Test Pattern Generator for Built-in Self-Test using Spectral Methods

Alok S. Doshi and Anand S. Mudlapur
Auburn University
200 Dept. of Electrical and Computer Engineering, Auburn, AL, USA
doshias,anand@auburn.edu

Abstract

A new method for test pattern generation (TPG) in a built-in self-test (BIST) environment is proposed here. The TPG uses the characteristic information of the circuit to generate the test vectors internally. The characteristic information of the circuit is extracted using known spectral methods. The algorithm was tested on different counter circuits and performs exceptionally well compared to the random test patterns by ATALANTA [15]. The hardware required for the TPG in the counter circuit is the same irrespective of the size of the counter. Thus the area overhead is minimal for greater length counter circuits.

1 Introduction

As the technology improves by the day, the logic-to-pin ratio on the chip increases proportionally [1]. This makes it difficult to analyze the signals on the device. Also the test application time associated with external testers reaches an upper bound in large circuits. At speed testing using an external ATE (Automatic Test Equipment) is extremely expensive and prohibitive amounts of test data must be stored in the ATE. BIST has been emerging as the single most effective solution to the above mentioned problems. BIST is an on-chip testing system that generates test vectors to detect faults and verifies whether the hardware is performing correctly. The main components of a BIST system are a TPG that applies a sequence of patterns to the circuit under test (CUT), a response compacter that compacts the responses into a signature, and a signature comparator that compares the signature to a fault free signature value.

Testing using pseudo-random test patterns often results in large test sets or insufficient fault coverages. Due to the low hardware costs BIST based on random patterns is very attractive. Linear Feedback Shift Registers (LFSRs) are commonly used in pseudo-random test pattern generators in BIST schemes. Pseudo-random sequences are useless for random-pattern-resistant faults [2]. A common solution to random-pattern-resistant faults is using weighted random patterns, which are found to yield better fault coverages [3]. While generating weighted random patterns the probability of obtaining a 0 or 1 at a particular input is biased toward detecting random-resistant faults. But this reasoning collapses since no one set of weight may be suitable for all faults. Deterministic BIST techniques such as stored pattern testing involve the application of specific test vectors, each providing an increase in fault coverage. However, a high cost is associated with the storing of the large number of patterns.

Walsh and Rademacher-Walsh function analysis have been proposed in the past [4, 5] only for response compaction. The use of these functions to generate vectors for sequential circuits and in an System on Chip (SOC) environment has been proposed in [6, 7]. Although the spectral analysis of the CUT is performed and new vectors based on the circuit information is generated, compaction needs to be performed at every iteration or requires an embedded controller or processing unit to generate test patterns. The predictability of a signal is the property that is made use of while generating new vectors. A signal can be reconstructed in its entirety if one knows the past and the current value of it.

Hadamard functions can be used to represent the spectral information in digital circuits. The Hadamard coefficients for input patterns that have high fault coverage is determined to predict new vectors. Hence extracting the spectral information from the CUT would help us in determining the natural frequencies of the circuit and generate customized vectors targeting specific faults.

The rest of the paper is organized as follows: In section 2 we present the overview and motivation to use Hadamard transforms to analyze the spectral information of the CUT. In section 3 we present the proposed TPG for BIST. In section 4 the results obtained for the counter circuits are presented. Outlook toward future work is described in section 5. We conclude the paper in section 6.

2 Overview and Motivation

We want to extract the information embedded in the input signals and output responses. Hence, we apply signal processing techniques to extract this information. In order to meet the above objective, we make use of frequency decomposition techniques, i.e. A signal can be projected to a set of independent, periodic waveforms that have different frequencies. This set of waveforms, forms the basis matrix. The projection operation reveals the quantity that each basis vector contributes to the original signal. This quantity is called the decomposition coefficient. With the aid of the decomposed information, one can easily enhance the important frequencies and suppress the unimportant ones. This process leads us to a new and better quality signal, easing the complexity (in our case i.e. of test generation).

The projection matrix chosen is the Hadamard transform as it is a well-known non-sinusoidal orthogonal transform used in signal processing. A Hadamard matrix consists of only 1s and -1s, which makes it a good choice for the signals in VLSI testing (1 = logic 1, -1 = logic 0). Each basis (row/column) in the Hadamard matrix is a distinct sequence that characterizes the switching frequency between 1s and -1s. The idea of extracting the spectral information is to generate new vectors spanning the likely vector space only using the basis vectors (Hadamard coefficients). This can potentially help drive the circuit into hard-to-reach states that requires specific inputs at the primary inputs, making it easier to detect hard-to-detect faults faster.

Hadamard matrices are square matrices containing only 1s and 1s. They can be generated recursively using the formula,

$$H(k) = \begin{bmatrix} H(k-1) & H(k-1) \\ H(k-1) & -H(k-1) \end{bmatrix}, \ k = 1, 2, ..., n,$$

where, H(0) = 1 and n = $\log_2$N. For k=1 and k=2 we get,

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

$$H(2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

The Walsh functions can be interpreted as binary (sampled) versions of the sine and cosine, which are the basic functions of the Discrete Fourier Transform. This interpretation led to the name BInary FOurier REpresentation (BIFORE) [8]. The inverse Hadamard transform is given by the transpose of the Hadamard matrix scaled by the factor 1/N. It must also be observed that Hadamard transform has no twiddle factors since its basis functions are square waves of either 1 or -1. Hence the Hadamard transform requires no multiplications and only N $\log_2$(N) additions. The inverse transform of Hadamard matrix is the same as its forward transform. Hence, reconstruction is very simple.

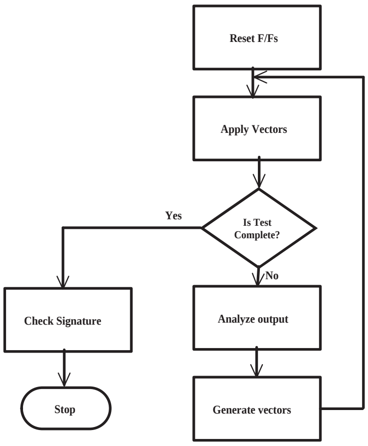

Figure 1: BIST Flow

Each row/column of the Hadamard matrix is a basis vector with a distinct frequency component. As an example H(2) has four basis vectors given by, [1 1 1 1], [1 0 1 0], [1 1 0 0] and [1 0 0 1]. Any bit sequence of length 4 can be represented as a linear combination of these basis vectors. For example, the vector [1 1 1 0] can be written as 1 × [1 1 1 1] + 1 × [1 -1 1 -1] + 1 × [1 1 -1 -1] − 1 × [1 -1 -1 1]. Hence, the test sequence is projected onto the Hadamard bases, certain frequencies are filtered out and an inverse transform is performed to get the de-noised sequence.

3 Proposed TPG for BIST

The flow chart in Figure1 represents the working of the BIST structure. The circuit shown in Figure2 is the TPG proposed for BIST for counters. It consists of a TPG, which is a simple lookup table with a multiplexer for selecting the cut-off value, and a series of FFs for feeding the test patterns into the counter circuit. The FFs are either set or reset initially (they are powered up in [0 0 0 1] state), and the only input to the counter circuit is reset, which resets the circuit when a logic '1' is applied to it. Hence when the external FFs are reset and ready for the BIST to start, the input applied at the reset pin is '0'. The circuit is clocked and responses are compacted in a response compacter. A feedback is drawn from the output of the circuit to compute new vectors. This is a crucial step in the TPGs functioning. The following paragraphs explains how the feedback is used to generate the vectors that quickly attain a very high fault coverage.

Algorithm:

Let a_i be the input bit sequence for primary input i.

```
/* coefficient extraction */

1. for(each primary input i in test set)
   coefficient vector $c_i = H \times a_i$

2. for(absolute value in the coefficient matrix [$c_0$,...,$c_n$])
   if(absolute value of coefficient < cutoff)
   Set the coefficient to 0.
   else
   Set the coefficient to 1 or -1, based on its absolute value.

/* generation of vectors based on coefficients */

1. for(each primary input i)
   extension vector $e_i$ = modified $c_i \times H$

2. if(weight > 0)
   Extend the vector set with value 1 to PIi.
   else if(weight < 0)
   Extend the vector set with value -1 to PIi.
   else if(weight==0)
   Extend the vector set with value 1 to PIi.
```

The above algorithm is exactly same as [7], except that the spectral analysis is performed after the application of every vector and the cut-off changes depending on the feedback. Also we choose the value 1 when the weight turns out to be a zero.

The illustration of the above algorithm and the working of the test pattern generator is described with the help of a four-bit counter circuit. On power up, clear is '0' which will reset the first 3 FFs and set the last FF of the TPG. So let us consider this stream of bits, i.e. [0 0 0 1] being fed to the TPG. Replacing '0' with -1 we get,

$$\begin{bmatrix} PI \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} PI \\ -1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

Next, we perform spectral analysis. Multiplying H(2) with the bit stream for PI, we obtain its coefficient vector:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \\ -2 \\ 2 \end{bmatrix}$$

Since clear is a 0 and MSB is unknown, cutoff chosen is 1 (see Figure3). For the later cycles, clear will be 1, so cutoff will be chosen solely depending on the MSB of counter. Applying the coefficient extraction part of the algorithm. Thus new c_i becomes [-1 -1 -1 1]^T. Now apply the second part of the algorithm;

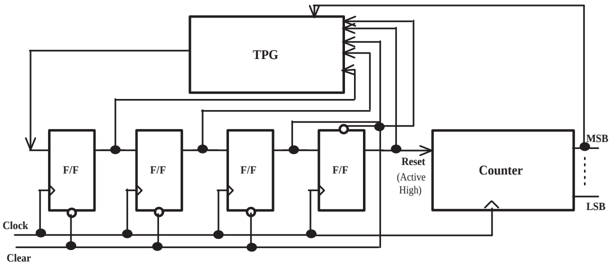

Figure 2: Proposed TPG.

[-1 -1 -1 1]× H(3) = [-2 -2 -2 2]

Hence the extended vector becomes [-1 -1 -1 1] which is [0 0 0 1]. All these matrix multiplications are not performed in hardware. Instead we have a look-up table from which final extended sequences can be taken depending on the original bit stream and the cut-off value. Hence the look-up table will consist of 32 values, 16 for cut-off value of 1 and 16 for cut-off value of 4. From the extended sequence we are only interested in the first value as this is the bit that is fed to the FFs as the last bit from the FFs is applied to the counter circuit. The look-up table size can be further reduced by just storing the first bit from the extended sequence as this is the only bit that we are interested in. So, the size of look-up table will just be 32 bits.

In the above example of 4-bit counter, at end of first cycle we get a sequence [0 0 0 1] of which 0 is fed to the FFs. For all cycles after this the input bit stream to the TPG will be [0 0 0 0] with cut-off value selected as 1. So the extended sequence will always be [0 0 0 0]. When the MSB becomes a 1, cut-off selected will be 4 and the extended sequence during that cycle will be [1 1 1 1]. From this the first bit which is a one will be fed to FFs. Three cycles later,this 1 will be applied to the counter circuit which will reset it again. A total of 8 cycles will be required for the MSB to become a 1. So, after reset 8 + 3 = 11 cycles are required to reset the counter again. The MSB was chosen because it is intuitive that it would take the longest time for a transition on this bit. Hence all the other states would have toggled by the time the MSB changes state. So this would result in a maximum fault coverage. So, the test vector length for the 4-bit counter will 11 + 1(Reset) = 12.

The idea of vector holding suggested in [16] is made use of as a constant 0 is held on the reset line till the MSB toggles.

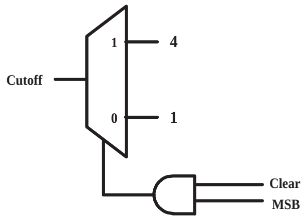

Figure 3: Cutoff Selection

Circuit	Random Patterns		Spectral BIST	
	# of vectors	fault coverage	# of vectors	fault coverage
mod10	224	43.73%	12	93.75%
mod16	224	88.46%	12	90.38%
mod32	224	72.06%	20	91.17%

Figure 4: Simulation against random patterns

4 Results

The results that we achieved in terms of vectors to detect all faults were comparable with that of the Gentest ATPG. In comparison to the random vector simulation performed using ATALANTA, we were able to achieve a higher fault coverage with an order of magnitude lesser number of vectors. Three counter circuits were simulated, which are modulo-10, modulo-16 and modulo-32. These circuits had 4, 5 and 6 potentially detectable Stuck-at-one faults on RESET line and its fanout stems respectively. The three circuit simulation results are as shown in Figure4:

5 An outlook toward future work

We have presented results for counter circuits in which we were able to justify our feedback connections from the most significant bit (MSB). The MSB was chosen because it is intuitive that it would take the longest time for a transition on this bit. Hence all the other states would have toggled by the time the MSB changes state. But this may not be the case with all the other circuits. Hence a procedure needs to be developed to identify those state elements in the circuit from which the feedback needs to be taken. One possible way of implementing the above scheme would be to execute an Euler walk of all states [12]. Here each state is visited exactly once, with all sates forming a closed path in the state transition graph of the state machine.

The circuit used for test pattern generation could be modified to include response compaction as well. Also the cut-offs can be chosen from the intermediate responses that are accumulated. The key in this technique is to choose the right FF, so that the vector generation favors maximum fault coverage. Larger circuits may be simulated and tested since this technique has a fixed area overhead. Last but not least, the idea of test generation based on output responses need not be confined to BIST schemes, but effective ATPG vectors may be generated and a correlation may be developed between the outputs and inputs of the CUT.

6 Conclusion

We have described a new way of implementing a TPG that generates vectors based on the spectral information of the circuit. The cut-off value is determined by the output of the circuit. Hence, the test patterns that are generated suit the circuit characteristic and provide high fault coverage with minimum number of vectors. The feedback must be chosen so as to drive the FF states of the circuit to all the possible states only once, i.e. the use of the Euler graph for an exhaustive self-test is made use of. The area overhead will increase as the number of inputs to the circuit increase, because each input will have a separate TPG circuit associated with it.

References

[1] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for digital, Memory and Mixed-Signal VLSI Circuits.* Kluwer Academic Publisher, 2000.

[2] V. D. Agrawal, "When to use random testing", *IEEE Trans. on Computers*, Vol. C-27, No. 11, pp. 1054-1055, Nov. 1978.

[3] M. F. Alshaibi and C. R. Kime, "Fixed-biased pseudo-random built in self test for random pattern resistant circuits," *Proc. Intl. Test Conf.*, 1994, pp. 929-938.

[4] A. K. Susskind, "Testing by verifying Walsh Coefficients," *IEEE Trans. Computers*, Vol. C-32, no. 2, Feb., 1983, pp. 198-201.

[5] T-C. Hsiao and S. C. Seth, "The use of Rademacher-Walsh spectrum in random compact testing," *IEEE Trans. Computers*, Vol. C-33, Oct., 1984, pp. 934-937.

[6] A. Giani, M. S. Hsiao, S. Sheng and V. D. Agrawal, "Efficient spectral techniques for sequential ATPG," in *Proc. of the Design Automation and Test in Europe Conf.*, 2001, pp. 204-208.

[7] A. Giani, M. S. Hsiao, S. Sheng and V. D. Agrawal, "Novel Spectral Techniques for Built-in Self-test in a System-on-a-Chip Enviornment", in *Proc. on the* 19^{th}*IEEE VLSI Test Symposium*, 2001, pp. 163-168.

[8] S. K. Mitra and J. F. Kaiser, *Handbook for Digital Signal Processing.* John Wiley and Sons, 1993.

[9] J. Zhang, M. L. Bushnell and V. D. Agrawal, "On Random Pattern Generation with the Selfish Gene Algorithm for Testing Sequential Circuits", in *Proc. of the International Test Conf.*, Oct. 2004, pp. 617-626.

[10] O. Khan and M. L. Bushnell, "Spectral Analysis for Statistical Response Compaction During Built-In Self-Testing", in *Proc. of the International Test Conf.*, Oct. 2004, pp. 67-76.

[11] S. H. Sangani, R. Saeks and S. R. Liberty, "A Spectral Theoretic Approach to Fault Analysis in Linear Sequential Circuits", *Journal of Franklin Institute*, Vol. 302, No. 3, September1976, pp. 239-258.

[12] V. D. Agrawal, R. D. Blanton and M. Damiani, "Synthesis of Self-Testing Finite State Machines from High-Level Specification", in *Proc. International Test Conf.*, pp. 747-766, 1996.

[13] V. D. Agrawal, C. R. Kime and K. K. Saluja, "A Tutorial on Built-in Self-Test, Part 1: Principles," *IEEE Design and Test of Computers*, vol. 10, pp. 73-82, March 1993.

[14] V. D. Agrawal, C. R. Kime and K. K. Saluja, "A Tutorial on Built-in Self-Test, Part 2: Applications," *IEEE Design and Test of Computers*, vol. 10, pp. 69-77, June 1993.

[15] H. K. Lee and D. S. Ha, "On the Generation of Test Patterns for Combinational Circuits," Technical Report No. 12_93, Dep't of Electrical Eng., Virginia Polytechnic Institute and State University.

[16] H. B. Min and W. A. Rogers, "A Test Methodology for Finite State Machines using Partial Scan Design," *Journal of Electronic Testing: Theory and Applications*, vol. 3, no. 2, pp. 127-137, May 1992.