Pseudorandom-Pattern Test Resistance in High-Performance DSP Datapaths

Laurence Goodby† and Alex Orailoğlu‡
†Dept. of Electrical & Computer Engineering
‡Dept. of Computer Science & Engineering
University of California, San Diego
La Jolla, CA 92093

Abstract

The testability of basic DSP datapath structures using pseudorandom built-in self-test techniques is examined. The addition of variance mismatched signals is identified as a testing problem, and the associated fault detection probabilities are derived in terms of signal probability distributions. A method of calculating these distributions is described, and it is shown how these distributions can be used to predict testing problems that arise from the correlation properties of test sequences generated using linear-feedback shift registers. Finally, it is shown empirically that variance matching using associativity transformations can reduce the number of untested faults by a factor of eight over variance mismatched designs.

1 Introduction

Digital signal processing applications impose strict timing constraints on a test strategy. Often, even a single added layer of logic can significantly impair performance. In these applications, conventional built-in self-test techniques can result in unacceptable performance penalties. While the basic operations involved in high-performance signal processing (e.g., shifts, additions, multiplications, registers) are by themselves highly testable, their composition into larger systems often is not. Test insertion strategies that aim to improve the observability and controllability of these composite structures without knowledge of their underlying behavioral characteristics often result in unacceptable test overhead.

Here, we examine testability problems encountered in applying pseudorandom testing techniques to some fundamental DSP structures with the aim of identifying these problems early in the design process. A probabilistic analysis of the signal flow graph provides a means of locating pseudorandom testability problems prior to design synthesis. The designer can gauge the testability of a design early, allowing the possibility of restructuring the design in a way that improves its testability without impacting performance. Many of the test problems discussed here can be ameliorated by transformation of the signal flow graph, including width scaling and associative operator reordering.

From the perspective of the test designer, the analytical tools presented here enable more focused test insertion techniques by pinpointing the cause of a missed fault. In addition, these techniques can be used to examine the performance of linear-feedback shift-register (LFSR) based pseudorandom pattern generators (PRPGs)

This work was supported in part by grants from the Semiconductor Research Corporation (contract DJ-538), the University of California MICRO Program, and Hughes.

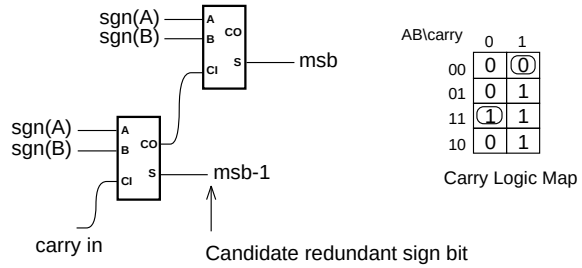


Figure 1: The redundant sign bit problem. The circled minterms are either don't-cares or difficult tests.

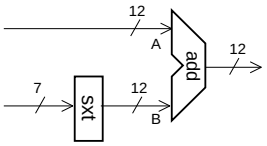
by modeling the correlation properties of the generated test sequences. This offers the possibility of objectively evaluating pseudorandom test generators based on their statistical and temporal properties. Furthermore, by providing insight into which tests are difficult to assert at the module level, it is possible to gauge the impact that the underlying gate-level fault representation has on reported fault coverage.

Over the past 20 years, the probabilistic behavior of logic circuits has been extensively studied [1], most recently in connection with estimation of power dissipation [2]. In linear networks, signal behavior can be characterized most efficiently through analysis of the signal flow graph, yielding insight into testing problems that is harder to discern at the gate level. For example, in DSP design it is common to encounter testing problems at the upper bits of adders. One reason for this is *redundant sign bits*, upper bits that always follow the most significant bit (MSB). The reason this is a test problem can be seen in Figure 1, which shows the top two adders in a ripple carry adder. The MSB and MSB-1 inputs to the adder are sign bits. If the output next-to-MSB bit always follows the output MSB, the carry logic of the next-to-MSB adder will be untestable. In some designs, the next-to-MSB output almost always follows the output MSB, but differs occasionally. In such designs the next-to-MSB adder carry logic is testable, but may be difficult to test using pseudorandom patterns.

Redundant sign bits can be identified using *scaling* [3, Sec. 6.9.2], a DSP design technique that is commonly used to adjust multiplier gains so that overflow is avoided. There is a close relationship between scaling and testability: scaling not only identifies redundant sign bits, but when these bits are removed, clears the path for other logic optimizations that remove redundant faults [4, 5]. Even after scaling, there may be upper adder bits that are "near-redundant", i.e. that almost always follow the MSB, and are correspondingly difficult to test. Furthermore, as the test signal passes through the datapath, it is altered in ways that may reduce its ability to assert some tests.

In Section 3, the observations regarding redundant sign bits will be extended to identify in a general way adder tests that are difficult. The probability of asserting the difficult tests will be derived using signal probability distributions, and variance matching is proposed as a technique for improving the random-pattern testability of designs. Section 4 will discuss how signal distributions

DAC 96 - 06/96 Las Vegas, NV, USA



$$\begin{array}{r} a_{11}\,a_{10}\,a_9\,a_8\,a_7\,a_6\,a_5\,a_4\,a_3\,a_2\,a_1\,a_0 \\ +\; b_6\;b_6\;b_6\;b_6\;b_6\;b_6\;b_5\;b_4\;b_3\;b_2\;b_1\;b_0 \\ \hline s_{11}\,s_{10}\,s_9\,s_8\,s_7\,s_6\,s_5\,s_4\,s_3\,s_2\,s_1\,s_0 \end{array}$$

Figure 2: An example of the variance gap problem.

can be computed in general networks consisting of adders, shifters, and registers. Section 5 will then show how this analytical technique can be applied to identify test problems that arise out of the correlation properties of linear-feedback shift-registers. Section 6 describes experimental results, including fault simulation results for four filters. The effectiveness of design transformations that improve testability are demonstrated, showing in excess of an order of magnitude decrease in the number of untested faults in these designs.

2 Notation & Conventions

We will assume two's-complement arithmetic throughout. Unless otherwise specified, an N-bit signal $b_0, b_1, \ldots, b_{N-1}$ is assumed to represent a number in the range range $-1 \leq x \leq 1 - 2^{-N+1}$, where x is given by

$$x = -b_0 + \sum_{i=1}^{N-1} b_i 2^{-i}$$

and b_0 is the MSB, b_{N-1} is the LSB. Although signals take on only discrete values, their probability mass functions are plotted here for clarity as continuous density functions on the interval [-1,1). The area under density functions is normalized to 1.

3 The Variance Gap Problem

Many high-performance signal processing datapaths consist primarily of networks of shift, add, and delay elements. In our examination of large DSP datapaths constructed out of these primitives, we found that most of the difficult faults were located in the upper bits of the carry chains of adders and subtractors. This seems to be consistent with the anecdotal accounts of designers, who report difficulty in fully exercising the carry chains of adders. We found the problem to be most severe in adders where signals of greatly differing amplitudes are combined.

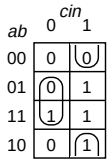
An example is shown in Figure 2, where a 7-bit number is added to a 12-bit number. The smaller *B* input is sign-extended to conform to the width of the adder, represented by the *sxt* operator in the register-transfer level (RTL) schematic. The *B* input may not be immediately recognizable as a 7-bit signal; for example, it might be a full 12-bit wide signal that only uses 7 bits of its dynamic range.

In an adder with a variance gap, it is difficult to generate the tests X10 and X01 at upper adder slices, where the cube specifies the values of the a_i, b_i, and carry inputs, respectively, and the B input is the lower variance input. The upper bits of the B input follow the sign of B. If a particular test requires the carry input to an upper adder slice to differ from the sign of B, a number of the upper bits of the A input will be constrained. This is shown in Figure 3, where the test 001 is to be applied to the next-to-MSB full adder. The MSB adder is assumed to not generate a carry.

This test is *difficult* in the sense that it is dependent on the difference between the input widths, and requires fairly specific values to appear at the *A* input. If the *A* input values are uniformly distributed,

$$\begin{array}{r} 0\;\boxed{1}\;1\;1\;1\;1\;c_5\;c_4\;c_3\;c_2\;c_1 \\ a_{11}\;\boxed{0}\;1\;1\;1\;1\;a_5\;a_4\;a_3\;a_2\;a_1\;a_0 \\ +\;0\;\boxed{0}\;0\;0\;0\;0\;b_5\;b_4\;b_3\;b_2\;b_1\;b_0 \\ \hline s_{11}\;1\;0\;0\;0\;0\;s_5\;s_4\;s_3\;s_2\;s_1\;s_0 \end{array}$$

Figure 3: Implications of applying test 001 to bit 10 of variance mismatched adder (5 bit variance gap). Demonstrates the difficulty of testing the carry logic of upper bits in variance mismatched adders.



test	*A* vector
0 0 1	X 0 1 1 1 1 X X $\cdots$
0 1 0	X 0 0 0 0 0 X X $\cdots$
1 1 0	X 1 0 0 0 0 X X $\cdots$
1 0 1	X 1 1 1 1 1 X X $\cdots$

Figure 4: Difficult tests.

the waiting time for the test follows a geometric distribution, the mean of which is exponentially dependent on the number of bits that span the gap between the input widths. The test time to reach full coverage can be expected to double for each additional bit of variance gap.

It is not always necessary to generate all the difficult tests, since there may exist an easier test for the failure mode in question, and not all difficult tests may be essential. For example, if a failure mode of a full-adder has the effect of removing the cube 11X from the on-set of the carry function, the easier test 111 can be chosen over the difficult test 110. Which tests are essential is dependent on the adder structure and fault model used. Under most gate-level single-stuck fault models of full-adders, testing the carry logic presents the greatest difficulty since the carry logic tends to have essential tests that are difficult, while the sum logic fault model generally includes an easier test among the alternative tests for each fault.

The difficult tests are shown circled in Figure 4, along with the logic function for the carry logic. The *A* input values required to assert each difficult test are shown in the accompanying table.

At a minimum, the failure modes of the carry logic typically include faults where the carry output on-set expands to include the cubes X1X or XX1. For these failure modes, the difficult tests 010 and 001 are essential. A gate-level model for such an adder is shown in Figure 5.

Two common full-adder gate-level models are shown in Figures 5 and 6. The first design is optimistic under the single-stuck fault model in that two of the four difficult tests are non-essential, and consequently it is likely that these tests will not be applied even if 100% fault coverage is reported by fault simulation. The table shows the essential tests for the carry logic (labeled "e"), and the test equivalence classes for test cubes that do not include an essential test (labeled "1" and "2"). The carry logic is fully tested if each of the three essential tests are applied and a test from each of the two equivalence classes is applied. The second design is more conservative from a testing perspective, in that all four difficult tests are required. The first circuit may be too optimistic for use as a general full-adder model in fault simulation; if test generation is stopped as soon as 100% coverage is reported, it is quite possible that some adders with large variance gaps will not have had the 110 and 101 tests applied, which may be required to detect real faults in fabricated devices.

Thus, it is possible to identify—relatively independent of implementation structure—that (at least) the tests 010 and 001 will be problems in an adder where signals of widely differing amplitudes are combined.

The testing problem described here is compounded by another

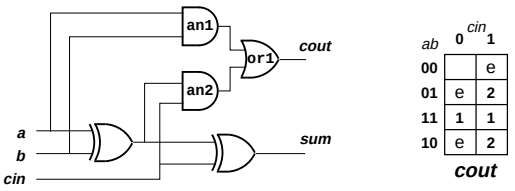


Figure 5: Adder design 1 and associated tests.

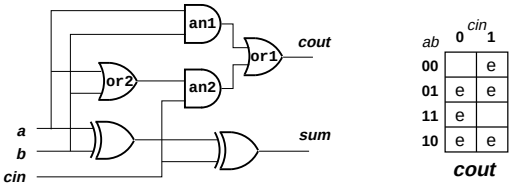


Figure 6: Adder design 2 and associated tests.

effect that commonly arises in DSP datapaths. Rather than the adder *A* input taking on a uniform distribution of values, samples tend to be more densely distributed around the mean. This is a consequence of the Central Limit Theorem [6]; at internal nodes signals will tend to take on a normal distribution. This is illustrated in Figure 7, where a continuous density function has been used to represent the discrete probability function of the *A* input. The figure shows the approximate locations of vectors that will assert the indicated difficult tests. As the variance gap is increased, the width of each of the eight gray regions narrows. Fortunately, the tests that would be asserted at the tails of the input distribution (010, 101) are also asserted by vectors that lie close to the mean. The problem is typically worse for the tests that lie around ± 0.5, at least if the standard deviation (σ) of the signal is small compared to the full dynamic range available. The problem can be much worse for certain test signal generators, as will be shown in Section 5.

3.1 Fault detection probabilities

The previous section gave conditions required for the difficult tests to be asserted at the upper bit slices of a variance-mismatched adder. These conditions, while necessary, are not sufficient to ensure that the tests are asserted. The additional conditions are on the sign of the *B* (low variance) input and the carry output of the lower bits, which will be referred to as the *lower block* (bits 0–5 in the example). Specifically, the probability of a single vector asserting a difficult test at the next-to-MSB adder in an N-bit variance-mismatched adder with variance gap of M bits is

$$\begin{aligned} p_D &= P\{a_{N-2} = k_1, b_{N-2} = k_2, c_{N-2} = k_2'\} \\ &= P\{A \in V(k_1, k_2), c_{lb} = k_2', \text{sign}(B) = k_2\} \end{aligned}$$

where $k_1, k_2 \in \{0, 1\}$, c_{lb} is the carry out of the lower block, sign(*B*) is the value of the sign bit of the *B* input, and $V(k_1, k_2)$ is a cube determined as follows:

$$V(k_1, k_2) = \mathtt{X} k_1 \overbrace{k_2' \cdots k_2'}^{M-1} \overbrace{\mathtt{X} \cdots \mathtt{X}}^{N-M-1}$$

If the *B* input is fed by a uniform distribution, such as generated by an LFSR, $P\{\text{sign}(B) = k_2\} = 0.5$. Approximating the distribution of the *A* vectors in the lower block by a uniform distribution, independent of the upper bits, the probability that the lower block generates a carry is the same as the probability that the sum of two uniformly distributed, unsigned L-bit binary numbers is greater than or equal to 2^L, where L is the width of the lower block ($L = N - M - 1$):

$$P\{c_{lb} = 1\} = \frac{2^L - 1}{2^{L+1}}$$

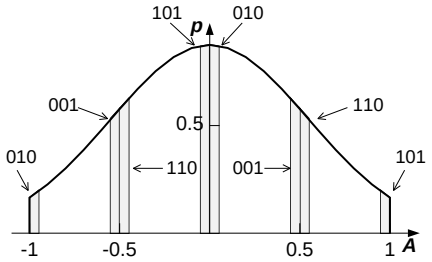


Figure 7: A hypothetical probability density function for the A input to a variance mismatched adder showing the zones the input vector must fall in for the difficult tests to be asserted.

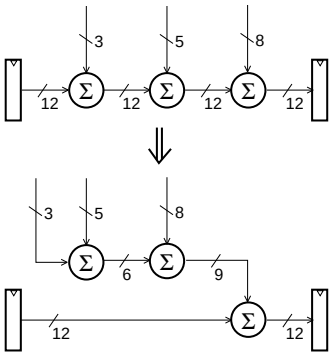


Figure 8: Variance matching transformation.

This function is 0.25 for $L = 1$, and rapidly increases towards its upper bound of 0.5 as L is increased.

p_D gives the probability of a test being applied at a given vector, representing a Bernoulli trial with probability of success $p = p_D$. The probability of first asserting the desired test at vector n follows a geometric distribution, $p_D(n) = q^{n-1}p$ (where $q = 1 - p$, $n = 1, 2, \ldots$). The expected number of vectors for first assertion of the test is $1/p$, with variance q/p^2.

For the earlier example, $N = 12$, $M = 5$, $L = 6$, $k_1 = 0$, $k_2 = 0$, giving

$$\begin{aligned} p_D &= P\{A \in \mathtt{X01111XXXXXX}, c_{lb} = 1, \text{sign}(B) = 0\} \\ &\approx 0.25\left(P\{A \in \mathtt{001111XXXXXX}\}\right. \\ &\qquad \left. + P\{A \in \mathtt{101111XXXXXX}\}\right) \end{aligned}$$

If the *A* input is uniformly distributed, $p_D = 0.25 \cdot 2^{-5}$, and the mean waiting time for application of the test is 128 vectors, with standard deviation $\sigma = 127$. In realistic applications, the uniform assumption breaks down, and it is necessary to either calculate or estimate the probability distributions of the inputs. Calculation of probability distributions will be discussed in Section 4.

3.2 Variance matching transformation

The variance gap problem can sometimes be reduced by restructuring additions using associativity. Ideally, the two smallest variance sources to be added are combined first, followed by the next two, and so on, in a manner analogous to the construction of a Huffman code tree. From a layout perspective this approach has drawbacks due to the potential irregularity of the resulting layout. Often, a linear chain of adders is preferred to a tree structure since this maximizes the regularity of the layout.

In large filters, it is common to have a high variance datapath into which smaller variance signals are added. In this case, one possible compromise between regularity and testability is to add small variance signals in their own chain before adding the result

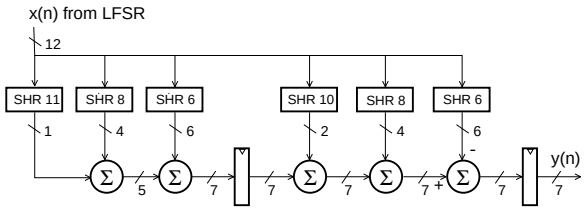


Figure 9: Scaled taps 64 and 63 from the filt64 example. The signal y(n) is the output from tap 63; the test signal is applied at x(n).

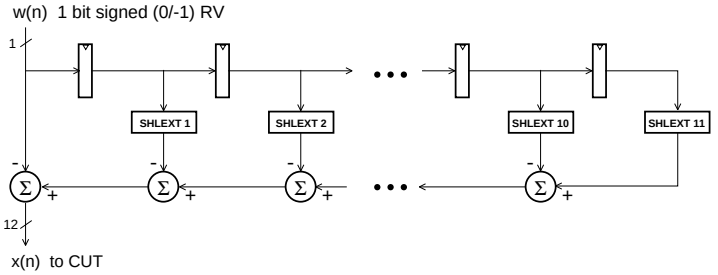


Figure 10: Modeling the correlation properties of a 12-bit LFSR, where the LFSR output is interpreted as a two's complement number. All subtractors are 12 bits wide with inputs sign-extended.

into the larger variance chain. This approach is illustrated in Figure 8. The top design will be referred to as the chain architecture, while the lower design will be referred to as the variance-matching architecture. The effect of this transformation on the pseudorandom testability of the design will be examined empirically in Section 6.

4 Computing Probability Distributions

For signal processing datapaths consisting of networks of shift, add, and delay elements, it is possible to efficiently compute the signal probability distribution at any adder output using the convolution property of discrete, lattice-type random variables (RVs) [6].

Given the probability distributions of two discrete, lattice-type independent RVs, X and Y, $p_X(n) = P\{X = n\}$, $p_Y(n) = P\{Y = n\}$, the distribution of the sum is given by the linear convolution of the distribution functions, $p_{X+Y} = p_X(n) * p_Y(n)$. In two's-complement arithmetic, the addition is performed using normal unsigned integer arithmetic, modulo 2^N. To account for the effect of overflow, L-point circular convolution is used in place of linear convolution, denoted $p_X(n) \textcircled{L} p_Y(n)$, $L = 2^N$,

$$p_{X+Y}(n) = \sum_{i=0}^{L-1} p_X(i) p_Y((n-i) \bmod L),$$

for $n = 0, 1, \ldots, L-1$.

The circular convolution can be computed efficiently using the Discrete Fourier Transform (DFT) or the Fast Fourier Transform (FFT),

$$p_{X+Y}(n) = \mathcal{DFT}^{-1}\{P_X(k) P_Y(k)\}$$

where $P_X(k) = \mathcal{DFT}\{p_X(n)\}$, $P_Y(k) = \mathcal{DFT}\{p_Y(n)\}$.

Thus, when adder inputs are independent, the output distribution can be most efficiently computed in terms of the input distributions. This is applicable to acyclic networks where adder outputs do not reconverge.

A more general model assumes that the network is acyclic, but that adder inputs are not independent (e.g., adder outputs may reconverge). In this case, the impulse response is computed for the adder's output (assuming a linear network), and the output probability distribution is computed from this. This is done by generating a distribution corresponding to each non-zero component of the impulse response, and convolving the results (or, more efficiently, multiplying together the FFTs of each distribution and taking the inverse FFT of the product). The generated distributions are simply suitably scaled versions of the PRPG source distribution. This approach assumes a single PRPG source, but can be extended to the case of multiple independent PRPGs by convolving the distributions computed for each independent source.

In the case of cyclic networks, the distributions can generally be approximated by truncating infinite impulse responses at a point where the energy in the tail is small compared to the total energy of the impulse response.

For the experimental studies described in this paper, we have implemented the single-source, acyclic version of the algorithm that supports reconvergent adder outputs.

DFT resolution and computational considerations: For wide signals, there is an issue of how many bits to use in the DFT representation of the signal. For our tests, we have found that 8 to 10 bits of resolution show most of the detail in the distributions, and should be adequate for estimating fault detection probabilities at the upper bits of adders, where stubborn faults are typically found. For networks with no reconvergent adder outputs, the DFT only needs to be computed twice and the inverse DFT once for each adder. For adders on reconvergent paths, the impulse response approach is used, requiring one DFT for each non-zero component of the impulse response, followed by one inverse DFT. This is potentially computationally expensive if a filter has a long impulse response. For the large filt64 example that will be introduced in Section 6, the impulse response method was found to use 66 CPU seconds on a 486-66MHz processor for 256-point (8-bit resolution) FFTs.

Nonlinear operations: The discussion here assumes a linear network model. However, truncation is a common non-linear operation found in DSP applications. This can be handled under a linear network model by representing truncation (or, equivalently, right-shifting) as division by a power of two combined with a noise source. For the experiments here, we ignore the noise introduced by truncation.

5 Modeling LFSR Correlation Properties

Common pseudorandom pattern generators (PRPGs) based on LFSRs do not produce statistically independent test vectors; significant correlation exists between successive tests. A typical LFSR-based PRPG might shift its contents from LSB to MSB for each test, introducing a new bit at the LSB. For this type of PRPG, each output sample (interpreted as a two's-complement number) is closely related to twice the preceding sample. The sign-extended output of an N bit LFSR can be expressed in terms of its previous output as

$$x[n+1] = \begin{cases} 2x[n] + \delta, & -2^{N-2} \le x[n] < 2^{N-2} \\ 2x[n] - 2^N + \delta, & 2^{N-2} \le x[n] < 2^{N-1} \\ 2x[n] + 2^N + \delta, & -2^{N-1} \le x[n] < -2^{N-2} \end{cases}$$

The first case occurs on average 50% of the time, with the other two cases each occuring 25% of the time (δ is the one-bit unsigned 0/1 signal shifted into the LFSR LSB). This representation shows why correlation effects might be a problem in a datapath structure: for example, if the datapath implements the function

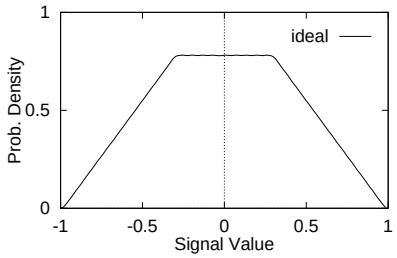


Figure 11: Theoretical probability distribution of y(n) in Figure 9, assuming an ideal PRPG providing statistically independent test vectors.

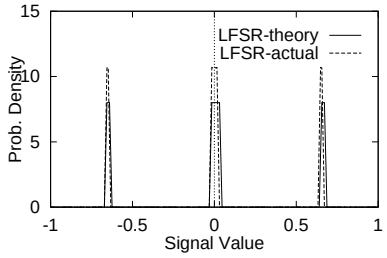


Figure 12: Theoretical and measured (histogram estimate) probability distribution of y(n) in Figure 9 for a 12-bit LFSR. The theoretical curve uses the LFSR model shown in Figure 10. Compared with Figure 11, this demonstrates the strong effect of LFSR correlation properties on signal distributions.

$y[n] = 41x[n-1] - 22x[n]$ (corresponding to Figure 9, if truncation is ignored), the output signal distribution will degenerate into narrow regions given by

$$y[n] \approx \begin{cases} -3x[n], & -2^{N-2} \le x[n] < 2^{N-2} \\ -3x[n] - 22 \cdot 2^N, & 2^{N-2} \le x[n] < 2^{N-1} \\ -3x[n] + 22 \cdot 2^N, & -2^{N-1} \le x[n] < -2^{N-2}, \end{cases}$$

where the typically small effect of δ has been neglected.

For PRPGs generating statistically independent samples, signal distributions are typically smooth after a few signals dependent on different PRPG samples are combined. However, PRPG correlation effects can destroy this property, introducing a large amount of fine structure in the signal distributions. This sort of correlation effect can be examined using the analytical techniques described in Section 4, where the PRPG input signal is replaced by a linear model of an LFSR, shown in Figure 10. The input to the LFSR model, $w(n)$, is an independent RV taking values 0 and -1, each with probability 0.5. The SHLEXT operator shifts its input left by the indicated amount, producing a signal wide enough to hold the shifted quantity.

For the datapath segment shown in Figure 9, an idealized PRPG producing statistically independent samples would produce the output distribution shown in Figure 11. By replacing the input to the circuit with the LFSR model of Figure 10, the curve labeled "LFSR-theory" in Figure 12 is produced, in agreement with the histogram produced by simulating the actual LFSR sequence. When compared with the test zones indicated in Figure 7, it can be seen that this signal would not be able to effectively test an adder with even a small variance gap in the following filter stage since tests 001 and 110 would not be activated.

This problem exists independent of the LFSR seed and polynomial (at least for the class of LFSRs that use external XOR feedback networks). This result shows how signal distributions are able to provide insight into testing problems that cannot be identified with gross measures, such as signal variance or maximum signal range.

design	adders	regs	widths		
			in	coef.	out
filt11	15	11	16	10	16
filt25	44	25	12	14	14
filt60	173	60	12	14	14
filt64	193	64	12	14	14

Table 1: Design statistics.

design	adder scaling		register scaling	
	#add.	#bits	#reg	#bits
filt11	7	30	4	20
filt25	15	44	9	26
filt60	49	161	18	63
filt64	91	281	30	97

Table 2: Scaling of chained-adder architecture.

design	adder scaling		register scaling	
	#adds	#bits	#regs	#bits
filt11	8	36	4	20
filt25	26	80	9	26
filt60	123	640	18	63
filt64	157	1016	30	97

Table 3: Scaling of variance-matching architecture.

design	orig	scaled	VMA
filt11	5370	4590	4316
filt25	13496	12380	10926
filt60	52670	48814	34450
filt64	58726	52024	31476

Table 4: Number of adder faults simulated.

6 Experimental Results

Four filter specifications were selected from the literature: a 64-tap filter [7], a 60-tap filter [8], a 25-tap filter [8], and an 11-tap filter [9]. The design statistics are shown in Table 1, including the number of adders, the number of state registers, the input signal width, the coefficient width, and the output signal width. Ripple adder chain structures were used to implement the fixed width datapath baseline designs, where all addition and subtraction operations are the width of the filter output. Scaling was then applied to remove redundant sign bits, shrinking portions of the datapath and enabling further redundancy elimination using logic optimizations [4, 5].

The scaling results are shown in Table 2, where the number of adders and registers scaled is shown, along with the number of adder and register bits removed via scaling. L_1 scaling was used [3, Sec. 6.9.2], which guarantees that the behavior of the design is not changed. For comparison purposes, the designs were also constructed using the variance matching architecture described in Section 3.2. The scaling results for the variance matching architecture are shown in Table 3.

The resulting designs were fault simulated using LFSR-generated test vectors. The total number of adder faults simulated in each design is shown in Table 4. Registers in these designs are highly testable, consequently their faults are excluded from consideration here. The fault simulation curves are plotted in Figures 13–16, showing the number of untested faults for the original (unscaled) chain architecture, the scaled chain architecture, and the scaled variance matching architecture. Variance matching results are not plotted for filt11 since it had too few adders per tap to allow significant optimization.

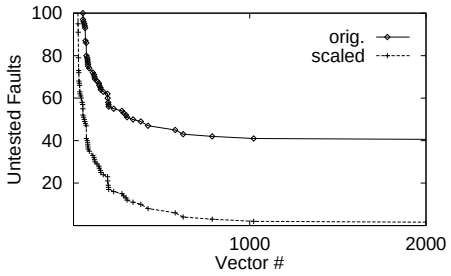


Figure 13: Fault coverage for the filt11 design, original (non-scaled) design, and the corresponding scaled design. In these figures, an LFSR-based PRPG provides the input test signal, and it is assumed that there is no aliasing in the output signature compressor.

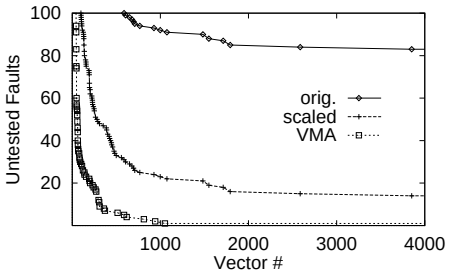


Figure 14: Fault coverage for the filt25 design. Original (non-scaled), scaled, and variance-matched (VMA) versions shown.

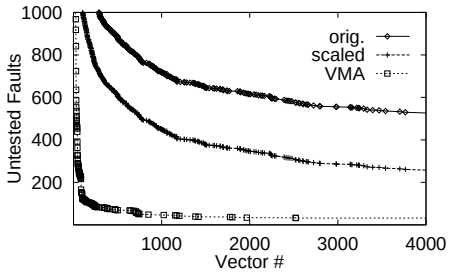


Figure 15: Fault coverage for the filt60 design, original, scaled, and variance matched versions.

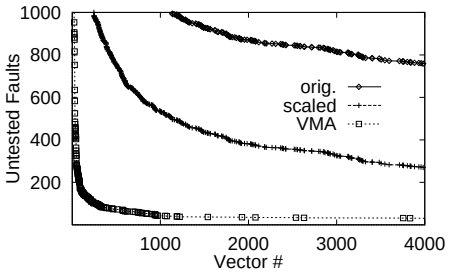


Figure 16: Fault coverage for the filt64 design, original, scaled, and variance matched versions.

All the designs were found to be highly testable in terms of percent fault coverage (all in the high 90s) but the original unscaled designs using the adder chain architecture have a significant number of untested faults even after several thousand test vectors (on average as many as 4 per adder in filt64). Redundancy elimination using scaling and logic optimization significantly reduced the number of untested faults in these designs, but a large number of stubborn faults remained in the cases of filt60 and filt64, as indicated by the high rate at which faults are still detected after 2000 vectors have been applied. For these filters, the variance matching architecture yielded significant gains in terms of reduced test time and fewer untested faults. In all designs, the best optimized design offered more than an order of magnitude reduction in the final number of untested faults over the unoptimized design. The results are summarized in Table 5 in terms of the average number of untested faults per adder after applying a maximum-length LFSR sequence (or 4k vectors for filt11).

arch	filt11	filt25	filt60	filt64
orig	2.67	1.87	3.03	3.93
scaled	0.07	0.32	1.48	1.40
VMA	–	0.02	0.18	0.16

Table 5: Final average untested faults per adder.

7 Conclusion

The testability properties of large DSP datapath structures under a pseudorandom self-test paradigm have been examined. Addition of variance-mismatched signals was identified as a testing problem, and the probability of detecting difficult faults in these structures was derived in terms of signal probability distributions. A method of calculating these distributions was described, and its ability to predict testing problems associated with LFSR correlation properties was demonstrated. Variance matching was empirically shown to improve the testability of the two largest designs, reducing the number of untested faults by at least a factor of eight over the scaled and optimized designs that did not use variance matching. In addition, test length was significantly reduced.

References

[1] K. P. Parker and E. J. McCluskey, "Analysis of logic circuits using input signal probabilities," in *Intl. Symp. on Fault-Tolerant Comp.*, pp. 1.8–1.12, 1974.

[2] J. Monteiro, S. Devadas, and B. Lin, "A methododogy for efficient estimation of switching activity in sequential logic circuits," in *Design Automation Conference*, pp. 12–17, 1994.

[3] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing.* Prentice Hall, 1989.

[4] L. Goodby and A. Orailoğlu, "Towards 100% testable FIR digital filters," in *International Test Conference*, pp. 394–402, Oct. 1995.

[5] L. Goodby and A. Orailoğlu, "Synthesizing self-testable filters via scaling and redundant operator elimination," in *29th Asilomar Conf. Signals, Systems, and Computers*, Oct. 1995. In Press.

[6] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw Hill, 3rd ed., 1991.

[7] T. Yoshino, R. Jain, P. T. Yang, H. Davis, W. Gass, and A. H. Shah, "A 100-MHz 64-tap FIR digital filter in 0.8-μm BiCMOS gate array," *IEEE J. of Solid-State Circuits*, vol. 25, pp. 1494–1501, Dec. 1990.

[8] H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Trans. on Circuits and Systems*, vol. 36, pp. 1044–1047, July 1989.

[9] R. Jain, P. T. Yang, and T. Yoshino, "FIRGEN: A computer-aided design system for high performance FIR filter integrated circuits," *IEEE Trans. Signal Processing*, vol. 39, pp. 1655–1668, July 1991.