Spectral RTL Test Generation for Microprocessors

Nitin Yogi and Vishwani D. Agrawal
Auburn University, Department of Electrical and Computer Engineering, Auburn, AL 36849, USA
yoginit@auburn.edu, *vagrawal@eng.auburn.edu*

Abstract

We introduce a novel method of test generation for microprocessors at the RTL using spectral methods. Test vectors are generated for RTL faults, which are the stuck-at faults on inputs-outputs of the different modules/registers in the circuit, and the vectors are analyzed using Hadamard matrices for Walsh functions and the random noise level at each primary input. This information then helps generate vector sequences. At the gate-level, a fault simulator and an integer linear program (ILP) compact the test sequences. RTL test appraisal also helps reveal the hard-to-test parts of the circuit. An XOR observability tree was used to improve the testability of those parts. We give results for a simple accumulator-based processor named Parwan. The RTL spectral vectors produced higher coverage in shorter CPU times as compared to a gate-level ATPG.

1. Introduction

With the current generation microprocessors becoming faster and more complex, new challenges are been faced in their testing process. Not only has the test generation problem grown intricate, but also new requirements are been placed on the test generation methods like testing of delay faults. Hence functional at-speed tests are been considered to replace conventional scan tests to solve some of the testing problems. Different works [32, 33, 36] have attempted to show the effectiveness of functional tests over structural scan tests in detecting chip faults.

Functional test generation for microprocessors has been an active research area [4, 41, 44]. However, functional tests achieve low fault coverage as they are not related to the stuck-at fault model. Recent approaches [19, 31] map pre-computed module level test sets to processor instructions. These methods use data structures like algebraic decision diagrams and bounded model checking to solve for instruction and circuit imposed constraints and hence require more engineering effort. Built-in Self Test (BIST) for microprocessors has also been an area of active research for processor testing. Conventional pseudo-random tests for logic BIST have been found to give low fault coverage due to the inherent random pattern resistance of microprocessors. Methods like deterministic BIST [22, 11] and weighted random patterns [5, 20, 47] have been tried to increase the fault coverage. Randomized instructions have also been used for testing microprocessors [2, 37]. One problem with using randomness as a basis of test is, it might require a very large number of instructions to achieve the required fault coverage. Also hardware-based BIST requires the circuit to be made BIST-ready. Insertion of BIST logic may require tedious design changes and may also impact the area and performance of the circuit. Software BIST [6, 30] has been proposed as an alternative to hardware BIST, which involves microprocessor testing using instructions loaded in program memory. Chen and Dey [6] develop pseudo-random test patterns for each processor component under the constraints imposed by the instruction set which are then applied using a software-emulated LFSR as tests to the microprocessor. Kranitis *et al.* [30] apply a similar approach; however they use deterministic patterns as opposed to pseudorandom patterns and use RTL information for mapping each pattern to a processor instruction. Both of these approaches require a good knowledge of the instructions affecting each component and the analysis of controllability and observability of the component terminals in connection with each of the instructions.

In this paper we present a spectral method of processor testing using RTL faults. Unlike some of the methods discussed earlier, the engineering effort required is low as our method is simulation-based. Also we target RTL faults, which brings with it the advantage of low testing complexity. Our method is based on spectral testing, on which various papers have been published. In 1983, Susskind [40] showed that Walsh spectrum can be used for testing a digital circuit. General properties and applications of digital spectra can be found in the published literature [3, 14, 24, 43]. Hsiao and Seth [23] further expanded that work to compact testing. More recently, Giani *et al.* [17, 18] have reported spectral techniques for sequential ATPG and built-in self-test. Hsiao's group at Virginia Tech has published further work on spectrum-based self test and core test [7, 8, 28]. Khan and Bushnell [29] have designed hardware signature analyzers using spectral components. Zhang *et al.* [49] further refined the method of extracting the spectra from a digital signal using a selfish gene algorithm. Recent work suggests that wavelet transforms can also be used for similar application [10]. Yogi and Agrawal [48] introduced a spectral RTL test generation method for sequential circuits. In this paper we expand that methodology to test microprocessors.

We sample synthesis independent RTL faults, which are all the faults on the inputs-outputs of the different modules in the circuit and inputs-outputs of the registers. The RTL vectors generated to cover these faults, in general, detect around 70 to 80% of all detectable faults in the synthesized circuit. Our spectral analysis determines the prominent digital function components and the noise level in the RTL vectors. Vector sequences generated from these properties are found to give good fault coverage results with comparable test lengths and lower test generation times as compared to a gate-level sequential ATPG.

The outline of the paper is as follows. Section 2 gives an overview of how bit-streams can be analyzed in the spectral domain. In Section 3 we present our method of RTL ATPG using spectral analysis and discuss its application on a processor circuit in Section 4. Section 5 describes an RTL design for testability (DFT) method for improving the fault coverage. Results are discussed in Section 6 and finally we conclude in Section 7.

2. Background

The spectral method of test generation is based on the premise that the spectrum of vectors that detect high-level faults of the circuit exhibit certain spatial and temporal correlations among the bits of primary input vectors, required to sensitize paths between primary inputs and outputs of a sequential circuit. However, any high level test sequence has, besides the relevant spectra, some amount of noise, which corresponds to the don't care bits in the tests of target faults. So we analyze the spectrum and the noise level, and then generate new vectors using the spectrum, to which noise samples are added.

We shall use Walsh functions [45] to analyze the spectrum because they have been used for testing with effective results. Walsh functions are a set of orthogonal functions. They consist of trains of square pulses having +1s and -1s as the allowed states and can only change at fixed intervals of a unit time step. For an order n, i.e., for a sequence of n time steps, there are 2^n Walsh functions given by the rows of a $2^n \times 2^n$ Hadamard matrix $H(n)$ [45], when arranged in the so-called "sequency" order [42, 46].

Hadamard matrices can be generated using the following recurrence relation:

$$H(n) = \begin{bmatrix} H(n-1) & H(n-1) \\ H(n-1) & -H(n-1) \end{bmatrix} \qquad (1)$$

where $H(0) = 1$ and 2^n is the dimension of the nth order Hadamard matrix, $H(n)$. Any bit-stream of k bits can be represented as a linear combination of the basis bit-streams from the Hadamard matrix, $H(\log_2 k)$, where the multiplicand used for a basis bit-stream is the projection of the object bit-stream on that basis bit-stream. We shall refer to them as coefficients. By analyzing these coefficients we will be able to determine the major contributing basis bit-streams in an original signal.

3. Spectral RTL ATPG

Our approach to RTL test generation consists of two principal steps:

1. RTL spectral characterization
   - Test generation for RTL faults
   - Spectral analysis
2. Gate-level test generation
   - Spectral vector generation
   - Vector sequence compaction and Coverage

3.1 Test Generation for RTL Faults

The RTL faults considered are the stuck-at faults on primary inputs and outputs of the different modules/circuit and on inputs and outputs of all flip-flops. Test vectors are generated to detect these faults.

3.2 Spectral Analysis

The generated vectors are analyzed using Hadamard matrix to find the major spectral components. The bit-streams entering various inputs are analyzed separately. The 0s and 1s in a bit-stream are represented as -1s and $+1$s respectively. To find the spectral components for a bit-stream, it is multiplied with the Hadamard matrix; which is a correlation operation of the bit stream with each of the basis bit-streams. A high value of the coefficient corresponds to a high correlation of the bit-stream to the corresponding basis bit-stream and vice-versa. Hence, basis bit-streams exhibiting high coefficient values are considered as important or essential components and others are considered as noise.

Figure 1 shows an example of generation of coefficients by projecting a bit-stream onto the basis bit-streams and determining the essential component(s). As shown, we obtained a single coefficient with high correlation, which we shall treat as an essential component and others will be treated as noise. Currently essential components are differentiated from noise components by a gradually increasing threshold level.

$$\begin{array}{ccccccccc}
\begin{matrix}1\\0\\1\\1\\1\\0\\1\\0\end{matrix} & 0\to -1 & \begin{matrix}1\\-1\\1\\1\\1\\-1\\1\\-1\end{matrix} & \begin{matrix}\textit{Spectral}\\ \textit{decomposition}:\end{matrix} & \begin{bmatrix}1&1&1&1&1&1&1&1\\1&-1&1&-1&1&-1&1&-1\\1&1&-1&-1&1&1&-1&-1\\1&-1&-1&1&1&-1&-1&1\\1&1&1&1&-1&-1&-1&-1\\1&-1&1&-1&-1&1&-1&1\\1&1&-1&-1&-1&-1&1&1\\1&-1&-1&1&-1&1&1&-1\end{bmatrix} & \begin{bmatrix}1\\-1\\1\\1\\1\\-1\\1\\-1\end{bmatrix} & = & \begin{bmatrix}2\\\mathbf{6}\\-2\\2\\2\\-2\\-2\\2\end{bmatrix} \\
\begin{matrix}\text{bit}\\\text{stream}\end{matrix} & & \begin{matrix}\text{modified}\\\text{bit}\\\text{stream}\end{matrix} & & \begin{matrix}\text{Hadamard matrix}\\H(3)\end{matrix} & \begin{matrix}\text{modified}\\\text{bit}\\\text{stream}\end{matrix} & & \begin{matrix}\text{spectral}\\\text{component}\\\text{magnitudes}\end{matrix}
\end{array}$$

Figure 1. Spectral analysis of a stream of 8-bits. The essential Walsh component in this bit-stream has magnitude 6 and is represented by the second row of Hadamard matrix, $H(3)$.

$$\text{(a) Perturbing spectra}: \begin{bmatrix}2\\\mathbf{6}\\-2\\2\\2\\-2\\-2\\2\end{bmatrix} \to \begin{bmatrix}1\\\mathbf{6}\\2\\-1\\3\\-2\\3\\-1\end{bmatrix}$$

(b) New bit − stream obtained from perturbed spectra :

$$\begin{aligned}&Sign\{[1\ \ \mathbf{6}\ \ 2\ \ -1\ \ 3\ \ -2\ \ 3\ \ -1]\times H(3)\}\\ &=[1\ \ 1\ \ 1\ \ -1\ \ 1\ \ -1\ \ 1\ \ -1]\to[1\ 1\ 1\ 0\ 1\ 0\ 1\ 0]\end{aligned}$$

Figure 2. Bit-steam generation by perturbing the spectra. Note that the essential component having a magnitude 6 is not perturbed.

3.3 Spectral Vector Generation

After spectral analysis of the RTL vectors, to generate test vectors for gate-level faults, the essential spectral coefficients are retained and others, being considered noise are perturbed in a confidence range in terms of magnitude and/or in phase to generate new coefficients. The confidence levels correspond to the amount of randomness to be added.

Test vectors can then easily be generated from the coefficients by multiplying the coefficients with the Hadamard matrix again. Figure 2 shows an example of perturbation of the spectra and reconstruction of test vectors. We generate M such test sets by perturbing the spectra. Since we add random noise, this variation gives different characteristics to each vector set facilitating detection of various gate-level faults.

3.4 Sequence Compaction and Coverage

We generate perturbation vectors sequences, V_1, V_2, $\cdot\ \cdot\ \cdot$, V_M, as described in the previous subsection, such that their coverage as determined by fault simulation of the gate-level circuit either reaches some target value or simply saturates. Next, we compact the test by selecting the smallest number of these sequences without reducing the coverage. Our compaction is done by an integer linear program (ILP), in a similar way as has been reported in the literature [12, 15, 27, 48].

4. A Microprocessor Example

We applied our RTL-spectral based test method to a simple accumulator-based processor named Parwan [35] to demonstrate our method's effectiveness. As shown in the schematic diagram of PARWAN processor in Figure 3, it includes the following components: accumulator (AC), arithmetic logic unit (ALU), shifter unit (SHU), status register (SR), instruction register (IR), program counter (PC), memory address register (MAR) and a control unit (CTRL). It has a 8-bit databus with a 12-bit address bus (addressing 4K memory). The circuit has around 800 gate modules and 53 flip-flops. Currently our method cannot handle bidirectional pins. Hence we had to split the bidirectional buses into separate input and output buses. Also we added a reset signal to initialize the circuit for testing.

The RTL fault set consisted of 737 stuck-at faults, which were all faults on the inputs and outputs of the different components (e.g., ALU, SHU, etc.), inputs and outputs of registers (e.g., IR, PC, etc.) and on the tri-state drivers. Vectors were generated to cover these RTL faults using a commercial sequential ATPG and new vectors were generated using the technique described in Section 3.

5. Design for Testability

Test generation for only the RTL faults has an additional advantage other than characterizing the circuit in our case. It reveals the bottlenecks in the testability of the circuit. Analysis of hard to detect faults at

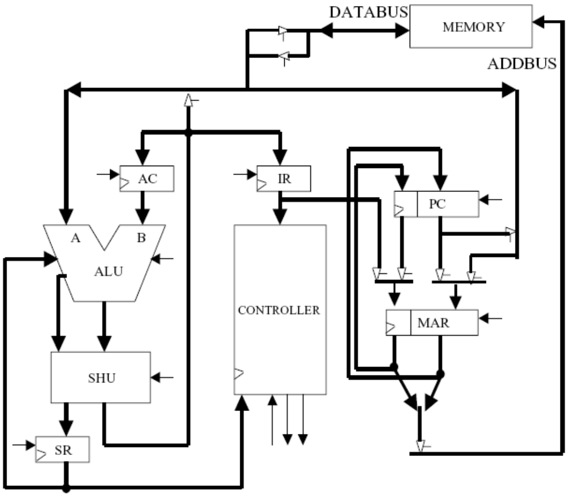

Figure 3. Parwan CPU.

the RTL gives an idea of the hard to test parts of the circuit. Hence, by increasing the testability of hard to test faults, we can expect to increase the testability of the overall circuit. The testability of a signals is improved by increasing its controllability and/or observability. Several methods [21, 25] of design for testability add control and observation points. Adding control points requires adding extra gates in the normal signal paths of the circuit, which may affect the performance. Hence, we shall constrain ourselves to addition of observation points because they are less intrusive. One option is to add latches for all observation points and connect them through a scan-out chain. This structure is often used for design debugging and can help in improving the fault coverage of tests [26]. Another option is to use an XOR tree to condense the logic values at the various observation points [9, 13, 16, 39]. Since the XOR tree requires less hardware, we use it to demonstrate an RTL DFT methodology in the next section.

6. Experimental Results

The gate-level netlist for Parwan processor was downloaded from an internet website [1]. As mentioned in Section 4, the bidirectional data-bus was separated into input and output buses and a global reset input was added. The test vectors for 737 RTL faults were obtained using the Mentor Graphics tool FlexTest [34], which is a sequential ATPG system with a built in fault simulator. Those 134 vectors were analyzed for their spectrum, new vector sequences were generated and compacted using the technique discussed in Section 3. These results were obtained on Sun Ultra 5 machines with 256MB RAM.

Table 1 shows the characteristics of RTL test vectors. Columns (left to right) give number of RTL faults, number of vectors generated, test generation CPU seconds, number of spectral coefficients derived, RTL fault coverage and gate-level fault coverage of RTL vectors. Faults in the clock network were not included and a 50% credit was given to potentially testable faults [38]. Coverages are defined as ratios of the number of detected faults to the total number of detectable faults as reported by FlexTest. The last column gives the fault coverage of the 134 RTL vectors for collapsed gate-level faults in the entire circuit, excluding clock faults.

Table 1. Spectral characterization of processor circuit by RTL vectors.

No. of RTL faults	No. of vectors	CPU time s	No. of spectral coeff.	RTL fault cov.	Gate fault cov.
737	134	640	128	96.30%	81.22%

As discussed in Section 5, the RTL test generation reveals bottlenecks in the testability of the circuit. To detect the faults in tri-state bus drivers we modeled the buses with memory using the high-impedance signal state. An analysis of the undetected RTL faults revealed 10 faults classified as unobservable by FlexTest. The remaining faults were either untestable or potentially tested. We selected the 10 unobservable fault sites as observation points and inserted a tree of nine XOR gates whose output was made into an added primary output. Thus, a DFT version of the processor was created. All RTL faults were now either detected or potentially detected by the same 134 vectors. However, the gate-level fault coverage showed only a moderate increase to 83.61%.

In Table 2 gives the results of the RTL spectral ATPG method. We achieve better test coverage as compared to the gate-level sequential ATPG in lower test generation time. The coverage is slightly lower than 100% because some faults were potentially testable and were given 50% credit. Since the RTL ATPG covered almost all faults in the original circuit, the benefit of DFT was small. However, the benefits of DFT were more for gate-level ATPG and random vectors. The test coverage plots of the original circuit and the circuit with DFT are shown in Figures 4 and 5.

To compare the effectiveness, we examined the gate-level faults that were left undetected by each method. For the original Parwan, the gate-level ATPG left 129 undetected (127 unobserved and 2 uncontrolled) faults. Besides, six faults were potentially detected. When these 135 faults were simulated with 2442 spectral ATPG vectors (Table 2), 106 unobserved, 1 uncontrolled and 1 potentially detected faults were detected. The other 5 potentially detected faults remained potentially detected. For the original Parwan, spectral ATPG had left 30 undetected (29 unobserved and 1 uncontrolled) faults. There were 5 potentially detected faults. These 35 faults were simulated with 1403 gate-level ATPG vectors shown in Table 2. Only 8 unob-

Table 2. Spectral RTL ATPG for processor circuits.

Circuit	Collapsed gate-level faults	RTL spectral ATPG			Gate-level ATPG			Random vectors	
		Cov. %	No. of vectors	CPU s	Cov. %	No. of vectors	CPU s	Cov. %	No. of vectors
Parwan (original)	2270	98.23	2327	2442	93.40	1403	26430	80.95	2814
Parwan (with DFT)	2320	98.77	1966	2442	95.78	1619	20408	87.09	2948

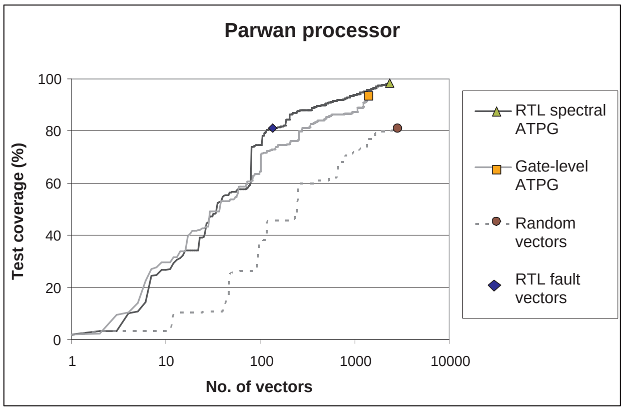

Figure 4. Test coverages for the original Parwan circuit [35].

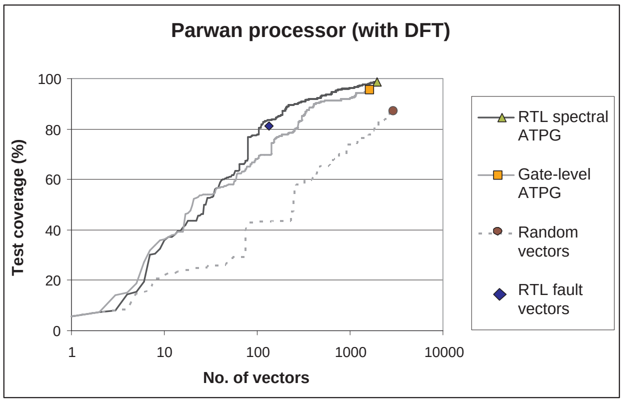

Figure 5. Test coverages for the Parwan circuit with DFT.

served faults were detected and all 5 potentially detected faults remained potentially detected.

For the DFT version of Parwan, gate-level ATPG left 82 undetected (78 unobserved and 4 uncontrolled) faults and there were 6 potentially detected faults. Of these, spectral ATPG vectors detected 62 unobserved and 3 uncontrolled faults. All 6 potentially detected faults remained the same way. Finally, in the reverse examination, 2442 spectral ATPG vectors had left 18 unobserved and 1 uncontrolled faults and had produced 6 potentially detected faults. The set of 1619 gate-level ATPG vectors only detected 2 of the unobserved faults. All 6 potentially detected faults remained the same way.

7. Conclusion

We have presented a new method of RTL test generation using spectral techniques for microprocessors. Test vectors for RTL faults are analyzed using Hadamard matrix to extract important features and new vectors are generated from them. We observe improved test coverage and lower test generation time as compared to a sequential gate-level ATPG tool. Since the method is based on RTL test generation, it has the advantages of lower memory and test generation time complexities. It enables the testability appraisal at RTL, and hence efforts can be made to improve testability when the design is conceptualized at higher levels of abstraction. An XOR observability tree designed at the RT level improved the gate-level fault coverage. Thus, RTL test methodology effectively reveals the bottlenecks in the testability of the circuit. Further, RTL ATPG enables the testing of cores for whom only the functional information is known.

References

[1] *http://aspire.ucsd.edu/∼lichen/260c/parwan/*.

[2] K. Batcher and C. Papachristou, "Instruction Randomization Self Test for Processor Cores," in *Proc. 17th IEEE VLSI Test Symp.*, Apr. 1999, pp. 34–40.

[3] K. G. Beauchamp, *Applications of Walsh and Related Functions with an Introduction to Sequency Theory*. Orlando, FL: Academic Press, 1984.

[4] D. Brahme and J. A. Abraham, "Functional Testing of Microprocessors," *IEEE Trans. on Computers*, vol. C-33, no. 6, pp. 475–485, June 1984.

[5] F. Brglez, C. Gloster, and G. Kedem, "Built-In Self-Test with Weighted Random-Pattern Hardware," in *Proc. IEEE Int. Conf. Computer Design*, Sept. 1990, pp. 161–167.

[6] L. Chen and S. Dey, "Software-Based Self-Testing Methodology for Processor Cores," *IEEE Trans. CAD*, vol. 20, no. 3, pp. 369–380, Mar. 2001.

[7] X. Chen and M. S. Hsiao, "Characteristic Faults and Spectral Information for Logic BIST," in *Proc. Int. Conf. Computer-Aided Design*, Nov. 2002, pp. 294–298.

[8] X. Chen and M. S. Hsiao, "Testing Embedded Sequential Cores in Parallel Using Spectrum-Based BIST," *IEEE Trans. Computers*, vol. 55, no. 2, pp. 150–162, Feb. 2006.

[9] R. P. Davidson *et al.*, "BELLMAC-32 A Self-Testable 32bit Microprocessor," in *Proc. International Test Conf.*, 1981, pp. 15–20.

[10] S. K. Devanathan and M. L. Bushnell, "Sequential Spectral ATPG Using the Wavelet Transform and

Compaction," in *Proc. 19th International Conference on VLSI Design*, 2006, pp. 407–412.

[11] R. Dorsch and H.-J. Wunderlich, "Accumulator Based Deterministic BIST," in *Proc. International Test Conf.*, Oct. 1998, pp. 412–421.

[12] P. Drineas and Y. Makris, "Independent Test Sequence Compaction through Integer Programming," in *Proc. Int. Conf. Computer Design*, 2003, pp. 380–386.

[13] E. B. Eichelberger and E. Lindbloom, "Random-Pattern Coverage Enhancements and Diagnosis for LSSD Logic Self-Test," *IBM J. Research and Development*, vol. 27, no. 3, pp. 265–272, May 1983.

[14] B. J. Falkowski, "Spectral Testing of Digital Circuits," *VLSI Design*, vol. 14, no. 1, pp. 83–105, 2002.

[15] P. F. Flores, H. C. Neto, and J. P. Marques-Silva, "An Exact Solution to the Minimum Size Test Pattern Problem," *ACM Trans. Design Automation of Electronic Systems*, vol. 6, no. 4, pp. 629–644, oct 2001.

[16] J. R. Fox, "Test-Point Condensation in the Diagnosis of Digital Circuits," *Proc. IEE*, vol. 124, no. 2, pp. 89–94, Feb. 1977.

[17] A. Giani, S. Sheng, M. S. Hsiao, and V. D. Agrawal, "Efficient Spectral Techniques for Sequential ATPG," in *Proc. Design, Autom. & Test in Europe (DATE) Conf.*, 2001, pp. 204–208.

[18] A. Giani, S. Sheng, M. S. Hsiao, and V. D. Agrawal, "Novel Spectral Methods for Built-In Self-Test in a System-on-a-Chip Environment," in *Proc. 19th IEEE VLSI Test Symp.*, 2001, pp. 163–168.

[19] S. Gurumurthy, S. Vasudevan, and J. A. Abraham, "Automated Mapping of Pre-Computed Module-Level Test Sequences to Processor Instructions," in *International Test Conf.*, Nov. 2005, pp. 294–303.

[20] D. S. Ha and S. M. Reddy, "On the Design of Random-Pattern Testable PLA Based on Weighted Random-Pattern Testing," *J. Electron. Testing: Theory and Applications*, vol. 3, no. 2, pp. 149–157, May 1992.

[21] J. P. Hayes and A. D. Friedman, "Test Point Placement to Simplify Fault Detection," in *Proc. Fault Tolerant Computing Symp.*, 1973, pp. 73–78.

[22] S. Hellebrand and H.-J. Wunderlich, "Mixed-Mode BIST using Embedded Processors," in *Proc. International Test Conf.*, Oct. 1996, pp. 195–204.

[23] T.-C. Hsiao and S. C. Seth, "An Analysis of the Use of Rademacher-Walsh Spectrum in Compact Testing," *IEEE Trans. Computers*, vol. 33, pp. 934–938, Oct. 1984.

[24] S. L. Hurst, D. M. Miller, and J. C. Muzio, *Spectral Techniques in Digital Logic*. Orlando, FL: Academic Press, 1985.

[25] V. S. Iyengar and D. Brand, "Synthesis of Pseudo-Random Pattern Testable Designs," in *Proc. International Test Conf.*, 1989, pp. 501–508.

[26] D. Josephson and B. Gottlieb, "Silicon Debug," in D. Gizopoulos, editor, *Advances in Electronic Testing: Challenges and Methodologies*, Springer, 2005. Chapter 3.

[27] K. R. Kantipudi and V. D. Agrawal, "On the Size and Generation of Minimal N-Detection Tests," in *Proc. 19th Int. Conf. VLSI Design*, 2006, pp. 425–430.

[28] G. Kasturirangan and M. S. Hsiao, "Spectrum-Based BIST in Complex SOCs," in *Proc. 20th IEEE VLSI Test Symp.*, Apr. 2002, pp. 111–116.

[29] O. I. Khan and M. L. Bushnell, "Aliasing Analysis of Spectral Statistical Response Compaction Techniques," in *Proc. 19th International Conference on VLSI Design*, 2006, pp. 801–806.

[30] N. Kranitis, D. Gizopoulos, A. Paschalis, and Y. Zorian, "Instruction-Based Self-Testing of Processor Cores," in *Proc. 20th IEEE VLSI Test Symp.*, Apr. 2002, pp. 223–258.

[31] L. Lingappan, S. Ravi, and N. K. Jha, "Test Generation for Non-Separable RTL Controller-Datapath Circuits using a Satisfiability Based Approach," in *Proc. 21st International Conference on Computer Design*, Oct. 2003, pp. 187–193.

[32] P. Maxwell, I. Hartanto, and L. Bentz, "Comparing Functional and Structural Test," in *Proc. International Test Conf.*, 2000, pp. 336–343.

[33] E. J. McCluskey, "Stuck-Fault Tests vs. Actual Defects," in *Proc. Int. Test Conf.*, 2000, pp. 336–343.

[34] Mentor Graphics, *FastScan and FlexTest Reference Manual*, 2004.

[35] Z. Navabi, *Analysis and Modeling of Digital Systems*. New York: McGraw-Hill, 1993.

[36] P. Nigh, W. Needham, K. Butler, P. Maxwell, and R. Aitken, "An Experimental Study Comparing the Relative Effectiveness of Functional, Scan, IDDq and Delay-Fault Testing," in *Proc. IEEE VLSI Test Symp.*, 1997, pp. 459–464.

[37] P. Parvathala, K. Maneparambil, and W. Lindsay, "FRITS - A microprocessor functional BIST method," in *Proc. Int. Test Conf.*, Oct. 2002, pp. 590–598.

[38] R. Raina, C. Njinda, and R. F. Molyneaux, "How Seriously Do You Take Possible-Detect Faults?," in *Proc. Intnternational Test Conf.*, Nov. 1997, pp. 819–828.

[39] E. M. Rudnick, V. Chickermane, and J. H. Patel, "Probe Point Insertion for At-Speed Test," in *Proc. IEEE VLSI Test Symp.*, 1992, pp. 223–228.

[40] A. K. Susskind, "Testing by verifying Walsh coefficients," *IEEE Trans. Computers*, vol. 32, no. 2, pp. 198–201, Feb. 1983.

[41] S. Thatte and J. A. Abraham, "Test Generation for Microprocessors," *IEEE Trans. Computers*, vol. C-29, no. 6, pp. 429–441, June 1980.

[42] A. R. Thompson, J. M. Moran, and G. W. Swenson Jr., *Interferometry and Synthesis in Radio Astronomy*. New York: Wiley Publisher, 1986.

[43] M. A. Thornton, R. Drechsler, and D. M. Miller, *Spectral Techniques in VLSI CAD*. Boston: Springer, 2001.

[44] A. van de Goor and T. J. Verhallen, "Functional Testing of Current Microprocessors," in *Proc. International Test Conference*, Sept. 1992, pp. 684–695.

[45] E. W. Weisstein. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com.

[46] S. Wolfram, *A New Kind of Science*. Champaign, IL: Wolfram Media, 2002.

[47] H.-J. Wunderlich, "Self Test using Unequiprobable Random Patterns," in *Proc. 17th Int. Symp. Fault-Tolerant Computing*, July 1997, pp. 258–263.

[48] N. Yogi and V. D. Agrawal, "High-Level Test Generation for Gate-Level Fault Coverage," in *15th IEEE North Atlantic Test Workshop*, May 2006.

[49] J. Zhang, M. L. Bushnell, and V. D. Agrawal, "On Random Pattern Generation with the Selfish Gene Algorithm for Testing Digital Sequential Circuits," in *Proc. Int. Test Conference*, 2004, pp. 617–626.