5	3	12	E	random logic circuit	[23]
6	3	13	E	random logic circuit	[55]
7	3	9	O	PLA	[20]
8	3	13	O	random logic circuit	[43]
9	4	10	E	random logic circuit	[22]
10	4	19	E	random logic circuit	[23]
11	4	19	E	random logic circuit	[22]
12	4	16	E	random logic circuit	[23]
13	4	26	E	random logic circuit	[4]
14	4	10	E	Intel 8212 I/O port WR	
15	4	31	E	ALU control decoder	[47]
16	4	22	E	receiver logic	[51]
17	4	14	O	parity predicter	[50]
18	5	41	O	2 out of 5 checker	[16]
19	5	16	O	random logic circuit	[19]
20	5	12	O	random logic circuit	
21	5	8	E	random logic circuit	
22	5	28	E	random PLA	
23	5	16	E	random NAND circuit	
24	6	21	E	and-or latch w/o fback.	[53]
25	6	17	O	random logic circuit	[44]
26	6	39	E	XOR latch w/o feedback	[46]
27	6	23	E	error correction circuit	[48]
28	6	15	E	random logic circuit	
29	6	27	E	random PLA	
30	6	34	E	random NAND circuit	
31	7	14	E	building block of 5MR	[16]
32	7	11	O	random logic circuit	[44]

33	7	12	O	random logic circuit	[43]
34	8	38	E	ALU	[52]
35	8	38	E	34, last gate changed	
36	8	18	O	random logic circuit	[52]
37	8	13	O	random logic circuit	[55]
38	8	15	O	random logic circuit	[45]
39	9	29	O	8-bit carry circuit	[54]
40	9	23	O	8-bit carry circuit	[49]
41	9	23	E	SN7480 gated adder	
42	10	21	E	TI 74LS630 parity gen.	
43	12	29	O	random logic circuit	[22]
44	12	24	O	random logic circuit	[32]

Table 5.8

Empirical Aliasing
Syndrome, n-bit LFSR, 4-Valued Syndrome Testing

Circuit	Num. of Inputs	Syndrome	LFSR	4-Val Synd.
1	3	0.1176	0.4706	0.1176
2	3	0.0	0.0	0.0
3	3	0.1579	0.1842	0.1579
4	3	0.2857	0.2143	0.2143
5	3	0.4706	0.2353	0.3529
6	3	0.25	0.125	0.0
7	3	0.0	0.2	0.0
8	3	0.0	0.0714	0.0
avg	3	0.1602	0.1876	0.1053
9	4	0.3333	0.25	0.0
10	4	0.0	0.0	0.0
11	4	0.1786	0.0714	0.0357
12	4	0.1818	0.0455	0.1818
13	4	0.2667	0.0333	0.0667
14	4	0.0	0.0	0.0
15	4	0.0556	0.0278	0.0
16	4	0.2667	0.0667	0.2
17	4	0.0	0.0625	0.0
avg	4	0.1425	0.0619	0.0316
18	5	0.0	0.0	0.0
19	5	0.0	0.0	0.0
20	5	0.0	0.0	0.0
21	5	0.3636	0.0909	0.1818
22	5	0.0625	0.0	0.0625
23	5	0.0	0.0	0.0
avg	5	0.0710	0.0151	0.0407
24	6	0.0	0.0	0.0
25	6	0.0	0.0	0.0
26	6	0.0	0.0	0.0
27	6	0.0	0.0	0.0
28	6	0.0	0.0833	0.0
29	6	0.2857	0.0	0.2857

30	6	0.0	0.0	0.0
avg	6	0.0408	0.0119	0.0408
31	7	0.0	0.0	0.0
32	7	0.0	0.0	0.0
33	7	0.0	0.0	0.0
avg	7	0.0	0.0	0.0
34	8	0.9130	0.0	0.9565
35	8	0.0930	0.0	0.0930
36	8	0.0	0.0	0.0
37	8	0.0	0.0	0.0
38	8	0.0	0.0	0.0
avg	8	0.2012	0.0	0.2099
39	9	0.0	0.0	0.0
40	9	0.0	0.0	0.0
41	9	0.0	0.0	0.0
avg	9	0.0	0.0	0.0
42	10	0.9474	0.0	0.9474
avg	10	0.9474	0.0	0.9474
43	12	0.0	0.0	0.0
44	12	0.0	0.0	0.0
avg	12	0.0	0.0	0.0

Table 5.9

Empirical Aliasing
Spectral Coefficients

circ	n	a	b	c	d	e	f	g	h	i	j
1	3	0.1176	0.0588	0.3529	0.4118	0.3529	0.4118	0.3529	0.4118	0.3529	0.3724
2	3	0.0	0.1364	0.1364	0.1364	0.1364	0.1364	0.1364	0.1364	0.1364	0.1364
3	3	0.1579	0.0	0.4211	0.4737	0.4211	0.4737	0.4211	0.4737	0.4211	0.4387
4	3	0.2857	0.0	0.0	0.4286	0.0	0.0	0.4286	0.4286	0.4286	0.2857
5	3	0.4706	0.5294	0.2941	0.4118	0.2941	0.4118	0.2941	0.4118	0.4118	0.3724
6	3	0.25	0.0	0.0	0.5	0.0	0.0	0.5	0.5	0.5	0.3333
7	3	0.0	0.2	0.0	0.1	0.0	0.0	0.1	0.1	0.1	0.0667
8	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
avg	3	0.1602	0.0906	0.1506	0.3078	0.1506	0.1792	0.2791	0.3078	0.2939	0.2507
9	4	0.3333	0.8333	0.1667	0.5	0.1667	0.1667	0.5	0.5	0.1667	0.3333
10	4	0.0	1.0000	0.0	0.8	0.0	0.0	0.5	0.8	0.5	0.45
11	4	0.1786	0.25	0.1786	0.1786	0.1786	0.1786	0.1786	0.1786	0.1786	0.1786
12	4	0.1818	0.1818	0.1818	0.2727	0.1818	0.2727	0.1818	0.2727	0.1818	0.2386
13	4	0.2667	0.0667	0.1667	0.2667	0.1667	0.2667	0.1667	0.2667	0.1667	0.2
14	4	0.0	0.8333	0.0	0.3333	0.3333	0.3333	0.0	0.1667	0.1667	0.1667
15	4	0.0556	0.1389	0.1111	0.2222	0.1111	0.2222	0.1667	0.1667	0.1667	0.1667
16	4	0.2667	0.0667	0.5667	0.6333	0.5667	0.5667	0.5667	0.6333	0.5667	0.5833
17	4	0.0	0.0625	0.0	0.0625	0.0	0.0625	0.0	0.0625	0.0625	0.0156
avg	4	0.1425	0.3815	0.1524	0.3633	0.1894	0.2299	0.2517	0.3386	0.2396	0.2592
18	5	0.0	0.2857	0.0238	0.1667	0.0238	0.0476	0.1667	0.1667	0.0238	0.0619
19	5	0.0	0.1667	0.0	0.1667	0.0	0.0	0.0	0.1667	0.0	0.0333
20	5	0.0	0.0714	0.0	0.0714	0.0	0.0	0.0	0.0714	0.0	0.0143
21	5	0.3636	0.0	0.3636	0.8182	0.3636	0.3636	0.8182	0.8182	0.3636	0.5455
22	5	0.0625	0.9375	0.0625	0.4688	0.125	0.125	0.4688	0.4688	0.0625	0.1813
23	5	0.0	1.0000	0.0	0.2778	0.2222	0.2222	0.0	0.2788	0.2788	0.2
avg	5	0.0710	0.4102	0.0750	0.3283	0.1224	0.1264	0.2423	0.3283	0.1215	0.1727
24	6	0.0	1.0000	0.0	1.0000	0.0	0.0	1.0000	1.0000	0.0	0.1667
25	6	0.0	0.125	0.0	0.0625	0.0625	0.0625	0.0625	0.0625	0.0	0.0521
26	6	0.0	0.6818	0.0	1.0000	0.0	0.0	0.0909	1.0000	0.0909	0.3636
27	6	0.0	0.8889	0.0	0.8889	0.0	0.0	0.8889	0.8889	0.8999	0.5926
28	6	0.0	0.8333	0.0	0.8333	0.0	0.0	0.75	0.8333	0.0	0.4028
29	6	0.2857	0.0	0.2857	0.8214	0.2857	0.2857	0.75	0.8214	0.75	0.6190

30	6	0.0	1.0000	0.0	1.0000	0.0357	0.0357	1.0000	1.0000	0.1071	0.2143
avg	6	0.0408	0.6470	0.0408	0.8009	0.0548	0.0548	0.6489	0.8009	0.2568	0.3444
31	7	0.0	0.875	0.0	0.25	0.25	0.25	0.125	0.125	0.125	0.0892
32	7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
33	7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
avg	7	0.0	0.2917	0.0	0.0833	0.0833	0.0833	0.0417	0.0417	0.0417	0.0297
34	8	0.9130	0.0217	0.9665	0.9665	0.9665	0.9665	0.9665	0.9665	0.9665	0.9665
35	8	0.0930	0.0232	0.0930	0.9535	0.3256	0.3256	0.9535	0.9535	0.0930	0.4017
36	8	0.0	0.0556	0.0	0.0556	0.0	0.0556	0.0	0.0	0.0556	0.0069
37	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
38	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
avg	8	0.2012	0.0201	0.2119	0.5840	0.2584	0.2695	0.3840	0.3840	0.2230	0.2750
39	9	0.0	0.3333	0.0	0.25	0.0	0.0	0.25	0.25	0.0	0.1142
40	9	0.0	0.1481	0.0	0.0741	0.0	0.0	0.0741	0.0741	0.0370	0.0411
41	9	0.0	0.0	0.0	0.2593	0.0	0.0	0.2593	0.2593	0.2593	0.2305
avg	9	0.0	0.1605	0.0	0.1947	0.0	0.0	0.1947	0.1947	0.0988	0.1286
42	10	0.9474	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
avg	10	0.9474	0.0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
43	12	0.0	0.1176	0.0	0.0294	0.0	0.0	0.0	0.0	0.0294	0.0147
44	12	0.0	0.0769	0.0	0.0385	0.0	0.0	0.0385	0.0385	0.0385	0.0256
avg	12	0.0	0.0973	0.0	0.0340	0.0	0.0	0.0193	0.0193	0.0340	0.0202

Table 5.10

Empirical Aliasing
Syndrome and Additional Spectral Coefficient

circ	n	a	b	c	d	e	f	g	h	i	j
1	3	0.1176	0.0589	0.1176	0.0589	0.1176	0.0589	0.1176	0.0	0.0589	0.0782
2	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	3	0.1579	0.1579	0.1579	0.1579	0.1579	0.1579	0.1579	0.0	0.1579	0.1579
4	3	0.2857	0.0	0.2857	0.0	0.0	0.2857	0.2857	0.0	0.2857	0.1907
5	3	0.4706	0.1176	0.2353	0.1176	0.2353	0.1176	0.2353	0.3529	0.2353	0.1959
6	3	0.25	0.0	0.25	0.0	0.0	0.25	0.25	0.0	0.25	0.1667
7	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
avg	3	0.1602	0.0418	0.1308	0.0418	0.0639	0.1235	0.1308	0.0441	0.1235	0.0987
9	4	0.3333	0.0	0.3333	0.0	0.0	0.3333	0.3333	0.3333	0.0	0.1667
10	4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	4	0.1786	0.0357	0.1428	0.0357	0.1428	0.0357	0.1428	0.0	0.1428	0.1160
12	4	0.1818	0.0	0.1818	0.0	0.1818	0.0	0.1818	0.0	0.1818	0.1364
13	4	0.2667	0.0667	0.0667	0.0667	0.0667	0.0667	0.0667	0.0	0.0667	0.0667
14	4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	4	0.0556	0.0	0.0556	0.0	0.0	0.0556	0.0556	0.0	0.0	0.0139
16	4	0.2667	0.1333	0.2	0.1333	0.2	0.2	0.2	0.0	0.1333	0.1833
17	4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
avg	4	0.1425	0.0262	0.0867	0.0262	0.0657	0.0546	0.0867	0.0370	0.0583	0.0759
18	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21	5	0.3636	0.1818	0.3636	0.3636	0.3636	0.1818	0.1818	0.0	0.1818	0.2909
22	5	0.0625	0.0	0.0625	0.0	0.0	0.0625	0.0625	0.0625	0.0	0.0375
23	5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
avg	5	0.0710	0.0303	0.0710	0.0606	0.0606	0.0407	0.0407	0.0104	0.0303	0.0547
24	6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25	6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
26	6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
27	6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
28	6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
29	6	0.2857	0.2143	0.2857	0.2857	0.2857	0.2143	0.2143	0.0	0.2143	0.2381

30	6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
avg	6	0.0408	0.0306	0.0408	0.0408	0.0408	0.0306	0.0306	0.0	0.0306	0.0340
31	7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
32	7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
33	7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
avg	7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
34	8	0.9130	0.9130	0.9130	0.9130	0.9130	0.9130	0.9130	0.0217	0.9130	0.9130
35	8	0.0930	0.0465	0.0930	0.0930	0.0930	0.0465	0.0465	0.0	0.0465	0.0814
36	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
37	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
38	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
avg	8	0.2012	0.1919	0.2012	0.2012	0.2012	0.1919	0.1919	0.0043	0.1919	0.1989
39	9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
40	9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
41	9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
avg	9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
42	10	0.9474	0.9474	0.9474	0.9474	0.9474	0.9474	0.9474	0.0	0.9474	0.9474
avg	10	0.9474	0.9474	0.9474	0.9474	0.9474	0.9474	0.9474	0.0	0.9474	0.9474
43	12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
44	12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
avg	12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 5.11

Empirical Aliasing
Best Signature

circuit	number of inputs	syndrome/height	synd., n-bit LFSR	n-bit LFSR	16-bit LFSR
1	3	0.0	0.0588	0.4706	0.0
2	3	0.0	0.0	0.0	0.0
3	3	0.0	0.0	0.1842	0.0
4	3	0.0	0.0	0.2143	0.0
5	3	0.3529	0.1765	0.2353	0.0
6	3	0.0	0.0625	0.125	0.0
7	3	0.0	0.0	0.2	0.0
8	3	0.0	0.0	0.0714	0.0
avg	3	0.0441	0.0372	0.1876	0.0
9	4	0.3333	0.0833	0.25	0.0
10	4	0.0	0.0	0.0	0.0
11	4	0.0	0.0	0.0714	0.0
12	4	0.0	0.0	0.0455	0.0
13	4	0.0	0.0333	0.0333	0.0
14	4	0.0	0.0	0.0	0.0
15	4	0.0	0.0	0.0278	0.0
16	4	0.0	0.0333	0.0667	0.0
17	4	0.0	0.0	0.0625	0.0
avg	4	0.0370	0.0167	0.0619	0.0
18	5	0.0	0.0	0.0	0.0
19	5	0.0	0.0	0.0	0.0
20	5	0.0	0.0	0.0	0.0
21	5	0.0	0.0	0.0909	0.0
22	5	0.0625	0.0	0.0	0.0
23	5	0.0	0.0	0.0	0.0
avg	5	0.0104	0.0	0.0151	0.0
24	6	0.0	0.0	0.0	0.0
25	6	0.0	0.0	0.0	0.0
26	6	0.0	0.0	0.0	0.0
27	6	0.0	0.0	0.0	0.0
28	6	0.0	0.0	0.0833	0.0
29	6	0.0	0.0	0.0	0.0

30	6	0.0	0.0	0.0	0.0
avg	6	0.0	0.0	0.0119	0.0
31	7	0.0	0.0	0.0	0.0
32	7	0.0	0.0	0.0	0.0
33	7	0.0	0.0	0.0	0.0
avg	7	0.0	0.0	0.0	0.0
34	8	0.0217	0.0	0.0	0.0
35	8	0.0	0.0	0.0	0.0
36	8	0.0	0.0	0.0	0.0
37	8	0.0	0.0	0.0	0.0
38	8	0.0	0.0	0.0	0.0
avg	8	0.0043	0.0	0.0	0.0
39	9	0.0	0.0	0.0	0.0
40	9	0.0	0.0	0.0	0.0
41	9	0.0	0.0	0.0	0.0
avg	9	0.0	0.0	0.0	0.0
42	10	0.0	0.0	0.0	0.0
avg	10	0.0	0.0	0.0	0.0
43	12	0.0	0.0	0.0	0.0
44	12	0.0	0.0	0.0	0.0
avg	12	0.0	0.0	0.0	0.0

Chapter 6: Conclusions

Performance measures for various data compression techniques in both the theoretical and empirical domains have been derived. Trends predicted in theory are observable in the empirical data, although there are exceptions. We have shown, also, that the theoretical framework provided by the "all equally likely" assumption is unable to correctly predict the parity of fault effects, although the consequences of this observation remain unclear.

On the basis of theoretical aliasing probability, the techniques examined rank in the following order: syndrome and additional spectral coefficient testing, LFSR testing, 4-valued syndrome testing, and finally binary syndrome testing. These theoretical results implicitly depend on the "all equally likely" assumption, hence reflect error coverage. The assumption's validity determines their fault coverage.

Of course, fault coverage is not the only issue. In terms of tester cost, the LFSR is the least expensive, and the 2 coefficient signature is probably the most expensive. LFSR testing, though, is random in nature, and no deterministic analysis of its performance, other than exhaustive simulation, is known. An upper bound on aliasing can be found deterministically for any signature which includes the syndrome, and empirical results indicate that this bound also applies to 4-valued testing.

The empirical performance measures indicate that the aliasing of an m-bit LFSR is most accurately given by 2^{-m}, rather than the 2^{-n} given by Carter [6] for exhaustive testing. In addition, syndrome testing is observed to have, for larger circuits, either no aliasing at all, or unacceptably high levels of aliasing.

It was observed that the 4-valued syndrome is, for circuits of only a few inputs, a significant improvement over the binary syndrome. For larger circuits, however, the results obtained are inconclusive.

The performance of individual first-order spectral coefficients was seen to be substantially worse than syndrome counting on average. This result directly contradicts the equal coverage predicted in theory. Furthermore, when these coefficients were combined with the syndrome, only a small improvement in coverage was observed, and none at all on those circuits where the syndrome categorically failed as a signature (aliasing >0.5). The heuristic of choosing the largest absolute value coefficient produces better than average results, but not always the best possible.

The height by itself is observed to be a very bad signature (again contradicting the coverage predicted by the "all equally likely" assumption). On the other hand, its coverage appears in practice to be orthogonal to that of the syndrome, and together they provide a signature which rivals the LFSR in performance. The syndrome/height coefficient is the combination

of two spectral coefficients which more closely matches the theoretically predicted aliasing behaviour than any of the others examined.

The syndrome and the LFSR together provide a very good signature, which outperforms both the syndrome/height and n-bit LFSR alone. The example of the 16-bit LFSR shows, however, that by extending the length of an LFSR, it is always possible to obtain 100% fault coverage. There is no known method of determining the minimal length LFSR necessary, however, outside of exhaustive simulation. Because those signatures which include the syndrome can be analyzed deterministically, though, they are more suited to design for testability techniques.

Based on the empirical data presented here, and supported by theoretical evidence, any one of the syndrome/height, syndrome/n-bit LFSR, or the $\geqslant$n-bit LFSR is an excellent signature to use for a given digital circuit.

Some areas where further research is necessary have been brought to light during the preparation of this thesis. These are presented below as open problems:

- Is the "all equally likely" assumption the best characterization of actual fault patterns?
- Is the assumption's failure to correctly predict fault parity symptomatic of overall flaws?

- Is the "no aliasing"/"unacceptably high aliasing" trend observed for syndrome testing a general phenomenon?
- Is 4-valued syndrome testing an improvement (in terms of fault coverage) over binary syndrome testing?
- Is any deterministic analysis of LFSR aliasing for actual circuits (other than exhaustive simulation) possible?
- Can deterministic upper bounds on aliasing be found for spectral coefficients other than the syndrome (especially the height)?
- How applicable is the marking algorithm of Miller and Muzio [22] to 4-valued syndrome testing?
- How does the 4-valued syndrome compare with other linear combinations of spectral coefficients in terms of fault coverage?
- How do higher order spectral coefficients other than the height perform empirically?
- Does a syndrome and n-bit LFSR signature outperform a 2n-bit LFSR?

Bibliography

[1] V.K. Agarwal, "Increasing Effectiveness of Built-In Testing by Output Data Modification", Proceedings of FTCS-13, Milano, Italy, June 1983.

[2] R.C. Aitken, P.K. Lui, M. Serra, and J.C. Muzio, "Aliasing Probabilities of Some Data Compression Techniques", Proceedings of Technical Workshop: New Directions for IC Testing, Victoria, B.C., March 1986.

[3] J.A. Bate and D.M. Miller, "On the Exhaustive Testing of Stuck-Open Faults in CMOS Combinational Circuits", Proceedings of Technical Workshop: New Directions for IC Testing, Victoria, B.C., March 1986.

[4] F. Brglez, "Testability in VLSI", Technical Digest, Canadian Conference on VLSI, Waterloo, ON, October, 1983.

[5] J.A. Brzozowski, "Testable CMOS Cells", Proceedings of Technical Workshop: New Directions for IC Testing, Victoria, B.C., March 1986.

[6] J. Carter, "The Theory of Signature Testing for VLSI", Proceedings of 14th ACM Symp. Theory of Computing, May 1982.

[7] R. Frohwerk, "Signature Analysis: A New Digital Field Service Method", Hewlett-Packard Journal, May 1977.

[8] H. Fujiwara, *Logic Testing and Design for Testability*, The MIT Press, Cambridge MA, 1985.

[9] S.W. Golomb, *Shift Register Sequences*, Holden-Day Inc., San Francisco CA, 1967.

[10] H. Gould, *Combinatorial Identities*, Morgantown Printing and Binding Co., Morgantown WV, 1972.

[11] J.P. Hayes, "Transition Count Testing of Combinational Logic Circuits", IEEE Trans. Comp., Vol. C-25, June 1976.

[12] T.C. Hsiao and S.C. Seth, "The Use of Rademacher-Walsh Spectrum in Testing and Design of Digital Circuits", IEEE Inter. Conf. on Circ. and Comp., New York NY, September 1982.

[13] T.C. Hsiao and S.C. Seth, "An Analysis of the Use of Rademacher-Walsh Spectrum in Compact Testing", IEEE Trans. Comp., Vol. C-33, October, 1984.

[14] S.L. Hurst, *The Logical Processing of Digital Signals*, Crane Russak, New York NY, 1978.

[15] S.L. Hurst, D.M. Miller, and J.C. Muzio, *Spectral Techniques in Digital Logic*, Academic Press, London, 1985.

[16] P.K. Lala, *Fault Tolerant and Fault Testable Hardware Design*, Prentice-Hall Inter., Englewood Cliffs NJ, 1985.

[17] J. Losq, "Efficiency of Random Compact Testing", IEEE Trans. Comp., Vol C-27, June 1978.

[18] P.K. Lui and J.C. Muzio, "Spectral Signature Testing of Multiple Stuck-at Faults in Irredundant Combinational Circuits", submitted to IEEE Trans. Comp.

[19] E.J. McCluskey, "Functional Testing: A State of the Art Survey", Tutorial Session Notes, ICCAD 85, Santa Clara CA, November 1985.

[20] M.M. Mano, *Digital Logic and Computer Design*, Prentice-Hall Inter., Englewood Cliffs NJ, 1979.

[21] D.M. Miller and J.C. Muzio, "Spectral Fault Signatures for Internally Unate Combinational Networks", IEEE Trans. Comp., Vol. C-32, November, 1983.

[22] D.M. Miller and J.C. Muzio, "Spectral Fault Signatures for Single Stuck-At Faults in Combinational Networks", IEEE Trans. Comp., Vol. C-33, August, 1984.

[23] J.C. Muzio, "Data Compression Techniques for Fault Detection", University of Victoria Fault Detection Research Group Report 04-

1984, 1984.

[24] W.W. Peterson, *Error-Correcting Codes*, The MIT Press, Cambridge MA, 1965.

[25] M.W. Roberts and P.K. Lala, "An Algorithm for the Partitioning of Logic Circuits", IEE Proceedings, Vol 131, Pt. E, No. 4, July 1984.

[26] J.P. Robinson and C-H Tung, "Fast Algorithms for Optimum Syndrome Space Compression", submitted to IEEE Trans. Comp.

[27] J.P. Robinson and N.R. Saxena, "Counting and Signatures are Almost Orthogonal", Proceedings of Technical Workshop: New Directions for IC Testing, Victoria, B.C., March 1986.

[28] F. Ruskey, personal communication.

[29] J. Savir, "Syndrome-Testable Design of Combinational Circuits", IEEE Trans. Comp., Vol. C-29, June 1980.

[30] J. Savir, "A New Empirical Test for the Goodness of Random Integer Generators", IBM Research Report RC 9446 (#41410), May 1982.

[31] J. Savir, G. Ditlow, and P. Bardell, "Random Pattern Testability", Proceedings of FTCS-13, Milano, Italy, June 1983.

[32] J. Savir, "Good Controllability and Observability Do Not Guarantee Good Testability", IEEE Trans. Comp., Vol. C-32, December 1983.

[33] J. Savir and W. McAnney, "On the Masking Probability with One's Count and Transition Count", Proceedings of ICCAD-85, Santa Clara CA, November 1985.

[34] N.R. Saxena, "Effectiveness Measures for Data Compression Techniques under Unequally Likely Error Model", Proceedings of Technical Workshop: New Directions for IC Testing, Victoria, B.C., March 1986.

[35] M. Serra and J.C. Muzio, "Testability by Sum of Syndromes", Proceedings of Technical Workshop: New Directions for IC Testing,

Victoria, B.C., March 1986.

[36] J. Smith, "Measures of the Effectiveness of Fault Signature Analysis", IEEE Trans. Comp., Vol. C-29, June 1980.

[37] A.S. Tanenbaum, *Computer Networks*, Prentice-Hall Inc., Englewood Cliffs NJ, 1984.

[38] E. Titchmarsh, *The Theory of Functions*, 2nd ed., Oxford University Press, 1939.

[39] T.H. Wonnacott and R.J. Wonnacott, *Introductory Statistics*, 3rd ed., John Wiley and Sons, New York NY, 1977.

[40] J.J. Zasio, "Non-Stuck Fault Testing of CMOS VLSI", Proceedings of IEEE Compcon, Spring, 1985.

[41] Y. Zorian and V.K. Agarwal, "Higher Certainty of Error Coverage by Output Data Modification", Proceedings of IEEE Int. Test Conf., October 1984.

[42] Y. Zorian and V.K. Agarwal, "A General Scheme to Optimize Error Masking in Built-In Self-Testing", Proceedings of Technical Workshop: New Directions for IC Testing, Victoria, B.C., March 1986.

Additional References for Circuits

[43] M. Abramovici and M.A. Breuer, "Deduction of Internal Signal Values in Combinational Networks Based Upon Test Results", Proceedings of FTCS-9, Madison WI, June 1979.

[44] A.N. Airapetian and J.F. McDonald, "An Improved Test Set Generation Algorithm for Combinational Circuit Control", Proceedings of FTCS-9, Madison WI, June 1979.

[45] Y. Crouzet, and C. Landrault, "Design of Self-Checking MOS-LSI Circuits Application to a Four-Bit Microprocessor", Proceedings of FTCS-9, Madison WI, June 1979.

[46] H. Eiki, K. Inagaki, and S. Yajima, "Autonomous Testing and its Application to Testable Design of Logic Circuits", Proceedings of FTCS-10, Kyoto, Japan, October 1980.

[47] E. Fujiwara, "A Self-Testing Group-Parity Prediction Checker and its Use For Built-In Testing", Proceedings of FTCS-13, Milano, Italy, June 1983.

[48] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits", Proceedings of FTCS-10, Kyoto, Japan, October 1980.

[49] R. Kallman, "A Fast 8-Bit Carry Circuit", IEEE Trans. Comp., Vol C-32, December 1983.

[50] B. Khodadad-Mostashiry, "Parity Prediction in Combinational Circuits", Proceedings of FTCS-9, Madison WI, June 1979.

[51] L.H. Pollard and J.H. Patel "Correction of Errors in Data Transmission Using Time Redundancy", Proceedings of FTCS-13, Milano, Italy, June 1983.

[52] circuit due to M. Serra, January 1986.

[53] S. Waser and M.J. Flynn, *Introduction to Arithmetic for Digital Systems Designers* Holt, Rinehart and Winston, New York NY, 1982.

[54] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, Addison Wesley Inc., Reading MA, 1985.

[55] X. Xu and E.J. McCluskey, "Test Generation and Fault Diagnosis for Multiple Faults in Combinational Circuits", Proceedings of FTCS-13, Milano, Italy, June 1983.

VITA

Surname: Aitken Given Names: Robert Campbell

Place of Birth: Vancouver, B.C. Date of Birth: April 21, 1963

Educational Institutions Attended, with Dates of Entering and Leaving:

University of Victoria, B.C.	1981 to 1986
	to
	to
	to

Degrees, Diplomas, Etc., Awarded, with Dates and Names of Institutions:

B.Sc. (Honours)	1985	University of Victoria, B.C.

Honors and Awards:

NSERC Post Graduate Scholarship, 1985/86

Publications:

R.C. Aitken et al., "Aliasing Probability of Some Data Compression Techniques", Proceedings of Technical Workshop: New Directions for IC Testing, Victoria, B.C., March 1986.

PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my thesis (the title of which is shown below) to users of the University of Victoria Library, and to make *single copies only* for such users or in response to a request from the library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Performance Measures of Data Compression Techniques in Fault Detection

Author [redacted]

Robert Campbell Aitken

APRIL 24, 1986

Date

Examiners:

[redacted]

Dr. Jon C. Muzio

[redacted]

Dr. Frank Ruskey

[redacted]

Dr. George A. Beer

[redacted]

Dr. Fayez H. El Guibaly