# Transition Delay Fault Testing of Microprocessors by Spectral Method

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 spectral method of delay test generation for microprocessors at the register-transfer level (RTL). Vectors are first generated by an available ATPG tool for transition faults on inputs and outputs of the RTL modules of the circuit. These vectors are analyzed using Hadamard matrices to obtain Walsh function components and random noise levels for each primary input. A large number of vector sequences is then generated such that all sequences have the same Walsh spectrum but they differ due to the random noise in them. At the gate-level, a fault simulator and an integer linear program (ILP) compact these vector sequences. The initial RTL vector generation also reveals 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 an accumulator-based processor named Parwan. The RTL technique produced higher gate-level transition fault coverage in shorter CPU time as compared to a gate-level transition fault ATPG.

## I. Introduction

With the current generation microprocessors becoming faster and more complex, new challenges are faced in their testing. The test generation problem has grown intricate and new requirements are been placed on the test generation methods like testing of delay faults. Hence functional at-speed tests are sometimes used in addition to the conventional scan tests. Several studies [21], [24] have shown the effectiveness of functional tests over structural scan tests in detecting chip defects.

Several research papers have been published on functional test generation of microprocessors [3], [29], [30]. However, functional testing achieves low fault coverage as it is not aimed at detecting faults modeled at the gate level. Recent work [13], [20] performs mapping of precomputed module level test sets to processor instructions using data structures like algebraic decision diagrams and bounded model checking to solve for instructions and circuit imposed constraints and hence requires more engineering effort. Papers have been published in the area of test vector generation using built-in self-test (BIST) for microprocessors and consists of pseudo-random tests, deterministic patterns, weighted random patterns [4], [33], and randomized instructions [2], [25] for BIST, but these techniques generally require very large number of test patterns. Some work has also been published on software BIST [5], [18] which involves microprocessor testing using instructions loaded in program memory and require a good knowledge of the instructions affecting each component and the analysis of controllability and observability of the component terminals in with respect to each instruction. Most work we mentioned does not address delay testing.

In this paper, we present an RTL spectral method for delay testing of processors. Unlike some of the methods discussed earlier, the ATPG effort required is low as our method is simulation-based. Also since we target delay faults at RTL, test generation complexity is low. Our method is based on spectral testing.

Susskind [28] 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 [10], [15]. More recently, Giani *et al.* [11], [12] have reported spectral techniques for sequential ATPG and built-in self-test. Hsiao *et al.* [6], [7] have published works on spectrum-based self test and core test. Zhang *et al.* [36] refined the method of extracting the spectra from a digital signal using a selfish gene algorithm. Yogi and Agrawal introduced a spectral RTL test generation method for stuck-at faults for sequential circuits [34] and for microprocessors [35]. The contribution of this paper is a method for deriving transition fault tests [19] used in delay testing of processor circuits. The ATPG method described in this paper is an extension of our previous work [35], which dealt with stuck-at faults in processor circuits.

The outline of the paper is as follows. Section II gives an overview of how bit-streams are analyzed in the spectral domain. In Section III, we present our RTL ATPG method for transition faults using spectral analysis and discuss its application to a processor circuit in Section IV. Section V describes a design-for-testability (DFT) method. Results

1-4244-1051-7/07/$25.00 ©2007 IEEE

are discussed in Section VI and finally we conclude in Section VII.

## II. 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 [31] to analyze the spectrum because they have been used for testing with effective results. Walsh functions are a set of orthogonal functions, which 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 $H(n)$ [31], when arranged in the so-called "sequency" order [32].

Hadamard matrices can be generated using the following recurrence relation:

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

where $H(0) = 1$ and $2^n$ is the dimension of the $n$th 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 and are called as coefficients. By analyzing these coefficients we will be able to determine the major contributing basis bit-streams in an original signal.

## III. Spectral RTL ATPG

Our approach to RTL test generation consists of two principal steps [35]:

1) RTL spectral characterization
2) Gate-level test generation

### A. RTL spectral characterization

The RTL faults considered are the transition delay 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.

The generated vectors are analyzed using Hadamard matrix to find the major spectral components. The bit-streams entering various inputs are analyzed separately. To find the spectral components for a bit-stream, it is multiplied with the Hadamard matrix. A high value of the coefficient corresponds to a high correlation of the bit-stream to the corresponding Walsh function and vice-versa. Hence, Walsh functions 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 Walsh functions and determining the essential component(s).

### B. Gate-level test generation

After spectral analysis of the RTL vectors, to generate test vectors for gate-level transition delay faults, the essential spectral coefficients are retained and others, being considered noise are perturbed. Test vectors are then generated from the coefficients by multiplying the coefficients with the Hadamard matrix again.

We generate $M$ such test sets by perturbing the spectra. We then 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 [8], [34].

## IV. Microprocessor Testing

We applied our RTL-spectral test method to a simple accumulator-based processor named Parwan [23]. Figure 2 shows a schematic of the PARWAN processor. It has an 8-bit data-bus and 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.

We sampled a total of 737 RTL transition faults, which were all the faults on the inputs-outputs of the different components (ALU, SHU, etc.), inputs-outputs of the registers (IR, PC, etc.) and the faults on the tri-state drivers. Vectors were generated to cover these RTL transition faults using a commercial sequential ATPG and new vectors were generated using the technique described in Section III.

## V. Design-for-Testability

Test generation for only the RTL faults has an added advantage of revealing the bottlenecks in the testability of

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

**Fig. 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, $H(3)$.**

the circuit. Analysis of hard to detect faults at the RTL gives an idea of the hard to test parts of the circuit and by increasing their testability, we could expect to increase the testability of the overall circuit. The testability of signals can be increased by increasing their controllability and/or observability [14], [16]. Adding control points requires adding extra hardware, which may affect the performance of the circuit. Being less intrusive we shall consider only observation points. One option is to add latches for all the 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 [17]. Another option is to use a XOR tree to condense the logic values at the various observation points [9], [27]. Since the XOR tree incurs less hardware, we used it as our DFT methodology.

## VI. Experimental Results

The gate-level netlist for Parwan processor was downloaded from an internet website [1] and modified as described in the previous section. The test vectors for RTL transition delay faults were obtained using the Mentor Graphics tool FlexTest [22], which is a sequential ATPG system with a built in fault simulator. Those RTL vectors were analyzed for their spectrum, new vector sequences were generated using the technique discussed above and finally they were compacted. Results were obtained on Sun Ultra 5 machines with 256MB RAM.

**TABLE I. Spectral characterization of processor circuit by RTL transition faults.**

| No. of faults | No. of vectors | CPU s | RTL transition (t-f) coverage | Gate-level t-f coverage |
|---|---|---|---|---|
| 737 | 160 | 3652 | 77.07% | 47.84% |

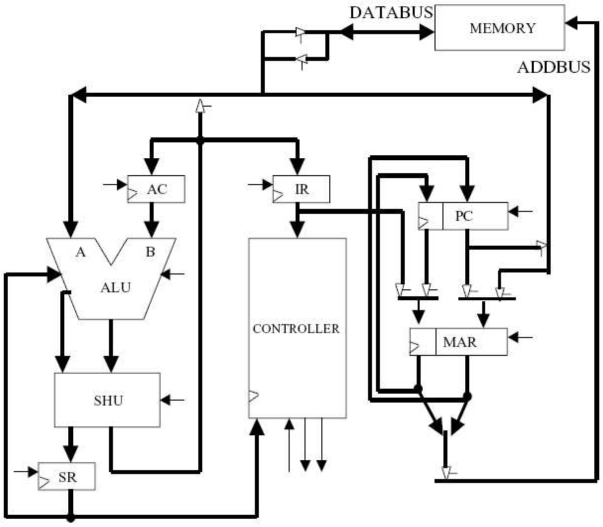

**Fig. 2. Parwan CPU [1].**

Table I shows the characteristics of RTL test vectors. Transition delay faults in the clock and reset network were not included and a 50% credit was given to potentially testable faults [26]. 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 160 RTL vectors for collapsed gate-level transition faults in the entire circuit, excluding clock and reset faults.

As discussed in Section IV, the RTL test generation reveals the bottlenecks in the testability of the circuit. Analysis of the RTL undetected transition delay faults for Parwan processor revealed around 24 unobservable faults. The remaining undetected transition faults were untestable or potentially testable. The circuit used tri-state drivers for driving buses and to improve the testability of their faults we modeled the high-impedance state. It was observed that almost all of the undetected RTL faults were on

**TABLE II. Comparing RTL-spectral and other ATPG strategies.**

| ATPG used | Version of Parwan circuit | CPU s for Sun Ultra 5 (256MB) | Number of vectors | Stuck-at fault coverage (%) | Transition fault coverage (%) |
|---|---|---|---|---|---|
| RTL-spectral for stuck-at faults [35] | Original | 2442 | 2327 | 98.23 | 79.47 |
| | DFT for s-a-f | 2442 | 1966 | 98.77 | 82.86 |
| RTL-spectral for transition faults | Original | 6428 | 6700 | 97.60 | 81.85 |
| | DFT for t-f | 6428 | 5120 | 98.25 | 85.94 |
| RTL-spectral combined stuck-at and transition tests | Original | | 9027 | 98.47 | 81.85 |
| | DFT for s-a-f | | 7086 | 98.91 | 85.87 |
| | DFT for t-f | | 7086 | 98.77 | 86.27 |
| Gate-level FlexTest for stuck-at faults | Original | 26430 | 1403 | 93.40 | 72.02 |
| | DFT for s-a-f | 20408 | 1619 | 95.78 | 80.49 |
| Gate-level FlexTest for transition faults | Original | 43574 | 1318 | 92.44 | 73.79 |
| | DFT for t-f | 40119 | 1444 | 96.29 | 81.90 |
| Random vectors | Original | | 51200 | 82.28 | 58.67 |
| | DFT for s-a-f | | 51200 | 86.20 | 65.82 |

the terminals of tri-state drivers. This could be because of the functional constraints of the processor and such faults might be functionally redundant delay faults. We selected the 24 unobservable fault sites as our observation points and condensed them using a 23 XOR gates tree and fed its output to an extra added output pin. All RTL transition delay faults were then either detected or potentially detected by the same 160 vectors and the RTL transition fault coverage of the RTL vectors increased to 58.40%.

The final results are tabulated in Table II, which compares the RTL-spectral ATPG with gate-level sequential ATPG (FlexTest) and random vectors. Results are given for three circuits; the original Parwan circuit, Parwan with DFT for stuck-at faults (s-a-f) [35], and Parwan with DFT for transition faults (t-f). CPU times for RTL-spectral ATPG are one order of magnitude lower than the gate-level FlexTest. Interestingly, the RTL-spectral ATPG vectors for transition faults, which numbered about 9,000, produced 2-3% higher coverage of transition faults and only a marginally lower stuck-at coverage. Best coverages are obtained when we combine RTL-spectral ATPG vectors for stuck-at faults [35] and those for transition faults. Both DFT strategies give similar result, i.e., about 99% stuck-at and 86% transition coverage. However, the number of vectors is larger. Coverages for a large set of random vectors are significantly lower.

Note that the highest achievable test coverage cannot be 100% as there was a small fraction of undetectable faults due to fault-induced uninitializability. These are possibly (or potentially) testable faults and were given a detection credit of 50%. We also observe an increase in the gate-level FlexTest coverages due to the DFT that was based on the RTL test points. Similarly, random vectors perform better on the circuit with DFT. The test coverages for the original circuit and the circuit with DFT by the combined RTL-spectral vectors are shown in Figures 3 and 4.

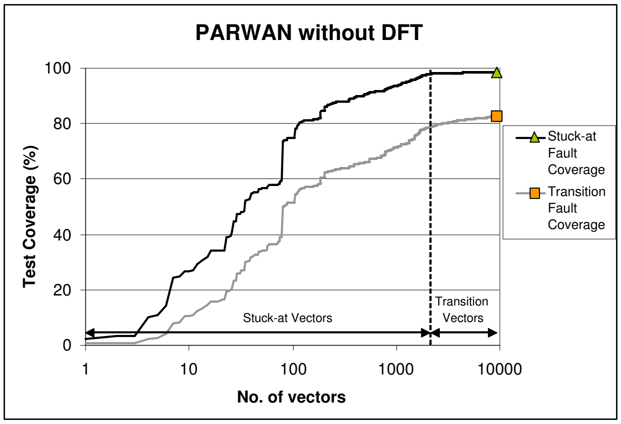

**Fig. 3. Test coverages of RTL-spectral vectors for original Parwan circuit [23].**

## VII. Conclusion

Building on the previous work [35] we have demonstrated a RTL-spectral ATPG for testing transition delay faults in microprocessors. Our method obtained an improvement in the test coverage with lower test generation times as compared to a gate-level sequential ATPG tool. The RTL test generation brings with it the advantages of lower memory usage, and reduced computation complexity. 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. Further, RTL ATPG enables the testing of cores for whom only the functional information may be known. Insertion of a XOR observability tree was found to improve both stuck-at and transition fault coverages. Observation test points, selected after appraising the RTL ATPG results, correctly revealed the testability bottlenecks of the circuit.

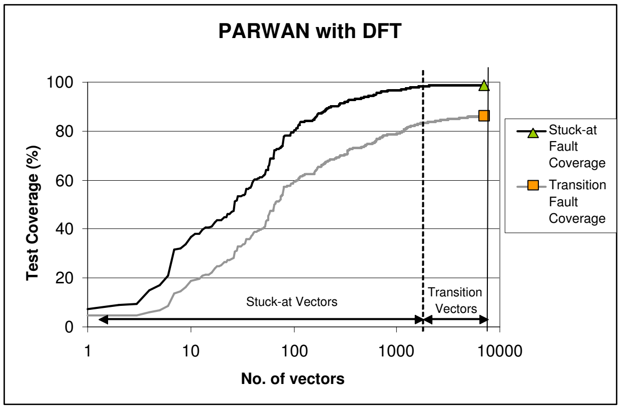

**Fig. 4. Test coverages of RTL-spectral vectors for Parwan with transition fault DFT.**

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] P. Drineas and Y. Makris, “Independent Test Sequence Compaction through Integer Programming,” in *Proc. International Conf. Computer Design*, 2003, pp. 380–386.

[9] 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.

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

[11] 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.

[12] 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.

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

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

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

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

[17] D. Josephson and B. Gottlieb, “Silicon Debug,” in D. Gizopoulos, editor, *Advances in Electronic Testing: Challenges and Methodologies*, Springer, 2005. Chap. 3.

[18] 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.

[19] A. Krstić and K.-T. Cheng, *Delay Fault Testing for VLSI Circuits*. Boston: Springer, 1998.

[20] 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 Int. Conf. Computer Design*, Oct. 2003, pp. 187–193.

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

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

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

[24] 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.

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

[26] 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.

[27] 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.

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

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

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

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

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

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

[34] N. Yogi and V. D. Agrawal, “Spectral RTL Test Generation for Gate-Level Stuck-at Faults,” in *Proc. 15th Asian Test Symposium*, Nov. 2006, pp. 83–88.

[35] N. Yogi and V. D. Agrawal, “Spectral RTL Test Generation for Microprocessors,” in *Proc. 20th International Conf. VLSI Design*, Jan. 2007, pp. 473–478.

[36] 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. International Test Conf.*, 2004, pp. 617–626.