At-Speed On-Chip Diagnosis of Board-Level Interconnect Faults

Artur Jutman
Tallinn University of Technology
artur@pld.ttu.ee

Abstract

This article describes a novel approach to fault diagnosis suitable for at-speed testing of board-level interconnect faults. The approach is based on a new parallel test pattern generator and a specific fault detecting sequence. The test sequence has tree major advantages. At first, it detects both static and dynamic faults upon interconnects. Secondly, it allows precise on-chip at-speed fault diagnosis of interconnect faults. Third, the hardware implementation of both the test generator and the response analyzer is very efficient in terms of silicon area.

1. Introduction

The problem of board level interconnect test has been thoroughly studied in the past. Most methodologies proposed during the last decade tend to rely on the Boundary Scan (BS) technique [1] as a powerful and universal test access mechanism. The related problem – the problem of Interconnect BIST (IBIST) is usually approached in the same way. The major difference between known methods of IBIST is usually defined by the manner they treat several important issues like:

- code used for test generation (e.g. Walking Sequence, Counting Sequence, True/Complement Code etc.)
- hardware implementation of both the test pattern generator (TPG) and the response analyzer (RA)
- covered defect types (fault models) - as a result of selected code and hardware implementation
- handling of the multiple driver contention problem

There are also some other important issues like the power consumption level or the ground bounce problem [2].

The widely accepted interconnect fault model usually handles static defects like opens and shorts. The literature, however, agrees on the fact that the dynamic defects like transition (delay) faults, crosstalk, or switching noise are becoming very important due to higher clock speeds and lower supply voltages [3]. Unfortunately, these latter effects do not show up in practice at the testing frequencies or the frequencies the Boundary Scan operates on. The nature of this drawback originates from the fact that the Boundary Scan architecture was not designed for at-speed testing and diagnosis of timing-related faults. For instance, the interval between the update of test stimulus and capture of the response data spans at least 2,5 test clock cycles. Another drawback, which also results from the Boundary Scan nature, is the need for a long shift operation between two consecutive test vectors. This shift operation depends on the BS chain length and may take hundreds of clock cycles to complete. These two facts together limit or totally prevent efficient testing of timing-related defects by standard interconnect test methods. Therefore, the conception of *at-speed* interconnect test and BIST has become very important lately [4], [5].

The simplest way to approach the problem is to place a standard LFSR (Linear Feedback Shift Register) as a test stimuli generator on one side of interconnect wires and a MISR (Multiple Input Shift Register) as a response compactor on the other side. This approach has been successfully adopted by the industry due to its simplicity. However, such a solution cannot be easily standardized as, for example, the BS due to the following reasons. First, the LFSR as the TPG is not easily scalable since its efficiency depends on characteristic polynomials, which should be selected separately for each board configuration. The same hold for the MISR which fault signature also depends on the configuration. Secondly, there is the aliasing probability for MISR signatures, which might prevent fault detection. It may also reduce the fault diagnosis resolution of the method. Third, there is always a question of how many pseudo-random test patterns would be enough to detect all possible static and timing-related faults. Usually, the LFSR is set for generation of several thousands of patterns, which is usually considered as a good enough test length. Hence, this solution is not the most efficient one in terms of test application time.

Another direction of research in this area is focused on modification of the standard BS cells so that the resulting architecture is capable of sending the test vector and capturing the response within a single clock cycle [6]. There are approaches for single as well as for multiple clock schemes. Such a solution complicates the test application mechanism of BS and brings additional silicon area. It should also be pointed out, that the test data is shifted serially into the BS chain. At first, the desired vector is shifted in, then it is applied to the nets and

finally the response is shifted out and analyzed. This is repeated for each pattern and requires at least $T \bullet L$ clock cycles, where T is the test length and L is the length of the BS chain. At first, L is a large number usually ranging between 10^2 and 10^3, which results in very long test application times compared to the initial test length T. Secondly, this framework does not directly support the *at-speed testing* of interconnect.

The problem of long test application time has also been studied and some solutions have been proposed in literature [7]. The common idea, which bounds them, is the usage of limited shift operations between two patterns, that is, a one-bit shift in the best case. All of these solutions, however, suffer from the issue referred as a multiple driver contention problem when several drivers feed the same bus simultaneously with different values. This happens when test data, while being shifted, unintentionally activates conflicting drivers by setting up the corresponding values in control cells situated in the same scan chain and interleaved with data cells.

All the mentioned problems resulted in the fact that there are still very few methodologies, which could be effectively used for at-speed testing of timing-related interconnect faults [3].

This paper describes a novel approach to fault diagnosis suitable for at-speed testing of board-level interconnect faults. The approach is based on a new parallel test pattern generator and a specific fault detecting sequence called the *Interleaved True/Complement code*. The test sequence has tree major advantages. At first, it detects both static and dynamic faults upon interconnects. Secondly, it allows precise on-chip at-speed fault diagnosis of interconnect faults without any aliasing or masking. Third, the fault signature needs to be shifted out from the RA just once per test session. Finally, the hardware implementation of both the test generator and the response analyzer is very efficient in terms of silicon area.

The proposed framework also reduces the test application time from original $T \bullet L$ clock cycles to T clock cycles, plus less than L clock cycles needed for shifting out the final fault signature.

The next section of the article gives an overview of known test generation methods for interconnect testing and describes a new sequence for testing timing-related faults called the Interleaved True/Complement code. Section 3 gives an overview of widely accepted test application scheme for interconnect BIST and a new original hardware implementation of the TPG and the RA for at-speed IBIST. Conclusions are given in the end of the article.

2. Test generation algorithms

In interconnect test it is of common practice to consider the following defects: *a)* shorts between multiple nets and *b)* opens upon single or multiple nets. Shorts are usually modeled by wired-AND (logic 0 dominates) or wired-OR (logic 1 dominates) faults while opens are represented by stuck-at-1 or stuck-at-0 faults.

In 1974, Kautz showed that in order to test N independent nets for all possible shorts one needs $\lceil \log_2(N) \rceil$ test patterns [8]. The main idea was to assign a unique code word to each net. Then, in case of a wired-OR (wired-AND) short between any two or more lines the number of 1s (0s) will be increased and the initial code will be distorted. The algorithm proposed by Kautz is called the *Counting Sequence* algorithm (Fig. 1a).

It has been shown later [9] that stuck-open faults could also be tested by this algorithm but then the all-1 and all-0 patterns should be avoided. This means that the counter should start from 1 rather than from 0 and it should finish before the all-1 pattern appears. Therefore, the number of required patterns increases in some cases (as it is shown in Fig. 1b) and becomes equal to $\lceil \log_2(N+2) \rceil$. Such a test called the *Modified Counting Sequence* guarantees the *detection* of all possible opens and shorts with a minimal number of test patterns. However, it does not guarantee the accurate *diagnosis* due to aliasing problem. Aliasing appears when some faulty response is equal to some correct response. Then it is not clear if that wire yielding the correct response is also faulty or not.

Later, Wagner [10] proposed an approach where the Counting Sequence is applied two times: once in its original form and then inverted. This algorithm is called the *True/Complement* algorithm and is illustrated in Fig. 1c. The code words applied to each line contain equal

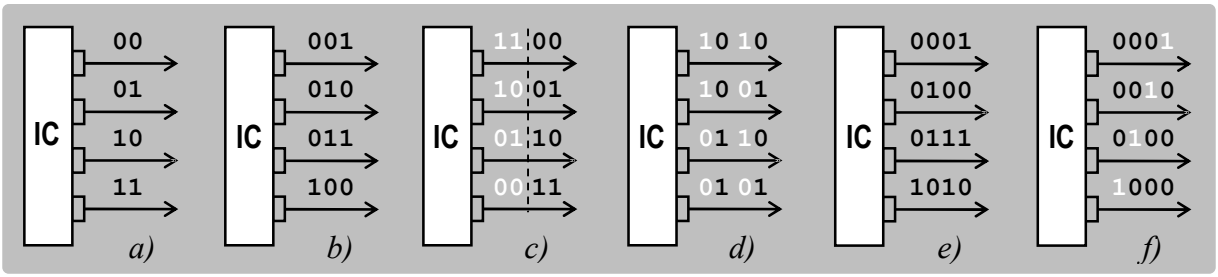

Figure 1. Different test pattern generation algorithms

quantities of 0s and 1s. These quantities are always increased or decreased when nets are getting shorted or broken. The lost balance between the number of 0s and 1s unambiguously guarantees the diagnosis of all shorts and opens without aliasing. In this approach all-0 and all-1 codes are not forbidden anymore, because they are getting complemented later, which makes the test length being equal to $2\lceil log_2(N) \rceil$.

It could be easily noted that if such a code is applied in a proper way, all the interconnect delay faults will be also captured. However, then the first and the last codes should still be avoided, since they do not produce both $1\rightarrow0$ and $0\rightarrow1$ transitions needed to sensitize both transition faults [6]. In our framework we are going to use a new algorithm, which we call an *Interleaved True/Complement* sequence. In the interleaved sequence the complemented vector comes right after its true-valued counterpart. We call such pair of vectors the *True/Complement Couple*. In figure 1d they are shown in white and black colors respectively. Such a code preserves all the diagnostic properties of its predecessor (the True/Complement code) but at the same time it has several additional useful features.

Theorem 1. The Interleaved True/Complement sequence of length $n \geq 4$ provides both $1\rightarrow0$ and $0\rightarrow1$ transitions on the interconnect lines.

Proof. At first, it is clear, that each true/complement couple always activates one of the two transitions upon any selected wire. Let this transition be $0\rightarrow1$. Then there are exactly two possible cases for the second couple at the same wire: $1\rightarrow0$ and $0\rightarrow1$, where the first case automatically yields the second required transition. In the second case, the required $1\rightarrow0$ transition appears *between* the two couples, i.e. $0\rightarrow\mathbf{1\rightarrow0}\rightarrow1$. The same holds for the case, when the first transition is $1\rightarrow0$. For any sequence longer than 4, both required transitions are contained in the first 4 vectors and then repeated with each additional couple■

In fact, all the four possible cases for the four-pattern Interleaved True/Complement sequence are illustrated in Figure 1d.

The fact that each true/complement couple of vectors always drives a wire to both the 0 and the 1 values within two adjacent clock cycles allows efficient design of the response analyzer, which becomes independent and produces exact diagnosis being totally isolated of other wires (unlike it is in MISR for example).

For crosstalk testing it might be important that there were both complemented values upon adjacent wires somewhere in the sequence, i.e. at least once the first wire should be set to 0 while the second was set to 1 and once - vice versa.

Theorem 2. The Interleaved True/Complement sequence always guarantees that at least once both complemented values 0/1 and 1/0 appear upon any two pair of independent interconnect nets.

Proof. For the Counting Sequence, it is known that the code words upon any two nets are different at least in one bit position. Let the difference be 0/1. In the True/Complement sequence each bit in the code word has its complemented counterpart. This yields the corresponding 1/0 pair of complemented values upon the same two nets■

It could be shown in a similar way that the Hamming distance between any two code words in the True/Complement code is at least 2. It is well known from the coding theory, that the larger the Hamming distance, the higher the probability of detection and diagnosis of faulty signals. This holds especially for the harmful influence exerted upon the interconnect wires by crosstalk, switching noise, or other electromagnetic phenomena. This is also important for such short faults where neither the 1 nor the 0 driver is dominant.

The test length of the proposed Interleaved True/Complement sequence is the same as the original True/Complement one has, i.e. $2\lceil log_2(N) \rceil$.

There is another algorithm known to yield code words with a Hamming distance of at least 2, and at the same time requiring less test patterns [11]. It is called the LaMa algorithm and it generates the test of length $\lceil log_2(3N+2) \rceil$. This algorithm is similar to the Modified Counting one but increments by 3 rather than 1 (see Figure 1e). This algorithm, however, is not directly suitable neither for interconnect delay fault testing, nor for crosstalk. The former is true because the two required transitions ($0\rightarrow1$ and $1\rightarrow0$) upon an interconnect wire are not guaranteed.

Table 1. Properties of TPG Algorithms

	Counting Sequence	**Modified Counting**	**True/Compl Algorithm**	**Interleaved True/Compl**	**LaMa Algorithm**	**Walking Sequence**
Number of vectors	$\lceil \log_2(N) \rceil$	$\lceil \log_2(N+2) \rceil$	$2\lceil \log_2(N) \rceil$	$2\lceil \log_2(N) \rceil$	$\lceil \log_2(3N+2) \rceil$	N
Hamming distance (min)	1	1	2	2	2	2
Defect detection	shorts	shorts, opens	shorts, opens, (delays)	shorts, opens, delays	shorts, opens	shorts, opens, (delays)
Diagnostic properties	bad	bad	good	good	average	best

The testing of crosstalk is not guaranteed due to the fact that the two bit positions where the two code words are different could have the same difference, i.e. both 0/1 or both 1/0. This case is illustrated in Figure 1e where the code words upon the second and the third wire have only the 0/1 difference in both bit positions. The fault diagnosis properties of the LaMa algorithm are similar to the ones of the Modified Counting method with the only difference in the Hamming distance.

Another algorithm - the Walking Sequence is probably the simplest one to automatically generate. There are two types – walking 0 and walking 1. The latter is illustrated in Figure 1f. In a slightly modified way, the walking sequence keeps most important diagnostic properties like the Hamming distance and the required transitions for delay fault testing. At the same time, however, it is one of the longest test sequences used in practice, since it features the length equal to *N*.

The summary of properties of the discussed algorithms is given in Table 1. There are also other test sequences proposed by different authors (for example [7] and [15]) but they are mostly modifications to these basic ones.

It is seen from the table that there are three algorithms with comparatively good fault detection and diagnostic properties. These are both True/Complement algorithms and the Walking Sequence. Two of them require additional vectors in order to be able to detect delays upon all the nets. This makes the Interleaved True/Complement code be the shortest sequence to detect also delay faults in addition to commonly considered shorts and opens. The diagnostic properties of the Walking Sequence are the best among these codes, but its length is the largest one.

3. At-Speed Interconnect BIST Hardware

Since the introduction of the IEEE 1149.1 Boundary Scan (BS) standard the powerful and universal BS architecture became the main mechanism of the interconnect test. There are quite a lot of publications [4,6,7,12-14] suggesting efficient combination of the BS and IBIST. In most cases, on-board TPGs are used for feeding the BS shift register with test patterns. In [7], for instance, authors extensively study the problem and suggest different IBIST hardware realizations for the True/Complement and the walking 1 sequence.

Most of later works [12-14] consider the problem of multiple driver contention in the tri-state environment and on-board fault diagnosis. Usually they apply the True/Complement sequence modified in a certain way to avoid the contention. The test data then represents composite vectors [12] generated by two types of TPGs: C-TPG (for control signals) and D-TPG (for data signals) [13] (see Fig. 2a).

The general structure of the D-TPG is given in Fig. 2b. It consists of two counters. The first one generates the Counting Sequence while the second one selects current bit (current vector). The vectors are then generated serially and fed into the Boundary Scan chain intermixed with control signals coming from the C-TPG. As it was previously discussed such a technique needs in general $T \bullet L$ clock cycles to complete, where T is the test length and L is the length of the BS chain. Moreover, it does not directly support the *at-speed testing* of interconnect.

Basically there are four key elements in a successful at-speed testing solution. The first is the selection of the test sequence, which should detect all the necessary timing-related faults. The second is the parallel test pattern generation, i.e. each new pattern should be formed in the TPG at each new clock cycle (like in LFSR) and without any intermediate shift operations. Third, the response analysis must also be made at-speed and in parallel. Finally, the whole solution must be efficient in terms of silicon area. This section describes the hardware implementation of such a TPG for the Interleaved True/Complement code and the RA hardware used for fault diagnosis. The TPG forms test vectors right on chip (not on board) replacing

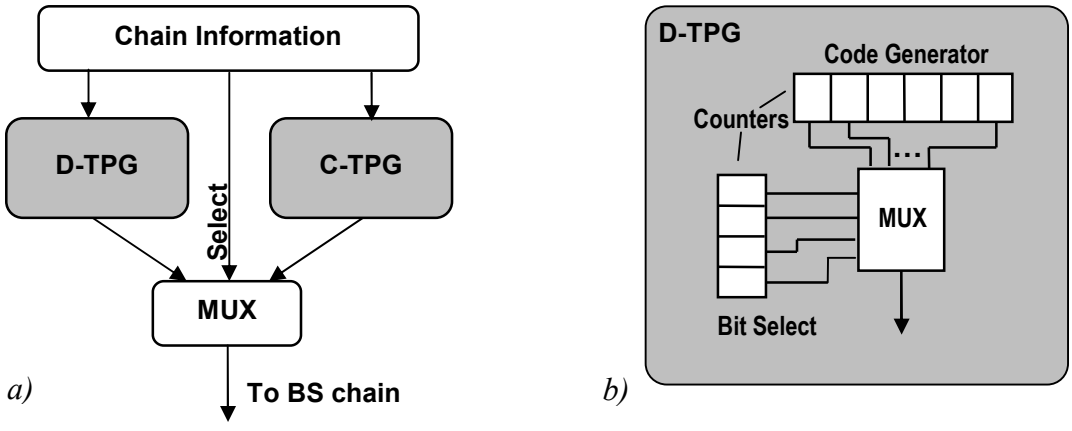

Figure 2. Partitioned serial TPG for BS based IBIST: a traditional approach

Figure 3. Serial vs. parallel test generation

and enhancing the functionality of D-TPG. The control data should still be formed in C-TPG as it is done in [13]. The main idea of the TPG architecture is based on a circular shift register and its proper initialization. The fault diagnosis is made right on-chip and only shifted out at the end of the test session. Compared to other known techniques, current approach provides the minimal hardware cost at the shortest test application time.

The main idea behind the TPG is illustrated in Fig. 3. Let us first consider the parallel generation of the Modified Counting Sequence. Fig. 3a illustrates the way it is generated by the counter in D-TPG. The arrow shows the direction of test generation, which is orthogonal to the direction of test application (Fig 3e). Therefore, each parallel test vector (the vector to be applied to the nets) needs to be generated bit by bit (Fig. 2b) each time before it is applied. Hence, the common counter-based approach (Fig. 2b) is not applicable here.

It is well known that the *fully configured* LFSR generates exactly the same 2^k different vectors except the all-zeros (see Fig. 3b). Since, all-zeros along with all-ones are forbidden code words in the Modified Counting Sequence, there is nothing to loose. However, as the code words coming from LFSR are of different order compared to the ones generated by counter, the all-ones code still might appear in the middle of the sequence and invalidate the test. Therefore the all-ones vector should be taken as the initial state that will be just skipped. In this case all the

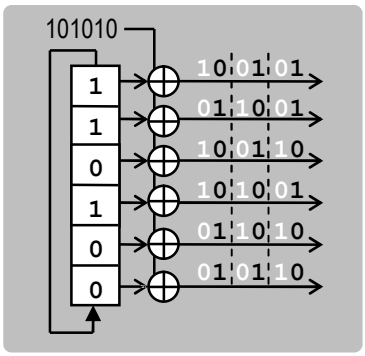

Figure 4. At-speed TPG

rest sequence will be valid (Fig. 3c). In this way, an LFSR, being cheaper in hardware can replace the counter for TPG purposes (also in BS based approaches). However, still it does not fit for parallel test pattern generation.

When one examines the patterns (columns) in Fig. 3c, one may notice that the middle one is exactly the rightmost one, but shifted circularly by one bit upwards. The same holds for the leftmost pattern compared to the middle one. It can be proved that by placing the leftmost (or the rightmost) vector into the circular shift register (without additional feedbacks) and applying $\lceil log_2(N+2) \rceil$ clocks the resulting test sequence generated in parallel will be exactly the same as the orthogonal one, generated by the corresponding LFSR.

In this way we achieve two goals: *a)* parallel test vector generation, *b)* reduction of TPG hardware cost compared to the counter-based approach. The resulting test maintains all the properties of the Modified Counting Sequence, which guarantees the detection of all the considered fault types except delay faults. Moreover, now it can be generated and applied at-speed.

The IBIST TPG hardware realization for the Interleaved True/Complement sequence is illustrated in Fig. 4. First, the shift register gets initialized in the same way as it was discussed above. Then it works at half the frequency and holds each vector for two clocks. The outputs of the

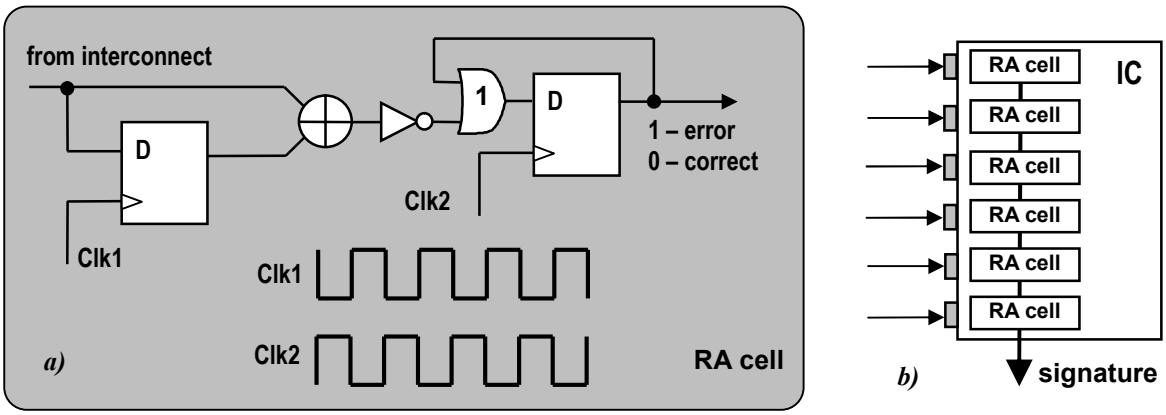

Figure 5. Response analysis hardware

register are connected to XOR gates where they get inverted at each second clock by an external signal 010101… coming from an oscillator. The black and the white figures upon the interconnect lines represent the true and complemented values of the corresponding vector. In Fig. 4, it can be clearly seen that the hardware complexity of the TPG is approximately the same as of the LFSR.

Now we are ready to approach the main result of this work – the board configuration independent at-speed precise fault diagnosis. As it was discussed above, the True/Complement Couples always contain a pair of 0 and 1 on each interconnect line. In case of a fault, some of such couples on a faulty line will be broken in such a way that they will contain either only 0-s or only 1-s. Therefore, for successful fault diagnosis it is enough to consider each couple separately of other couples. At the same time, each net can be considered independently.

Figure 5a shows possible hardware implementation of such a separate response analyzer for a single net. It consists of two D flip-flops driven by two antiphased clock signals. In this way the first part of the couple is stored in the first D flip-flop in order to be XOR-ed with the second part as soon as it comes along the interconnect line. The opposite values (the correct behavior) produce logic '1' at the output of the XOR gate, which becomes 0 after the inverter. At the same time the second D flip-flop becomes sensitive and that 0 will be captured inside.

As soon as there will be equal values in a couple (the erroneous behavior) the first part of the circuit will generate the error signal, which is equal to logic '1'. This error signal will be then captured inside the second flip-flop and it will be kept there until the end of the test session. In such a way, each faulty interconnect line will be explicitly identified by logic '1' in the corresponding RA cell. After the application of all the test patterns the final fault signature should be shifted out as shown in Fig 5b.

4. Conclusions

The main goal of this paper is to propose a new diagnostic solution for interconnect BIST, which is suitable for at-speed testing and for detection both static and dynamic faults. The idea is based on a new interconnect test sequence called the Interleaved True/Complement code and original hardware implementation of the TPG and the RA.

The TPG is capable of generating test vectors in parallel similarly to the pseudorandom vectors formed in LFSR. The main idea behind it is based on using a circular shift register and its proper initialization. The major advantage of this TPG compared to LFSR is the test lengths and the fault detection and diagnosis properties originated from the Interleaved True/Complement code. At the same time the vectors are generated in parallel, which reduces the testing time from original $T \bullet L$ clock cycles to T clock cycles. This generator forms test vectors right on chip (not on board) replacing and enhancing the functionality of D-TPG commonly used in IBIST.

Presented framework guarantees detection and diagnosis of short, open, and delay faults on interconnect lines by at-speed generated and applied test sequence of length $2\lceil log_2(N+1) \rceil$. The fault diagnosis is made right on-chip and only shifted out at the end of the test session.

The hardware implementation of both the test generator and the response analyzer is very efficient in terms of silicon area.

Acknowledgments

This work was supported by the Estonian Science Foundation Grants No 5910 and 5649.

References

[1] IEEE 1149.1-1990, "IEEE Standard Test Access Port and Boundary-Scan Architecture," 1990.

[2] E.J. Marinissen, B. Vermeulen, H. Hollmann, R.G. Bennetts, „Minimizing Pattern Count for Interconnect Test under a Ground Bounce Constraint," in *IEEE D&T of Comp*., March-April 2003, pp. 8-18.

[3] R.Pendurkar,A.Chatterjee,Y.Zorian,"Switching Activity Generation with Automated BIST Synthesis for Performance Testing of Interconnects," *IEEE Trans CAD/ICS*, vol.20, no.9, 2001.

[4] B.Nadeau-Dostie,et.al,"An Embedded Technique for At-Speed Interconnect Testing," in *Proc.ITC'99*, pp.431-438.

[5] A.Attarha,M.Nourani,"Testing Interconnects for Noise and Skew in Gigahertz SoC," in *Proc of ITC'2001*, pp.305-314.

[6] S.Park, T.Kim, "A new IEEE 1149.1 boundary scan design for the detection of delay defects," in *Proc. DATE'2000*, pp. 458-462.

[7] A.J.Hassan,J.Rajski,V.K.Agrawal,"Testing and Diagnosis of Interconnects…," *Proc. ITC'88*, 1988, pp.126-137.

[8] W.H. Kautz, "Testing of Faults in Wiring Interconnects," *IEEE Trans. Computers*, vol. 23, no. 4, 1974, pp. 358-363.

[9] P. Goel and M.T. McMahon, "Electronic Chip-in-Place Test," *Proc. ITC 82, IEEE Press*, 1982, pp. 83-90.

[10] P.T. Wagner, "Interconnect Testing with Boundary Scan," *Proc. Int'l Test Conf., IEEE Press*, 1987, pp. 52-57.

[11] L. Eerenstein and M. Muris, *Method for Generating Test Patterns to Detect an Electric Shortcircuit, a Method for Testing Electric Circuitry While Using Test Patterns So Generated…,* US patent 5636229, Washington, D.C., 1997.

[12] C.Su, S.W.Jeng, Y.T.Chen, "Boundary scan BIST methodology for reconfigurable systems," in *Proc. ITC'98*, *IEEE Press*, 1998, pp. 774-783.

[13] W.Feng,F.J.Meyer,F.Lombardi,"Novel control pattern generators for interconnect testing…," in *Proc. Int'l Symp. Defect and Fault Tolerance in VLSI*, 1999, pp. 112-120.

[14] C.Su,W.Tseng,"Configuration Free SoC Interconnect BIST Methodology," in *Proc. ITC'2001*, pp.1033-1038.

[15] W.T. Cheng, J.L Lewandowski, E. Wu, "Diagnosis for Wiring Interconnects," in *Proc. ITC'90*, pp. 565-571.