High-Level Test Generation for Gate-Level Fault Coverage

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

In this paper we introduce a spectral method of register transfer level (RTL) test generation for sequential circuits. We define RTL faults as stuck-at faults on all primary inputs, primary outputs, and flip-flop terminals. Test vectors generated to cover the RTL faults are analyzed using Hadamard matrices. The analysis determines the amplitudes of prominent Walsh functions and the random noise level for each primary input. That information is then used to generate vectors for any gate-level implementation. At the gate-level, a fault simulator and an integer linear program (ILP) are used to compact the test sequence. We give results for three ITC'99 and four ISCAS'89 benchmark circuits. Each ITC'99 circuit was synthesized two ways, separately for area and delay optimization. The RTL spectral vectors performed equally well on both implementations. When compared to a gate-level ATPG, the coverages of RTL vectors were similar and in many cases RTL vectors produced equal or higher coverage in shorter CPU time.

1. Introduction

Conventionally, test vectors are generated at the gate-level, i.e., after synthesis has been performed. Though this methodology has the advantage of being able to generate reliable, high fault coverage test vectors due to its direct use of the stuck-at fault model, it suffers from several disadvantages. For large circuits, the large number of faults and the algorithm complexity make the gate-level test generation time consuming and expensive. Since gate-level test generation is performed at a later stage in the design process it is difficult to deal with testability issues, revealed during test generation, in an already verified design. Also, the gate-level ATPG cannot be used for cores or circuits for which only the functional information is available. This scenario is frequently encountered in commercial environments.

RTL (register transfer level) or synthesis independent test generation eliminates the disadvantages of gate-level test generation discussed above. Several RTL test generation methods have been proposed. Ravi and Jha [32], Ghosh and Fujita [12], Kim and Hayes [25] and Goloubeva *et al.* [15] use pre-computed test sets for RTL constructs like adders, multiplexers etc. and derive test vectors for the whole RTL circuit. Pre-computed test sets either make some assumptions about the synthesis of the design or use a superset of the actually required test vectors. All of them use some kind of data structure or metrics to derive the RTL test sets, which have implications of large memory and computation overheads. Ravi and Jha [32], and Kim and Hayes [25] use controllability and observability metrics, while Ghosh and Fujita [12] and Goloubeva *et al.* [15] use data structures like Decision Diagrams. Yi and Hayes [41], and Pomeranz and Reddy [31] have proposed a fault model which considers fault activation and propagation from the primary inputs to primary outputs. Test vectors can then be generated to cover these faults. However these fault models have been restricted to only combinational circuits. Thaker *et al.* [35] have shown that a set of stuck-at faults of variables in high-level synthetic operators and at the boundaries of RTL modules can be used as a statistical sample for the gate-level coverage analysis. In our technique also we can use such faults in the case of a hierarchical RTL circuit.

Recent work by Kang *et al.* [21] uses a set of stuck-at faults on primary inputs for high-level coverage analysis. Their *sensitization fault* coverage considers the detection of each primary input fault separately at every primary output. The coverage of sensitization faults is shown to correlate well with the stuck-at fault coverage in any gate-level implementation. Sensitization faults can be used in the spectral method described here. In the present

*Student speaker.

work, however, we have used stuck-at faults at primary inputs and flip-flops, which can be detected at *any* primary output. In terms of test generation effort, these faults are easier to cover than the sensitization faults.

The gate-level spectral methods of test generation are relevant to the present research. In 1983, Susskind [34] 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 [2, 9, 20, 37]. Hsiao and Seth [19] further expanded that work to compact testing. More recently, Giani *et al.* [13, 14] 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 [3, 4, 23]. Khan and Bushnell [24] have designed hardware signature analyzers using spectral components. Zhang *et al.* [43] 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 [7].

In this paper, we present a spectral method of generating test vectors for sequential circuits using only RTL faults. The RTL faults considered here are faults on the inputs and outputs of the circuit and inputs and outputs of the FFs (since they remain invariant through synthesis). 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 detect almost as many faults as any gate-level ATPG. Besides, the sequences can be compacted to about the same size as that produced by the gate-level ATPG. In the RTL method, therefore, the use of gate-level ATPG is eliminated and only a fault simulator is used.

In another recent work, Yogi and Agrawal [42] consider structure-free vectors generated entirely from input and output function of the circuit. These vectors that might be normally used for simulation-based verification of the circuit are analyzed for prominent spectral components and new vectors are generated for improved gate-level fault coverage.

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 results in Section 4. Finally we conclude in Section 5.

2. Background

Our method of test generation is based on the premise that the spectrum of vectors that detect high-level faults of the circuit reflects important characteristics of the circuit. These characteristics may include spatial and temporal correlations among the bits of primary input vectors and the necessary vector sequence length 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.

Frequency decomposition means that any bit-stream or signal can be projected on or represented using a set of orthogonal functions. The projections of the signal on each of the functions give the contribution of the corresponding functions to the original signal. We shall use Walsh functions [39] 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 $H(n)$ [39], when arranged in the so-called "sequency" order [36, 40]. The Hadamard matrix is a symmetric matrix with each row being a unique Walsh orthogonal function, also called as the basis function *bit-streams*. Since it consists of only +1s and -1s, it is a good choice for the signals in VLSI testing ($+1$ = logic 1, -1 = logic 0). Also multiplications can essentially be computed using additions and subtractions only.

Hadamard matrices are square matrices containing only $+1$ and -1 elements and can be generated using the following recurrence relation:

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

where $H(0) = 1$ and 2^n is the dimension of the nth order Hadamard matrix, $H(n)$. For example, for $n = 1$ and $n = 2$, we have:

$$H(1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad H(2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{2}$$

$$
\begin{array}{c}
1\\0\\1\\1\\1\\0\\1\\0
\end{array}
\xrightarrow{0 \to -1}
\begin{array}{r}
1\\-1\\1\\1\\1\\-1\\1\\-1
\end{array}
\quad \textit{Spectral decomposition}: \quad
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1\\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1\\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1\\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1\\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1\\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1
\end{bmatrix}
\begin{bmatrix}
1\\-1\\1\\1\\1\\-1\\1\\-1
\end{bmatrix}
=
\begin{bmatrix}
2\\\mathbf{6}\\-2\\2\\2\\-2\\-2\\2
\end{bmatrix}
$$

bit stream — modified bit stream — Hadamard matrix $H(3)$ — modified bit stream — spectral component magnitudes

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, $H(3)$.

An important property of the Hadamard matrix is that its transpose is equal to its inverse. Thus, $H(n) \times H(n)^T = nI_n$, where I_n is the $2^n \times 2^n$ identity matrix. This simplifies reconstruction of the test vectors from the spectral domain. Any bit-stream of k bits can be represented as a linear combination of the basis bit-streams from the Hadamard matrix, $H(\log_2 k)$. For example consider $k = 4$. $H(2)$ has four basis bit-streams, [1 1 1 1], [1 −1 1 −1], [1 1 −1 −1] and [1 −1 −1 1]. An object bit-stream [1 −1 −1 -1] can be expressed as $-1 \times$ [1 1 1 1] + $1 \times$ [1 −1 1 −1] + $1 \times$ [1 1 −1 −1] + $1 \times$ [1 −1 −1 1], 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, which we shall regard as important basis bit-stream functions.

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 analysis

3.1 Test Generation for RTL Faults

The RTL faults considered are the stuck-at faults on primary inputs and outputs of the circuit and on inputs and outputs of all flip-flops. These faults remain invariant through logic synthesis.

3.2 Spectral Analysis

We obtain test vectors to detect RTL faults. These vectors are analyzed using Hadamard matrix to find the major spectral components. To analyze a vector sequence, the bit-streams entering various inputs are analyzed separately. The 0s and 1s in a bit-stream are represented as −1s and +1s. To find the coefficients for the bit-stream corresponding to an input, the bit-stream is multiplied with the Hadamard matrix. The multiplication operation is basically 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). In the example, an 8 bit-stream (0s and 1s in the original sequence being represented as −1s and +1s) is analyzed by multiplying by a third order 8×8 Hadamard matrix. The corresponding result gives the coefficients. 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.

The RTL test vector sequences and are not al-

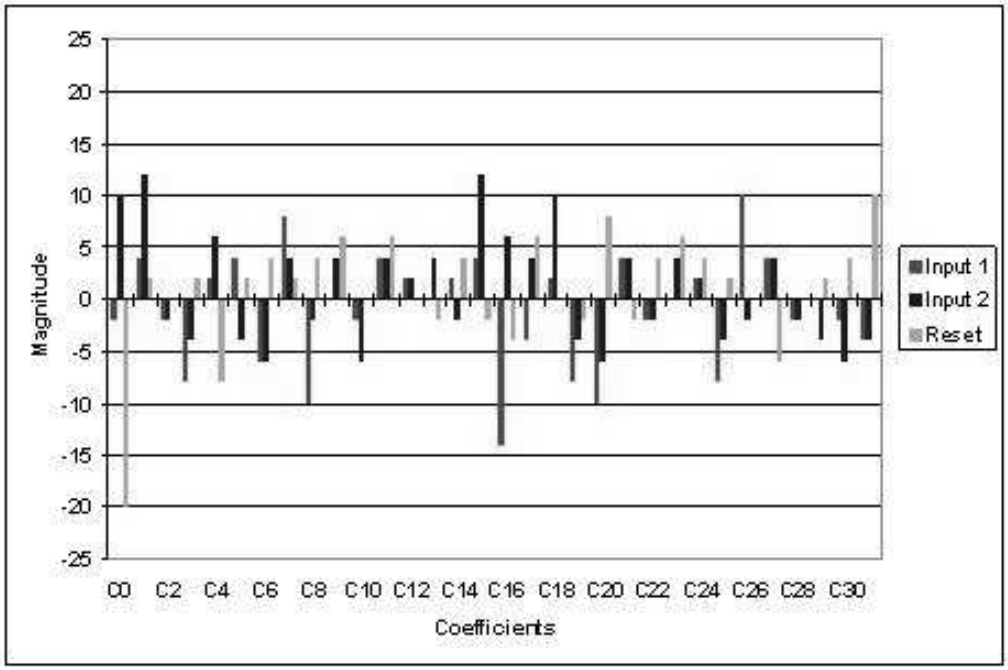

Figure 2. Walsh spectral coefficients for b01 circuit.

ways of lengths that are powers of 2. We have several options of analyzing the bit-streams. One option is to use a limited dimension Hadamard matrix and analyze pieces of the bit-stream individually. However this method has the disadvantage that, it might not be able to capture the long periodicity inherent in the bit-stream. We can analyze the whole bit-stream together. We either need to truncate the bit-stream or use a higher order Hadamard matrix. For practical reasons, we use a dimension which is closest to the length of the bit-stream. For example, for a bit-stream of length 135 we choose a Hadamard matrix of dimension 128 and analyze only the first 128 bits of the bit-stream. Analysis using a dimension of 256 is found to give rise to a large number of noisy components due to the uncertainty in the unspecified bits from 135 to 256 and hence presents difficulty in finding out the essential components. In cases where we choose a dimension greater than the bit-stream, the bit-stream is extended with 0s up to the dimension of the matrix to indicate non-inclusion of these bits in the correlation operation.

Figure 2 shows the spectral coefficients for the circuit b01 whose bit-streams are of length 38 but were analyzed using a 32×32 Hadamard matrix after truncating the original bit-stream. The high rising bars show high correlation with the corresponding basis bit-streams and these are considered essential.

To determine the threshold, which separates the essential components from the noise components, a technique in which the coefficient value is compared with the mean of the total spectrum is used. For *white noise* we shall have all equal valued coefficients and their magnitudes will be equal to the mean of any other arbitrary spectrum. Hence after all the coefficients are determined, they are squared and the magnitude of the maximum coefficient is compared with the mean of all coefficients by taking a ratio of the two. By performing this comparison we are actually comparing the percentage of power in the coefficient to the mean noise power. If the ratio of the magnitude of the coefficient being considered, to the mean of the coefficients is greater than some constant K, then the coefficient and hence the corresponding basis bit-steam is considered to be an essential component. The corresponding coefficient is removed from the spectrum and the same process is repeated for the next maximum coefficient and others correspondingly until the criteria are no longer satisfiable. The coefficients thus remaining after this process are considered non-essential or noise. The constant K affects which coefficients are being considered as essential ones. A very high value of K makes the selection process very acute, selecting only few components as essential ones while a low value of K may liberally select some non-essential components. We shall discuss in the next section

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

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

$$Sign\{[1\ \ \mathbf{6}\ \ 2\ \ -1\ \ 3\ \ -2\ \ 3\ \ -1] \times H(3)\} = [1\ \ 1\ \ 1\ \ -1\ \ 1\ \ -1\ \ 1\ \ -1] \rightarrow [1\ 1\ 1\ 0\ 1\ 0\ 1\ 0]$$

Figure 3. Bit-steam generation by perturbing the spectra. Note that the essential component having a magnitude 6 is not perturbed.

how we use this value of K to obtain efficient test vectors.

3.3 Spectral Vector Generation

After spectral analysis of the RTL vectors, the spectral coefficients are obtained. To generate test vectors for gate-level faults, the essential spectral coefficients decided by the threshold are retained and others, being considered noise, can be filtered out or changed as per the selected methodology. Filtering out the noise components [13] may have the disadvantage of losing some phase information inherent in those components, which is a part of the characterization of the circuit. Hence in our approach the noise components 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. These may depend on the circuit as well as the type of test vectors being used. We are working on refining these confidence levels and the nature of the spectral coefficients obtained from the RTL vectors, by studying their statistical properties. Currently, to find these levels we investigate different ranges by generating new coefficients in that range and analyze the fault coverages of the corresponding test vectors. The threshold is kept the minimum during this process to closely track the changes. A trend is generally visible and the optimum range for the confidence levels is then selected.

The test vectors can easily be generated from the coefficients by multiplying the coefficients with the Hadamard matrix again. Figure 3 shows an example of reconstruction of test vectors. As discussed before, the value of the constant K, to select the essential components, has an impact on the nature of the generated test vectors. A low value of K tends to preserve most of the nature of the original RTL vectors and has been found to give good initial fault coverage. A high value of K tends to select very few components as essential, regarding most as noise. This can sometimes generate vectors that may detect hard to detect faults. Hence we generate test vector sets with different values of K. Also since we are adding noise randomly, this variation gives different characteristics to each vector set. We generate multiple sets of vectors for each value of K.

3.4 Sequence Compaction and Coverage

Suppose the number of spectral components obtained for a circuit is n. This number approximately equals the number of vectors obtained for RTL faults. To be exact, this may equal the power of 2 that is closest to the number of RTL vectors. We generate perturbation vectors sequences, $V_1,\ V_2,\ \cdots,\ V_M$, as described in the previous subsection, each of length n, 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 [8, 10, 17, 22, 26]. During fault simulation, at the beginning of each vector sequence the complete fault list is restored and the circuit is set to an unknown state. Thus, the coverage obtained for a sequence remains valid irrespective to the order in which it is applied. Besides, the fault simulator provides a complete list of vector sequences that detect each fault. The vector sequence V_i is assigned an integer variable, $x_i \epsilon [0, 1]$, meaning that if $x_i = 1$, we will select the ith sequence, otherwise we will discard it. Now suppose kth fault is detected

by sequences V_3, V_4, and V_{11}. Then the following ILP constraint insures that we select at least one sequence that detects this fault:

$$x_3 + x_4 + x_{11} \geq 1 \quad (3)$$

The number of such constraints equals the number of faults. The ILP then determines the values of variables x_i's that satisfy all constraints of the type 3 with the following objective function:

$$\text{Minimize} \sum_{i=1}^{i=M} x_i \quad (4)$$

where M is the total number of vector sequences generated. For the results given in the next section, we used the ILP software contained in the AMPL mathematical programming package [11]. In the ILP solution, the smallest possible number of x's is assigned the value 1 and all others are assigned 0. If $x_i = 0$ then the sequence V_i is discarded. The remaining sequences form the compacted test set.

4. Results

The spectral technique of RTL-ATPG was applied to three ITC'99 RTL benchmark circuits b01, b09 and b11 and four ISCAS'89 benchmark circuits. The ITC'99 RTL benchmark circuits were synthesized in two ways, by optimizing area and by optimizing delay. The characteristics of these different circuits are shown in Table 1.

The test vectors for RTL faults were obtained using the Mentor Graphics tool FlexTest [27] 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 2 shows the characteristics of the RTL test vectors generated for the circuits. Column 1 lists the circuit name. Here b01-A and b01-D are the area and delay optimized implementations of the b01 ITC'99 benchmark. ISCAS'89 benchmarks are already at the gate-level. For s5378 and s9234, we created additional versions by adding a global reset input in the original circuits. These are denoted with an asterisk (*) in Tables 2 and 3.

Column 3 of Table 2 lists the number of RTL faults, which are the faults at the primary inputs, primary outputs, and the inputs and outputs of flip-flops. For example, consider the circuit b11, which

Table 1. Circuit description.

Circuit	PIs	POs	FFs
b01	2	2	5
b09	1	1	28
b11	7	6	31
s1488	8	19	6
s5378	36	49	179
s9234	37	39	211
s35932	36	320	1728

according to Table 1 has 7 primary inputs, 6 primary outputs and 31 flip-flops. These do not include the clock and reset inputs. Including those the primary input count is increased to 9. Each flip-flop has five terminals, three inputs (data, clock and reset) and two outputs (Q and $\overline{Q}$). Therefore, the RTL fault set consists of $2\times(9+6+31\times5) = 340$ stuck-at faults as shown in column 3 of Table 2.

Next in Table 2 appear the number of RTL test vectors, test generation time (CPU s) and the number of spectral components. In the absence of a true RTL ATPG program, we used FlexTest [27] to derive tests just for the selected stuck-at faults we designate as RTL faults. For the ITC'99 benchmarks this was done for two gate-level versions, one synthesized with area optimization and the other with delay optimization. The number of spectral components in the sixth column is the number of RTL vectors rounded off to the nearest power of 2.

The last two columns of Table 2 show the fault coverages of the RTL vectors. RTL Coverage is the coverage of just the RTL faults and gate-level coverage is the coverage of all stuck-at faults in the implementation. As expected, the gate-level coverage is lower than the normal requirement of being close to 100%. We will use our spectral technique to enhance this coverage. The low coverage of the RTL faults, however, may indicate a testability problem, which could limit our ability to increase the coverage either by the spectral technique or by gate-level ATPG.

Table 3 gives a comparison of the proposed RTL ATPG method with gate-level sequential ATPG. The first two columns give the circuit name and the number of gate level single stuck-at faults. The performances of RTL ATPG–spectral tests, gate-level ATPG, and random vectors can be compared by examining the data in the subsequent columns.

For RTL-ATPG, the number of vectors is the total number vectors in the compacted test sequences. Consider the circuit b01-A. In Table 2, there are 38 RTL vectors that provide 64, i.e., 38 rounded to next power of 2, spectral components. Ten se-

Table 2. Spectral characterization of circuits by RTL vectors.

Circuit		RTL Characterization					
Name	Gate-level synthesis	No. of faults	No. of vectors	CPU s	No. of spectral Components	RTL Cov. (%)	Gate-level Cov. (%)
b01-A	Area optimized	62	38	< 1	64	94.57	96.33
b01-D	Delay optimized	62	31	< 1	32	94.57	85.45
b09-A	Area optimized	248	109	519	128	75.22	78.18
b09-D	Delay optimized	248	193	418	256	75.22	72.69
b11-A	Area optimized	340	224	530	256	76.16	74.09
b11-D	Delay optimized	340	174	767	256	76.32	84.14
s1488	ISCAS'89	104	38	1	64	83.12	64.34
s5378	ISCAS'89	1602	115	1185	128	55.36	68.82
s5378*	ISCAS'89+Reset	1962	82	444	64	69.22	65.04
s9234	ISCAS'89	1840	16	706	16	18.48	16.45
s9234*	ISCAS'89+Reset	2264	59	2495	64	49.85	43.58
s35932	ISCAS'89	14536	92	50	64	68.80	94.03

quences, each of 64 vectors, were generated by the perturbation technique of Subsection 3.3. The fault simulator of FlexTest [27] provided a gate-level test coverage of 99.57% for the 228 stuck-at faults shown in column 2 of Table 3. Note that the test coverage of FlexTest is an upward adjusted coverage, accounting for faults that are found to be untestable. The ILP compaction technique of Subsection 3.4 selected two sequences reducing the test length to 128 vectors. The ATPG time in column 5 includes the times for RTL characterization (from Table 2), perturbed spectral sequence generation, fault simulation, and ILP compaction. Of these, RTL characterization and fault simulation are the dominant components, the other two being negligible.

Next, we find that FlexTest generates 75 vectors for a 99.77% test coverage and 640 random vectors (same number as in ten spectral sequences) have 97.78% test coverage. As we move down in Table 3, circuits become larger and we observe that RTL ATPG provides about the same test coverage and vector lengths as the gate-level ATPG, but its time increases slower. Moreover, the RTL faults used for circuit characterization and vector generation are implementation independent. Noticeably, the test coverage of random vectors tends to drop as circuits become larger.

Figures 4 and 5 compare the graphs of test coverage percentage against the number of vectors for two circuits, b11-A and s35932, for RTL spectral ATPG vectors, gate-level ATPG vectors and random vectors. The gate-level coverages of RTL vectors (generated to cover RTL faults only) are also shown by a point in each graph. For b11-A, the coverage of RTL ATPG is about 4% higher, vector length about double and CPU time about half when compared to the gate-level ATPG. For s35732, RTL ATPG obtained only a fractionally lower coverage with a much reduced test length and used about a third of CPU time.

5. Conclusion

We have presented a new method of RTL test generation using spectral techniques. Test vectors generated for RTL faults are analyzed using Hadamard matrix to extract important features and new vectors are generated retaining those features. Generation of different types of test sets and performing compaction on them has been found to be an efficient and reliable method for test generation. Results show that as circuits become larger the RTL method may have advantages over gate-level ATPG. This reveals a promise in generation of test vectors at RTL by spectral analysis.

Performing RTL test generation brings with it the advantages of lower memory, computation and generation time complexity. It enables the testability appraisal at RTL, and hence efforts can be made to improve testability when the design is conceptualized at the higher levels of abstraction itself. It also enables the testing of circuits for whom only the functional information is known.

Availability of a RTL ATPG will relieve the reliance of our method on the requirement of a synthesized circuit for test generation. Several high-level / RTL fault models and test generation systems have been proposed by Hayne and Johnson [16], Riesgo and Uceda [33], Armstrong *et al.* [1, 5, 28, 38]. These can be effectively used for a RTL ATPG.

Table 3. Comparison of RTL ATPG and Sequential gate-level ATPG results.

Circuit name	No. of gate-level faults	RTL ATPG – spectral tests			Gate-level ATPG			Random inputs	
		Cov. %	No. of vectors	Compaction CPU s	Cov. %	No. of vectors	ATPG CPU s	No. of vectors	Cov. %
b01-A	228	99.57	128	19	99.77	75	1	640	97.78
b01-D	290	98.77	128	19	99.77	91	1	640	95.80
b09-A	882	84.68	640	730	84.56	436	384	3840	11.71
b09-D	1048	84.21	768	815	78.82	555	575	7680	6.09
b11-A	2380	88.84	768	737	84.62	468	1866	3840	45.29
b11-D	3070	89.25	1024	987	86.16	365	3076	3840	41.42
s1488	4184	95.65	512	103	98.42	470	131	1600	67.47
s5378	15584	76.49	2432	2088	76.79	835	4439	3840	67.10
s5378*	15944	73.59	1399	718	73.31	332	22567	2880	62.77
s9234	28976	17.36	64	721	20.14	6967	18241	160	15.44
s9234*	29400	49.47	832	2734	48.74	12365	4119	2176	33.06
s35932	103204	95.70	256	1801	95.99	744	3192	320	50.70

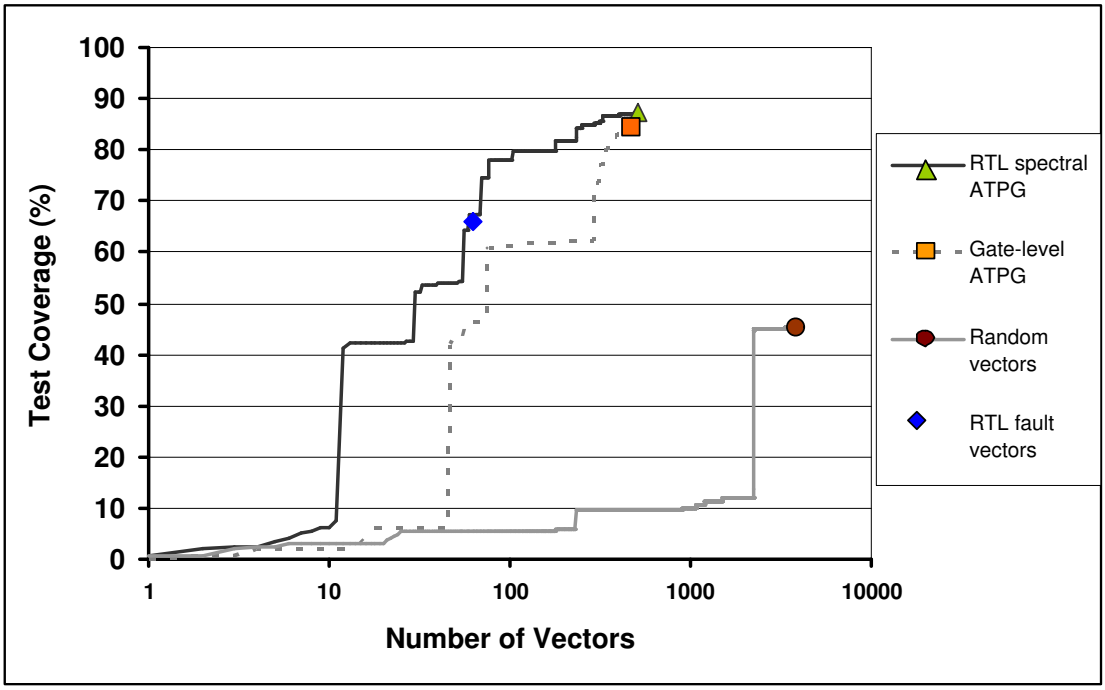

Figure 4. Test coverage of RTL ATPG (spectral vectors) for area optimized b11-A circuit.

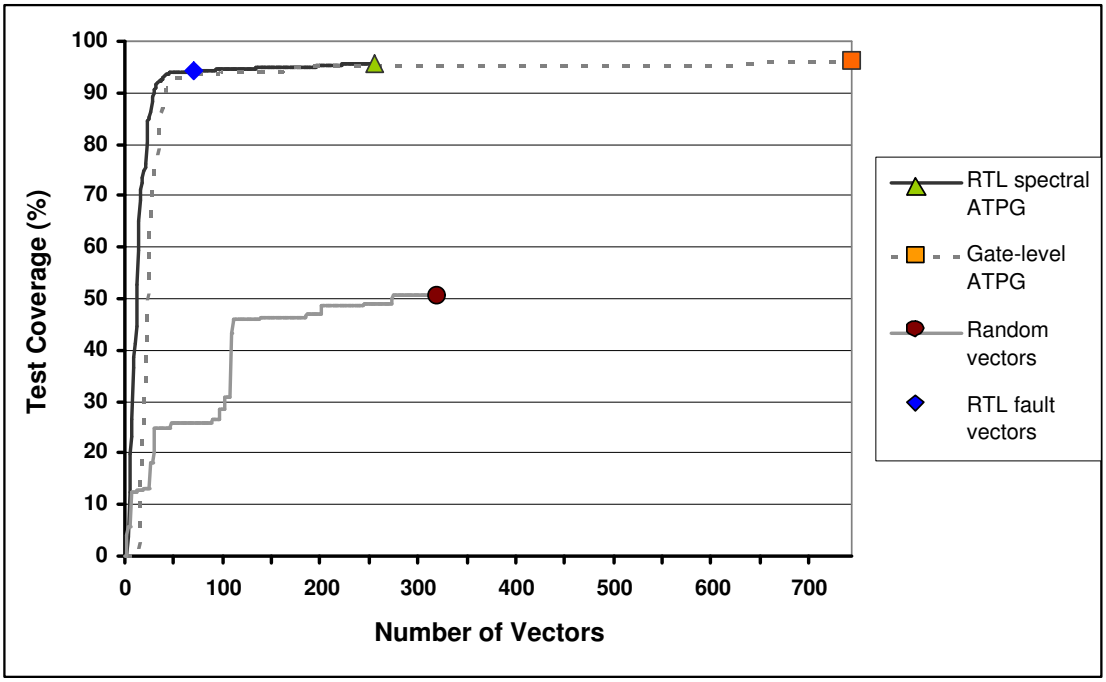

Figure 5. Test coverage of RTL ATPG (spectral vectors) for s35932 circuit.

Also means need to be found to circumvent the need of a synthesized circuit for test compaction. The need for the improvement in the current method is displayed in cases like the circuit b09. The current method derives it efficiency from the quality of RTL test vectors, and hence more effort in detection of RTL faults is bound to improve the RTL-ATPG results. Methods like N-detection or use of input-output fault pairs for RTL faults can bring about an improvement in the results. The computation time may be effectively reduced by a better compaction method [6, 18, 29, 30]. Another application of the spectral analysis to convert the functional verification sequences into compact gate-level fault coverage vectors is being investigated [42].

References

[1] D. S. Barclay and J. R. Armstrong, "A Heuristic Chip-level Test Generation Algorithm," in *Proc. 23rd Design Automation Conf.*, 1986, pp. 257–262.

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

[3] 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.

[4] 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.

[5] C. H. Cho and J. R. Armstrong, "B-algorithm: A Test Generation Algorithm," in *Proc. International Test Conf.*, 1994, pp. 968–979.

[6] F. Corno, P. Prinetto, M. Rebaudengo, and M. Sonza Reorda, "New Static Compaction Techniques of Test Sequences for Sequential Circuits," in *Europ. Design & Test Conf.*, Mar. 1997, pp. 37–43.

[7] 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.

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

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

[10] 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.

[11] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming.* South San Francisco, California: The Scientific Press, 1993.

[12] I. Ghosh and M. Fujita, "Automatic Test Pattern Generation for Functional RTL Circuits Using Assignment Decision Diagrams," in *Proc. 35th Design Automation Conf.*, 1999, pp. 43–48.

[13] 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.

[14] 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.

[15] O. Goloubeva, G. Jervan, Z. Peng, M. Sonza Reorda, and M. Violante, "High-level and Hierarchical Test Sequence Generation," in *Proc. of HLDVT*, 2002, pp. 169–174.

[16] R. J. Hayne and B. W. Johnson, "Behavioral Fault Modeling in a VHDL Synthesis Environment," in *Proc. 17th VLSI Test Symposium*, April 1999, pp. 333–340.

[17] D. S. Hochbaum, "An Optimal Test Compression Procedure for Combinational Circuits," *IEEE Trans. Computer-Aided Design*, vol. 15, no. 10, pp. 1294–1299, oct 1996.

[18] M. S. Hsiao, E. M. Rudnick, and J. H. Patel, "Fast algorithms for static compaction of sequential circuit test vectors," in *Proc. 15th IEEE VLSI Test Symp.*, Apr. 1997, pp. 188–195.

[19] 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.

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

[21] J. Kang, S. C. Seth, and V. Gangaram, "A Functional Fault Model for Coverage Analysis and Test Generation," 2006. Manuscript in preparation.

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

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

[24] 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.

[25] H. Kim and J. P. Hayes, "High-Coverage ATPG for Datapath Circuits with Unimplemented Blocks," in *Proc. International Test Conf.*, 1998, pp. 577–586.

[26] J. P. Marques-Silva, "Integer Programming Models for Optimization Problems in Test Generation," in *Proc. IEEE Asia-South Pacific Design Automation Conf.*, 1998, pp. 481–487.

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

[28] M. D. O'Neill, D. D. Jani, C. H. Cho, and J. R. Armstrong, "BTG: A Behavioral Test Generator," in *Proc. 9th Int. Symp. on CHDLs and their Applications*, 1989, pp. 347–361.

[29] I. Pomeranz and S. M. Reddy, "Dynamic test compaction for synchronous sequential circuits using static compaction techniques," in *Proc. Symposium on Fault Tolerant Computing*, June 1996, pp. 53–61.

[30] I. Pomeranz and S. M. Reddy, "On Static Compaction of Test Sequences for Synchronous Sequential Circuits," in *Proc. 33rd Design Automation Conf.*, 1996, pp. 215–220.

[31] I. Pomeranz and S. M. Reddy, "Pattern Sensitivity: A Property to Guide Test Generation for Combinational Circuits," in *Proc. 8th Asian Test Symposium*, 1999, pp. 75–80.

[32] S. Ravi and N. K. Jha, "Fast Test Generation for Circuits with RTL and Gate-Level Views," in *Proc. International Test Conf.*, 2001, pp. 1068–1077.

[33] T. Riesgo and J. Uceda, "A Fault Model for VHDL Descriptions at the Register Transfer Level," in *Proc. EURO-DAC/EURO-VHDL*, 1996, pp. 462–467.

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

[35] P. A. Thaker, V. D. Agrawal, and M. E. Zaghloul, "A Test Evaluation Technique for VLSI Circuits Using Register-Transfer Level Fault Modeling," *IEEE Trans. CAD*, vol. 22, no. 8, pp. 1104–1113, Aug. 2003.

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

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

[38] P. C. Ward and J. R. Armstrong, "Behavioral Fault Simulation in VHDL," in *Proc. 27th. ACM/IEEE Design Automation Conf.*, 1990, pp. 587–593.

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

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

[41] J. Yi and J. P. Hayes, "A Fault Model for Function and Delay Faults," *Journal of Electronic Testing: Theory and Applications*, vol. 21, no. 6, pp. 631–649, 2005.

[42] N. Yogi and V. D. Agrawal, "Spectral Characterization of Functional Vectors for Gate-Level Fault Coverage Tests," 2006. Manuscript in preparation.

[43] 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 Conference*, 2004, pp. 617–626.