JOURNAL OF ELECTRONIC TESTING: Theory and Applications 23, 341–355, 2007
© 2007 Springer Science + Business Media, LLC. Manufactured in The United States.
DOI: 10.1007/s10836-007-0759-5

IEEE Standard 1500 Compatible Oscillation Ring Test Methodology for Interconnect Delay and Crosstalk Detection

KATHERINE SHU-MIN LI
Department of Computer Science and Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan, Republic of China
smli@cse.nsysu.edu.tw

CHUNG-LEN LEE
Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan, People's Republic of China

CHAUCHIN SU
Department of Electronic Control, National Chiao Tung University, Hsinchu, Taiwan, People's Republic of China

JWU E CHEN
Department of Electrical Engineering, National Central University, Chungli, Taiwan, People's Republic of China

Received June 3, 2005; Revised July 15, 2006

Editor: M. Breuer

Abstract. A novel oscillation ring (OR) test scheme and architecture for testing interconnects in SOC is proposed and demonstrated. In addition to stuck-at and open faults, this scheme can also detect delay faults and crosstalk glitches, which are otherwise very difficult to be tested under the traditional test schemes. IEEE Std. 1500 wrapper cells are modified to accommodate the test scheme. An efficient algorithm is proposed to construct ORs for SOC based on a graph model. Experimental results on MCNC benchmark circuits have been included to show the effectiveness of the algorithm. In all experiments, the scheme achieves 100% fault coverage with a small number of tests.

Keywords: oscillation ring (OR) test scheme, interconnect test, SOC testing, stuck-at faults, open faults, delay faults, crosstalk glitches, IEEE P1500, wrapper cell design

1. Introduction

Rapid advance in the VLSI technology has rendered delay caused by interconnects to surpass that caused by transistors [1]. Interconnects have become the key element in determining circuit performance and signal integrity, especially for SOC ICs. In addition, reduced spacing between adjacent interconnects makes crosstalk an important source of anomaly in deep submicron VLSI [2–6]. It can induce glitches and extra delays for signals propagating along the interconnection lines. Buffer insertion is proposed to alleviate the problems associated with long signal line. As a result, signal lines used for communication consist of not only wire segments, but also logic gates [7–9].

Traditional test methods are mainly designed for functional check, and thus the signal integrity issue is usually not the main target. As a result, crosstalk and delay faults are difficult to detect under conventional test methods. The detection of crosstalk-induced glitches usually involves precise measurement of signals on the victim nets [10, 11], while complex clock control is

needed for delay fault detection due to the two-pattern tests [12, 13]. Therefore, much more extra effort has to be devoted to the detection of errors due to these problems.

Cores are usually provided with either predefined test vectors or built-in self-test (BIST) mechanism, so that the SOC system designers only need to consider how to apply test and control the test process. On the other hand, the interconnection structure of an SOC is designed by the system integrator, who is also responsible for defining the test set for interconnect. Interconnect testing occupies an important part in system and chip design [14]. Plenty research works on interconnect testing can be found in the literature. Earlier works in interconnect testing were targeted for board-level testing [14–18]. These papers described fault models and test generation algorithms for general interconnect structure. However, it is very difficult to apply these interconnect testing methods under SOC environment without design-for-testability (DFT) support. The IEEE Std. 1500 standard [19, 20] provides structural support for core testing as well as interconnect testing in SOC. The IEEE Std. 1500 SOC test environment consists of wrappers around cores in the SOC, and an optional centralized test access mechanism (TAM). The TAM defines the test control, while wrappers provide a standardized interface for test data transmission. The SOC test standard IEEE 1500 extends IEEE 1149.1 Boundary Scan test methodology so that interconnect test for SOC can be conducted in a way similar to those used in board-level interconnect test. In this approach, all pins of a core are replaced by wrapper cells, so that a scan path connecting all the pins can be formed during the test mode. In this way, test vectors can be applied to interconnection lines, and test results are captured and observed outside the core, and are propagated to the ATE for inspection. However, the proposed core test standard is designed for traditional test methodology, and the signal integrity issue is not considered under this framework. For example, if we need to apply two-pattern test to detect delay fault, we need to modify not only the wrapper cell structure but also the clock control so as to apply tests and capture responses correctly. The hardware overhead can be significant.

To solve this difficulty, we propose an oscillation-based test scheme and structure for interconnect in SOC ICs. Oscillation ring (OR) test is a useful and efficient method to detect faults in functional circuits [21–26]. An oscillation ring is a closed loop, which has *odd number of signal inversions*, of the circuit under test. Once the ring is constructed during the test mode, an oscillation signal appears on the ring. For a circuit with stuck-at and open faults, oscillation stops; and for a circuit with gate or path delay faults, the oscillation frequency is different from the fault-free case. By observing the oscillation signal at the output of the circuit, it can tell whether the circuit is faulty or not. There are several important works on the oscillation ring test scheme, e.g., [21–24]. Most of the results focus on detecting device faults for analog and/or mixed-signal circuits [21–24], or on detecting faults on gate-level circuits [25, 26].

In this approach, we construct a ring that goes through a series of interconnect wires (including inserted buffers) and some internal scan paths in core modules of the SOC. Once a ring with odd inversions is constructed, we can decide whether the ring is faulty by observing the oscillation signal on the ring. Various types of interconnect faults are detectable under this scheme, including stuck-at faults, open faults, delay and crosstalk glitch faults. Furthermore, this scheme can be used for circuit parameter measurement since the path delay can be obtained by measuring period of the oscillation signal. Fault diagnosis is also achievable with properly selected rings since fault can be located as multiple different rings passing it all fail.

In order to support the OR test, we modify IEEE Std. 1500 wrapper cell designs. We also develop an efficient algorithm to select interconnects to form the minimal number of oscillation rings to reduce test time. Experimental results on MCNC benchmark circuits show that the algorithm achieves 100% fault coverage with small number of tests. These results show that the proposed method is not only feasible with small hardware overhead, but also efficient in fault detection.

The paper is organized as follows. In the next section, we present the test architecture for oscillation ring based interconnect test, and give the modified IEEE Std. 1500 wrapper cell designs. A graph model of interconnect hypernet structure is discussed, and some related theoretical analysis is also given in Section 3. An efficient ring-generation algorithm that selects interconnects to form a minimal number of oscillation rings to minimize test time is presented in Section 4. Experimental results on MCNC benchmark circuits show that the algorithm achieves 100% fault coverage with a small number of tests in Section 5. Finally, some concluding remarks are given in Section 6.

2. Interconnect Test Architecture for Oscillation Ring Test

In this section, we propose the architecture for the oscillation ring test for interconnects.

2.1. General Architecture

In order to apply the proposed method for interconnect testing in SOCs, it is assumed that all cores are wrapped in this paper. It is first assumed that the top-level SOC circuitry in between the wrapped cores consists of wire interconnects only, while the condition in which glue logic exists in the interconnect structure will be explored later in this section.

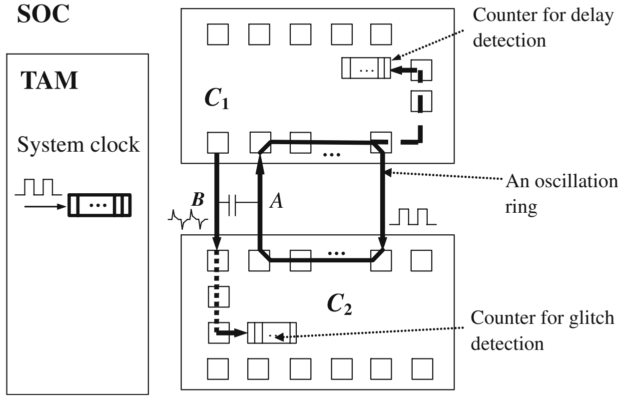

Fig. 1. Test architecture for interconnect crosstalk detection and delay measurement.

Figure 1 shows the proposed architecture, where C_1 and C_2 are cores implemented with boundary scan cells and a local counter, which is to capture the induced glitches for crosstalk fault detection and to measure delays of oscillation rings for delay detection. For this architecture, oscillation ring(s) will be formed as shown during the testing mode. If the formed oscillation ring fails to oscillate, it implies that there exists stuck-at or open fault(s) in components of the oscillation ring. The period of the oscillation signal is measured by using a delay counter in a local core to test wire delay faults, and a similar scheme is also applied for crosstalk glitch detection. If there is a crosstalk fault between a victim interconnect line and the oscillation ring interconnect lines, glitches will be induced on the victim interconnect line. Figure 2 shows the oscillation signal at the oscillation ring and the induced glitches at the victim interconnect line. These induced glitches will be captured by the local counter of the core and be shifted out for observation.

To test the delay fault, the delay of the oscillation ring will be measured through using the local counter and the central counter of TAM of the SOC. At this time, the central counter is enabled by signal *OscTest* and triggered by the system clock, and a local counter is connected to one wrapper cell of the oscillation ring so that the oscillation signal is fed to the local counter. When the oscillation test session starts (OscTest=1), the central counter as well as all local counters in cores are enabled. After the counter in TAM counts to a specific number n, the oscillation test session terminates and all local counters are disabled (OscTest=0). The counter contents are shifted out to an ATE for inspection.

Assume that m oscillation rings are tested. Let the frequency of the system clock be f, and the delay counter contents of the rings be n_1, n_2, ..., n_m, respectively. An estimation of the ith ring's oscillation frequency f_i can be approximated by

$$f_i = f \times n_i / n \tag{1}$$

Since the frequency of each ring is predetermined during the design phase, a wire delay fault is detected and measured by inspecting the contents of the delay counters. Let the oscillation frequency of the rings, according to the timing specification, be $f_{\min} \le f_i \le f_{\max}$, with the unit of measuring T_0 (= n/f). Thus, we have $n_{\min} \le n_i \le n_{\max}$, where $n_{\min} = f_{\min} \times T_0$ and $n_{\max} = f_{\max} \times T_0$. Let ξ be the resolution of delay measurement, and ε be the maximum measurement error. Since a counter's maximum measurement error is ±1, the requirement for ε should be the reciprocal of $f_{\min}$ and T_0.

$$\varepsilon = \frac{1}{f_{\min} \times T_0} \le \zeta \tag{2}$$

An example for delay measurement is given as follows. Let the frequency specification of the oscillation rings be 4 – 400 MHz and ξ be 0.001, implying the counter content $n_{\min}$ is at least 1,000. From Eq. 2, we have the required T_0 to be 250 μs. This example illustrates the feasibility of the oscillation test scheme from a measurement prospect, and this frequency specification is actually compliant with ATE specifications.

The crosstalk is caused by excessive coupling capacitance between adjacent wires, and it can incur two types of errors: glitch and delay [27–29]. When there is a signal transition in the aggressor while the victim signal is stable, a crosstalk-induced glitch appears in the victim net. On the other hand, a crosstalk-induced delay occurs when the victim net makes a signal transition opposite to the direction of the aggressor net's signal at roughly the same time. The crosstalk-induced delay is just a superposition of the original signal in the victim and the glitch induced by the aggressor [29]. Therefore, it is possible to detect crosstalk-induced delay simply by detecting induced glitches [29].

In order to detect the crosstalk glitch in Fig. 1, consider wires A and B in Fig. 1 and assume that there is a coupling crosstalk effect between *victim nets B* and the *aggressor nets A* of the oscillation ring. Interconnects adjacent to an oscillation ring are affected by the oscillation signal if there is an excessive coupling capacitance between

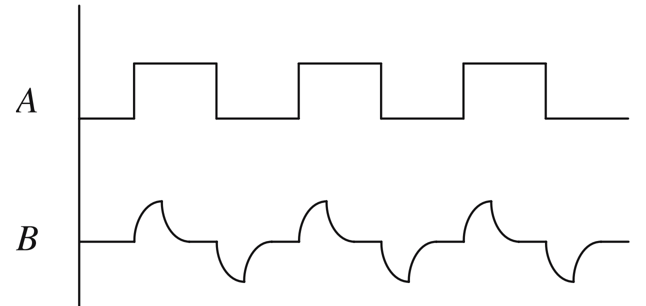

Fig. 2. The oscillation signal on wire A and the induced glitches on the victim interconnect B.

these two wires (A and B) [29]. When the oscillation signal occurs, crosstalk-induced glitches appear on the victim net B. Similar to the case for wire delay detection, the glitches on net B are sent to local counters in core C_2 through a series of modified wrapper cells. Since there is an inverter per modified wrapper cell in the OR test mode, the induced glitches are amplified when the glitches pass through the wrapper cells, and the amplified glitches are used to trigger the local counters in core C_2 for glitch detection.

2.2. *Crosstalk Fault Detection*

In order to verify that the proposed architecture can be applied to detect crosstalk-induced glitches, we conduct HSPICE simulation with TSMC 0.18 μm technology. An oscillation signal is generated on a ring as shown in Fig. 1, and a 1 mm wire with three times of normal coupling capacitance is assumed. The results are shown in Figs. 3 and 4. Figure 3 shows the oscillation signal on the ring, the induced glitches on the victim net, and the output of the counter. The crosstalk-induced glitch shown in Fig. 3 can be detected and verified since the counter changes the state on every positive glitch.

Figure 4 gives an illustration on how to detect the glitches. The oscillation signal is shown in top of Fig. 4, and the induced positive glitch, whose peak value is about 0.8V, is shown in the middle set of figures. This glitch is amplified by a detector, which is a specially designed inverter in our IEEE 1500-compliant input wrapper cell. We may adjust the W/L ratio of the detector's transistors to determine the detection threshold of glitches [29]. For example, in our experiment we set $(W/L)_{pu}/(W/L)_{pd}$ to be 1/4. In other words, the width of the pull-down nMOS is four times that of the pull-up pMOS, while the channel lengths of both transistors are set to the minimum. Since the positive crosstalk glitch and the negative glitch are symmetric, we only need a design to detect either a positive glitch or a negative glitch. Here we just give the basic detection principles for the positive glitch detection shown in Figs. 3 and 4.

The detector's output is passed through a chain of wrapper cells. In our experiment, there are five wrapper cells in the chain, and it can be seen that a near rectangular pulse is formed. This pulse is used to trigger a two-port T-type flip–flop (2P-TFF) successfully without causing any setup/hold time violation. The 2P-TFF can be triggered by two different signals, one port is triggered by the crosstalk glitch signal and the second port is trigger by the system clock to scan out the counter contents. In the oscillation test mode, this 2P-TFF is triggered by the amplified glitches and acts as a counter. When we need to scan out the counter contents, it is triggered by the system clock. All the transistors, except for the detector, are minimum-sized.

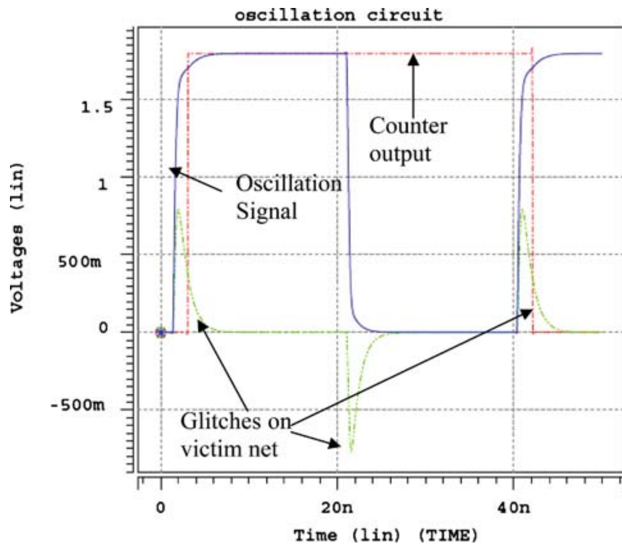

Fig. 3. Oscillation signal on the ring, induced glitches on the victim net, and counter output.

2.3. *Modified Wrapper Cell Design*

An oscillation ring for interconnect test consists of interconnect wires and part of the scan path in each core where the ring passes. Therefore, a wrapper cell must provide a path between input/output ports and scan in/ scan out ports. If oscillation test is used to test wires attached to/from pads, the boundary scan cells also have to be modified in a similar way. In order to facilitate the scheme, the IEEE Std. 1500 boundary wrapper cells need to be modified. In this section, the modified wrapper cell design is presented.

A normal wrapper cell provides two types of paths: a scan path connecting all wrapper cells into a shift register, and an interface buffering between core internal and the wire connected to the pin. Whenever oscillation test is applied, a third combination path must be provided. For an input pin, the wrapper cell must connect the pin input (IN) to scan output (SO), while for an output pin, it should connect scan in (SI) to pin output (OUT) during an oscillation test session. Examples of these connections are the four "corners" of the ring in Fig. 1.

The modified wrapper cell design is shown in Fig. 5 for input and output cells. In each cell, two MUXs are added for path selection. For an input wrapper cell, the extra paths are SI→SO and IN→SO, while for an output wrapper cell the extra paths are SI→SO and SI→OUT. The added inverting and non-inverting buffers in output cells are used to provide odd inversions on the oscillation ring path to generate oscillation signals for the OR test. *OscTest* is a global control signal, while *sel* is used in the input wrapper cell and *inv* is used in the output wrapper

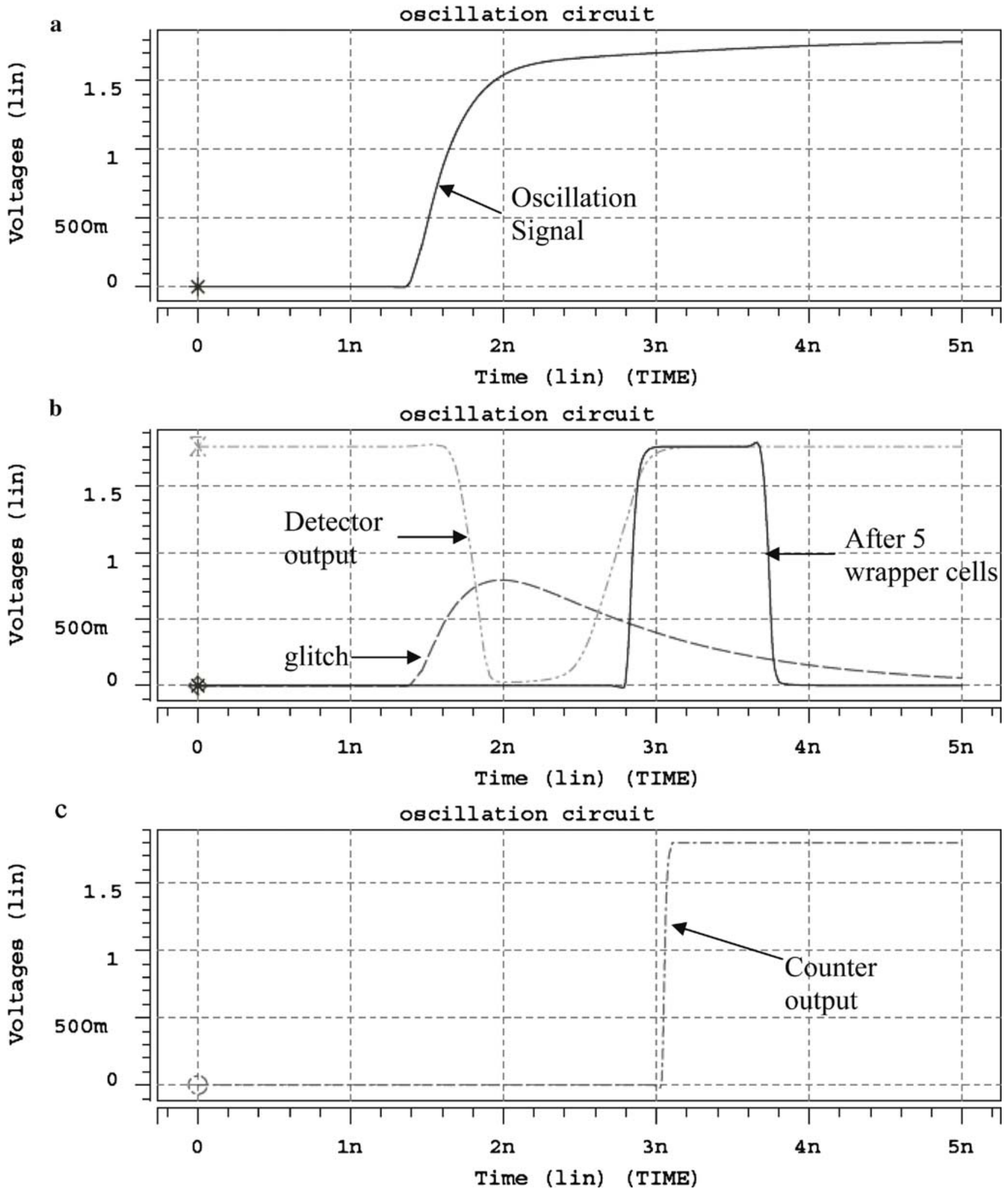

Fig. 4. Illustration on how the glitches are detected, **a** an oscillation signal, **b** the resulting crosstalk-induced glitch, the detector output, and the signal after five wrapper cells, **c** the counter output with the verified state change.

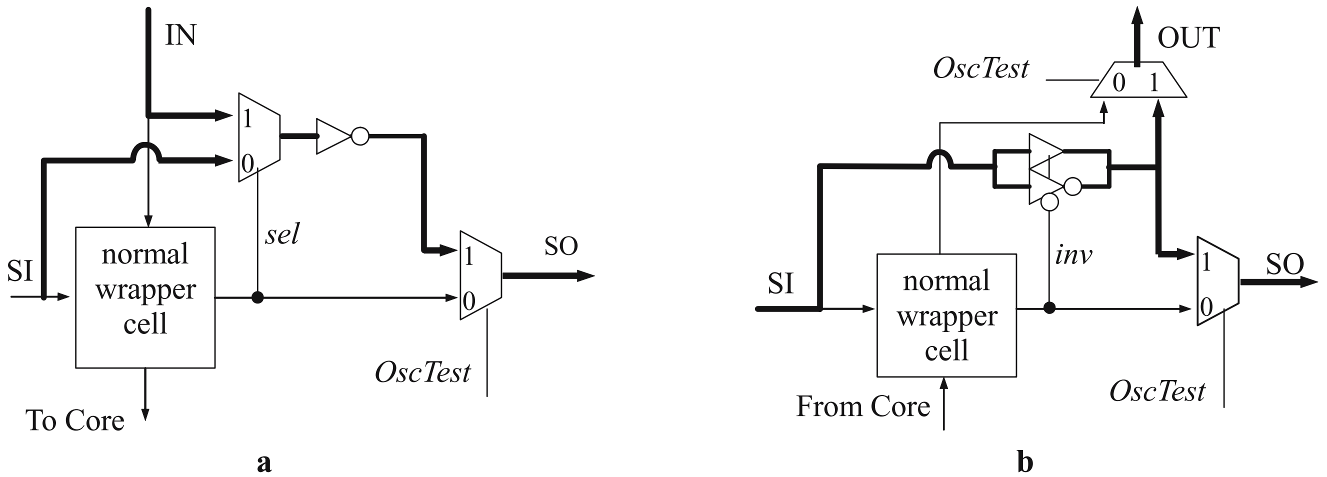

Fig. 5. Modified wrapper cells: **a** input cell **b** output cell.

cell. Signals *sel* and *inv* are individually set and are scanned into the wrapper cells before an OR test session.

In either the normal mode or the IEEE Std. 1500 test mode (OscTest=0), modified cells act as normal wrapper cells. In the OR test mode (OscTest=1), the part of "normal wrapper cell" is bypassed. For an input cell, *sel* is used to select either SI→SO or IN→SO, depending on the position of the input wrapper. If the cell connects an external interconnect to the internal scan path, it is configured as IN→SO. Otherwise, it is configured as SI→SO. For an output cell, the bit information stored in the cell is used for inversion control *inv*, which decides whether the passing signal should be complemented. This is also applied to buffered interconnects where inverters are used for timing closure and signal amplification.

A summary of control signals for the modified wrapper cells shown in Fig. 5 is given in Tables 1 and 2, respectively.

2.4. Coping with Glue Logic

Since cores in an SOC usually come from different sources, the signals among them may not be functionally compatible and thus some logic gates are required at the top level to generate appropriate signals. These gates are known as glue logic, and they are generally unwrapped. When the ring under test contains glue logic, additional care must be taken to ensure that the ring under test indeed can oscillate.

Assume that an interconnect wire in a ring passes through a block of glue logic consisting of combinational gates only. The ring can still oscillate if the path is sensitized properly; in other words, all off-path inputs of the logic gates passed by the path must be set to non-controlling value. An example of the path sensitization is illustrated in Fig. 6, in which the heavy line indicates an interconnect path in the ring under test. All off-path inputs of an AND (NAND) gate passed by the path are set to 1, while off-path inputs of an OR (NOR) gate passed by the path are set to 0. In this way, output of the block will oscillate if the input oscillates. The inputs of the glue logic can be set to the desired values by scanning them to the corresponding wrapper cells. One thing has to be noticed is that the path in the glue logic may create inversion as well, and thus the inversion control of the ring has to be adjusted accordingly.

It may not be possible to sensitize a path whenever storage elements appear in the glue logic. Whenever this is true, an extra path must be applied to bypass a storage element, similar to the connections for the modified wrapper cells shown in Fig. 5.

3. Oscillation Ring Construction: Model and Analysis

To apply the OR testing to an SOC with many cores connected with interconnect lines, it needs to form

Table 1. Control signals for the modified input wrapper cell.

OscTest	sel	Comments
1	1	~IN→SO (OscTest Mode)
1	0	~SI→SO (OscTest Mode)
0	–	normal or IEEE 1500 test mode

Table 2. Control signals for the modified output wrapper cell.

OscTest	inv	Comments
1	1	SI→SO and SI→OUT (OscTest Mode)
1	0	~SI→SO and ~SI→OUT (OscTest Mode)
0	–	normal or IEEE 1500 test mode

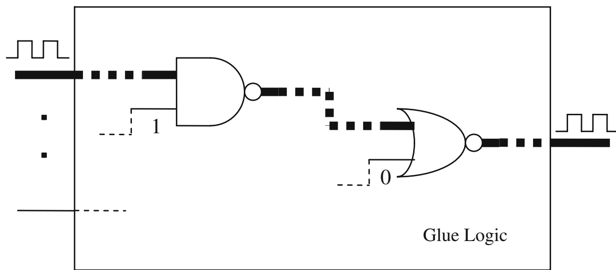

Fig. 6. Path sensitization in the glue logic.

oscillation rings which can cover all interconnects in order to completely test all interconnects of the SOC. In the two sections which follow, we will present the model and analysis and the algorithm to construct oscillation rings respectively.

As mentioned previously, the proposed methodology is targeted for SOC with IEEE Std. 1500. A more detailed example of the test mechanism is illustrated in Fig. 7, where there are three cores, C_1, C_2, and C_3. All pins in a core are connected into a scan path during the test mode, which is indicated by the broken line in Fig. 7a. Oscillation rings can be constructed with the help of scan paths provided by wrapper cells. The interconnect wires connecting cores are also shown by heavy lines in Fig. 7a, where an arrow indicates the direction of signal transmission. There are three nets in the figure, in which net N_1 connects three terminals (pins), while N_2 and N_3 connect two pins each. Only the heavy lines are the target of interconnect test.

For example, there are two rings in Fig. 7a, each corresponding to a circle in the graph mode given in Fig. 7b. The first circle consists of nets N_1 (and its right-hand side branch), N_2, and N_3, and it passes all three cores. The second circle consists of N_1 (and its left-hand side branch) and N_3, and scan paths in C_1 and C_3. In order to make the signal on a ring oscillate, we must ensure that the number of inversions on a ring is *odd*, and this includes the inversions on wire segments as well as those in wrapper cells.

3.1. Graph Model for Oscillation Ring Tests

In order to simplify the problem under investigation, we represent the circuit interconnection by using an abstract *hypergraph*.

Definition 1 A *hypergraph* $G'=(V, L)$ consists of a vertex set V and an edge set L, which consists of multi-

Fig. 7. **a** The interconnect diagram, **b** hypernet graph, **c** graph model with 2-pin nets.

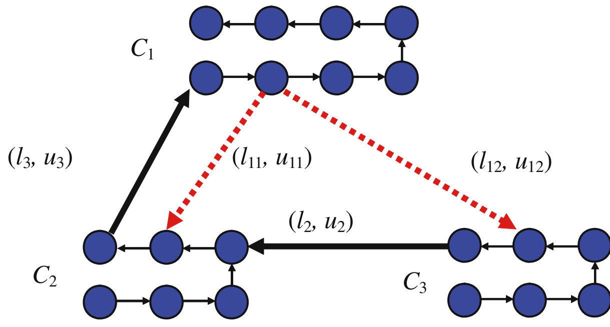

Fig. 8. Graph model for delay faults.

terminal edges connecting a set of vertices $V_i \subseteq V$ and $|V_i| \geq 2$. Such an edge is referred to as a *hypernet*.

For example, N_1 in Fig. 7b is a hypernet. Furthermore, we assume that in an n-terminal hypernet, one terminal is the source node (i.e., sending signal) while the others $n-1$ are the sink nodes (i.e., receiving signals).

The circuit structure of an SOC can be directly transformed into a hypergraph, in which each pin is a vertex while each signal net is a hypernet. However, this graph model is not good enough for our problem. Consider net N_1 in Fig. 7a again. When we apply oscillation test, the two branches of N_1 should belong to two different rings, one consisting of N_1, its right branch, N_2, and N_3, while the other consisting of N_1, its left branch, and N_3. These two rings cannot be tested simultaneously, since the lengths of the two rings are not the same and thus they oscillate at different frequencies. Therefore, it would be better to consider each branch of a hypernet individually or separately instead of treating them all as a whole. Each branch of a hypernet is thus a 2-pin net. For example, nets N_{11} and N_{12} in Fig. 7c are two 2-pin nets, which correspond to hypernet net N_1 in Fig. 7b, and each 2-pin net connects the source vertex to one of its sink vertices. An n-terminal hypernet is thus broken into $(n-1)$ 2-pin nets. The result is a normal graph $G=(V, E)$, where E is the set of 2-pin nets. There are two circles in Fig. 7c: $R_1=\{N_{11}, N_3\}$, $R_2=\{N_{12}, N_2, N_3\}$.

Definition 2 A *weighted graph* $G=(V, E)$ consists of a vertex set V and an edge set E, in which each edge, $e_i \in E$, is an ordered pair (u, v), where $u, v \in V$, and has a *weight* w_i.

A complete test for stuck-at faults and open faults for all interconnections is thus reduced to a problem of finding a set of circles (i.e., rings) that cover all edges corresponding to interconnection structure in the graph G. This is equivalent to find a set of sub-circuits (rings) $R=\{G_1, G_2, ..., G_n\}$, such that:

- $\forall G_i, G_i \subseteq G$, $G_i=(V_i, E_i)$, G_i is a circle.
- $\bigcup_{i=1}^{n} E_i = E$

A minimum test is thus the set of rings with minimum cardinality.

For the delay fault testing, signal delay on each net along the ring is considered. To deal with the delay fault, a weight w_i, which is the timing specification on a 2-pin net e_i by a 2-tuple $w_i=(l_i, u_i)$, where l_i and u_i are lower and upper bound on the distribution of normal path delay respectively, is defined. The graph model for Fig. 7(c) with aforementioned weights is shown in Fig. 8.

Let t_i be the actual propagation delay on net e_i, and the variance of delay on net e_i be $\delta_i = u_i - l_i$. The following lemma gives a sufficient condition under which the delay fault t_i is detectable by applying oscillation test.

Lemma 1: Consider a ring of n edges, $e_1, e_2, ..., e_n$. The delay fault on edge e_i, $1 \leq i \leq n$, is detectable if the following condition holds:

$$t_i - u_i \geq \sum\nolimits_{j \in \{1..n\}-\{i\}} \delta_j \tag{3}$$

Proof: In a fault-free circuit, the maximum delay in a ring will be the summation of the upper bounds of individual nets. A large delay t_i will not be masked if:

$$t_i + \sum\nolimits_{j \in \{1..n\}-\{i\}} l_j \geq \sum\nolimits_{j=1}^{n} u_j \tag{4}$$

Equation 3 can be obtained by rearranging Eq. 4.

From Lemma 1, it can be seen that a delay fault may be masked when $t_i > u_i$ but Eq. 3 is not satisfied. In order to reduce the probability of undetectable faults, we shall try to construct short rings, so that the accumulation of delay variance will not mask delay faults.

3.2. Analysis of Rings and Test Cost

Test cost is dominated by test application time. In the case of oscillation ring test, the single factor affecting test application time is the number of rings required to cover all nets. The number of rings is closely related to interconnect structure. We can analyze it by using the hypernet model.

For a hypernet, at most one of its fanout branches can be tested at a time. The reason is that no two branches

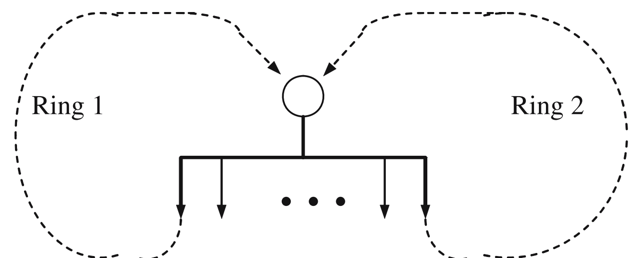

Fig. 9. Hypernet branches and rings.

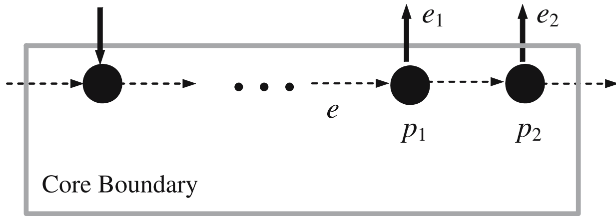

Fig. 10. Rings for adjacent output pins.

belong to the same ring, and any two rings containing the two branches under consideration share the stem of the hypernet before fanout point. If both rings are formed simultaneously, the two oscillation signals will interfere with each other. Therefore, at most one of the fanout branches can be tested at a time. This condition is illustrated in Fig. 9.

Since two interconnect wires are connected by a scan path in a core, the *positions of pins* (i.e. relative position between input pin and output pin) in a core will also affect the interconnection structure. The following lemma explains why *pin location* will affect the number of rings.

Lemma 2: Any two 2-pin nets driven by adjacent pins of a core must belong to two different rings for the oscillation ring test.

Proof: The graph model for two adjacent output pins is shown in Fig. 10. In this figure, a vertex represents a modified wrapper cell for a pin.

Any oscillation ring going out of a core from a given output pin p_i must enter the same core via an input pin whose wrapper cell is a predecessor to p_i in the graph. Thus, any ring going through either edge e_1 or e_2 must pass edge e, which is located on the scan path. Since a ring cannot go through an edge e twice, no rings contain edges e_1 and e_2 simultaneously.

The following theorem gives the minimum number of rings required for a core module.

Theorem 1: Assume that k output pins $p_1, p_2, ..., p_k$ are adjacent to each other in a core. Let the number of fanout branches of the hypernet connected to pin p_i be n_i. The minimum number of oscillation ring tests for interconnect wires attached to theses k output pins is $\sum_{i=1}^{k} n_i$.

The proof of Theorem 1 follows the results of Lemma 2. A lower bound on the number of oscillation rings required to test an SOC can be established as follows:

Corollary 1: A lower bound on the number of rings required for interconnect test is equal to the maximum number of 2-pin nets connected to a sequence of core output pins in all SOC cores.

4. Oscillation Ring Construction: Algorithm

In this section, we discuss how to generate rings for OR test. We analyze the complexity of ring generation algorithms, and propose a heuristic approach for ring generation.

4.1. Exact Algorithm

Since our goal is to reduce test time for the interconnect test, we should find out a set of rings with minimum cardinality. A very naive algorithm for the generation of minimum set of rings can be performed as follows. First, we find all possible rings in an SOC, and the minimum set of rings that cover all 2-pin nets. This approach, however, is not feasible for any reasonably sized SOC due to its high time complexity.

In our graph model, any ring is a sub-graph, which contains a subset of all 2-pin nets. The number of all possible rings grows exponentially as the number of 2-pin nets increases, and thus the first part of the naïve algorithm is already intractable. The problem of finding out minimum number of rings covering all 2-pin nets can be mapped to a column-covering problem, in which each column is a ring and each row represents a 2-pin net. However, it is well known that the column-covering problem in NP-complete.

A more refined exact algorithm can be conducted similar to Quine–McCluskey algorithm for two-level logic optimization. A ring is redundant if it is contained in other rings. In order to reduce the search space, we shall start with "prime" rings that are not contained in other rings. However, the problem of finding out "prime" rings is still difficult. For example, if there exists a Hamiltonian cycle in a graph, the cycle must be a prime ring. Unfortunately, searching for a Hamiltonian cycle is also NP-complete.

From the above discussion, it is obvious that any exact algorithm for searching a minimum set of rings can only work for small circuits. For larger SOC ICs, heuristic solutions must be applied.

4.2. Ring Generation Algorithm: A Heuristic Algorithm

We propose a heuristic algorithm to find a minimum set of rings that cover all 2-pin nets under test. The algorithm is a modified depth-first search which works as follows:

The SOC under test is first modeled as a hypergraph G'. This graph is then transformed into graph $G=(V, E)$ with 2-pin nets only. The vertex set V consists of pins in all cores. The edge set E is partitioned into two disjoint subsets E_i and E_e, where E_i is the set of internal edges

(i.e. those edges in the scan paths within modules/IP cores) and E_e is the set of external interconnect wires (i.e. interconnects). Our goal is to generate rings that cover E_e.

We generate a ring containing a 2-pin net $(u, v) \in E_e$ by starting from vertex v, which is an input pin. Then we try to find an output pin w that locates in the same core as v, and w is connected to a 2-pin net that is not yet covered by any other ring. If no such unvisited 2-pin net (w, x) exists, we just select the first available output net from any available set of output pins. This process is repeated until a ring is found. The procedure then goes over again and again until all 2-pin nets are covered.

The above heuristic works as follows: Whenever we start looking for a new ring, we explore paths containing 2-pin nets that are not yet covered. In this way, each new ring may cover as many other uncovered nets as possible. After all rings having been generated, a simple reverse order simulation is conducted to remove redundant rings. A net is *oscillation ring testable* if there exists at least one ring covering this net. The algorithm is outlined below.

```
Algorithm:  Ring Construction
Input:      A hypergraph G'=(V, L) representing a circuit
Output:     A list of rings R
1.   Transform G' to a new graph G=(V, E) with hypernets into
     equivalent 2-pin nets only, in which all nets are oscillation
     ring testable;
2.   R=∅;
3.   for every e=(u, v) ∈ E and e is not visited
4.       R=R U∪ find_ring(G, e);
5.   reverse-order simulation for rings in R, end program.
     function find_ring(G, e)
1.   Let e=(u, v) and v is an input pin in core C;
2.   if v is a pin in the starting core
3.       return the ring and mark all nets as visited;
4.   for every output pin w in C
5.       if there is an unvisited edge (w, x)
6.         find_ring(G, (w,x));
7.       else if there is an untried output net (w, x)
8.         find_ring(G, (w,x));
9.       else
10.        return ∅;
11.  end function
```

This algorithm can take into account the variation of delay on each individual wire. Long rings with very large variance in path delay will not be constructed since delay faults may be masked in these rings. The extra restriction increases the complexity of the ring searching algorithm, but it reduces the probability of error masking caused by process variation.

5. Experimental Results

We implement the proposed algorithm, and evaluate its performance with some MCNC benchmark circuits. The results are presented in this section.

5.1. An Illustrative Example

To validate the proposed OR test methodology, an MCNC benchmark "hp" consisting of 11 cores connected by 195 interconnects, was placed and routed as shown in Fig. 11 and then implemented using the TSMC 0.18 μm technology to simulate its oscillation condition.

The simulation results are shown in Fig. 12, where Fig. 12a shows the oscillation signal of the longest ring; and Fig. 12b shows the oscillation signal of the shortest rings. The cycle time of the longest rings (with nine interconnects) is about 38 ns and that of the shortest rings is about 2.8 ns. Thus, the oscillating frequency ranges from 26 to 357 MHz, and this shows that this oscillation detection scheme is feasible.

5.2. Process Variation Effects on Oscillation Signals

In order to consider process variation effect on this proposed OR scheme, we conducted experiment for a ring consisting of seven inversions and 20 μm lines. The Monte Carlo simulation is conducted by changing the W/L ratio of all transistors and the R, C parameters of the nets. The mean is the nominal value, while the distribution is Gaussian with 3σ=20% of the nominal value. In all, 400 simulation runs are performed, and the simulation results are shown in Fig. 13, in which all oscillation signals start at time 0. At the end of the first cycle, there is a small variation in the cycle period, and the variations are less than 1.2% of the nominal cycle period of the oscillation signal. The simulation results show that (1) this scheme can oscillate with an odd number of inversions, and (2) the process variation effects with 20% variance contribute to less than 1.2% in the frequency and oscillation period.

5.3. MCNC Benchmarks

We experiment the ring-generating algorithm with six MCNC benchmark circuits [7, 8]. The circuit statistics and results are all shown in Table 3, where the first column gives the circuit names, while the next four columns are circuit statistics, including number of cores (#core), number of pads (#pads), number of hypernets (#hypernets), and number of 2-pin nets (#2-pin), respectively. The next two columns are experimental results. The sixth column gives

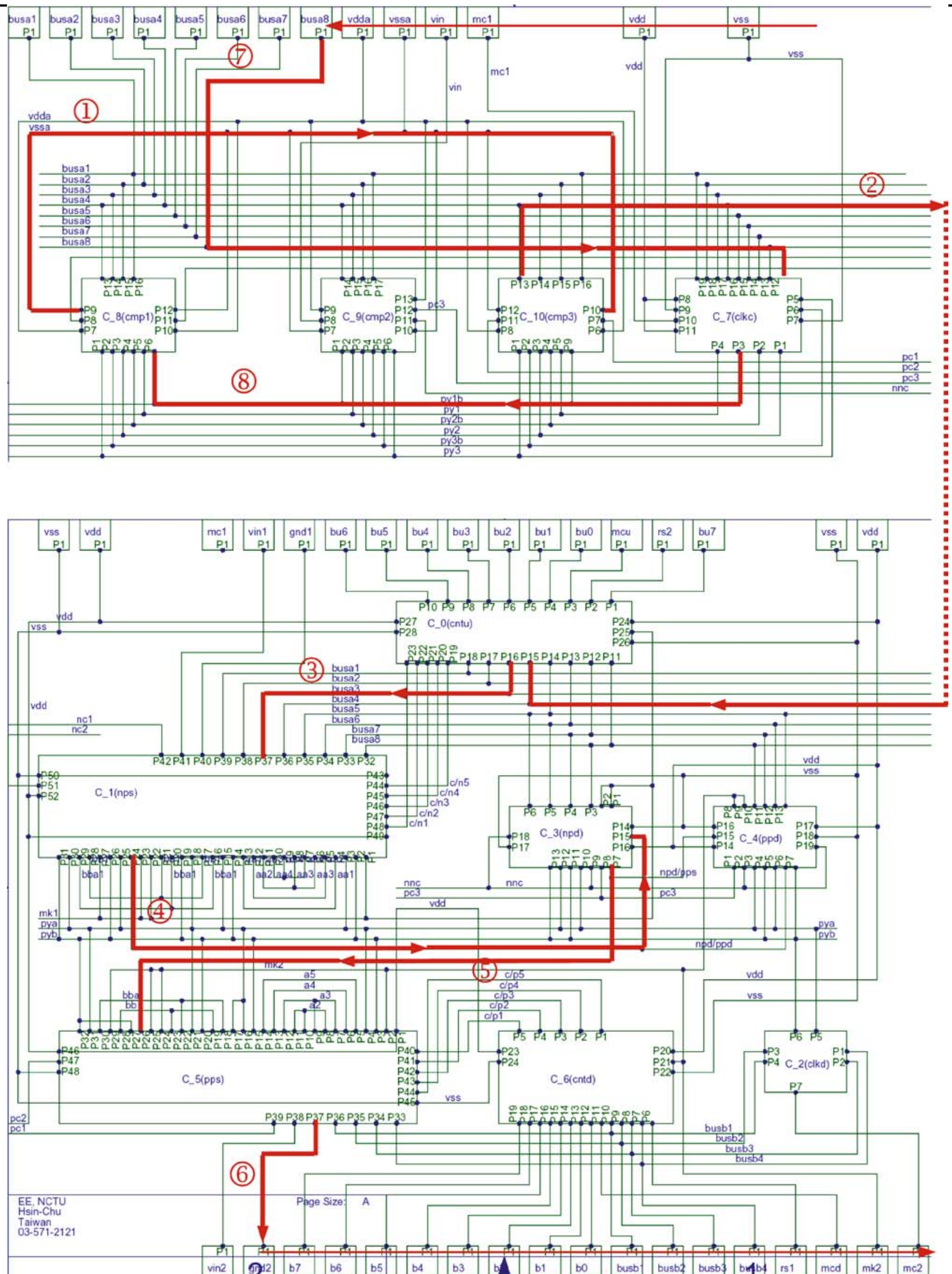

Fig. 11. The placement and routing of an illustrative example of the OR testing for a benchmark circuit *hp*.

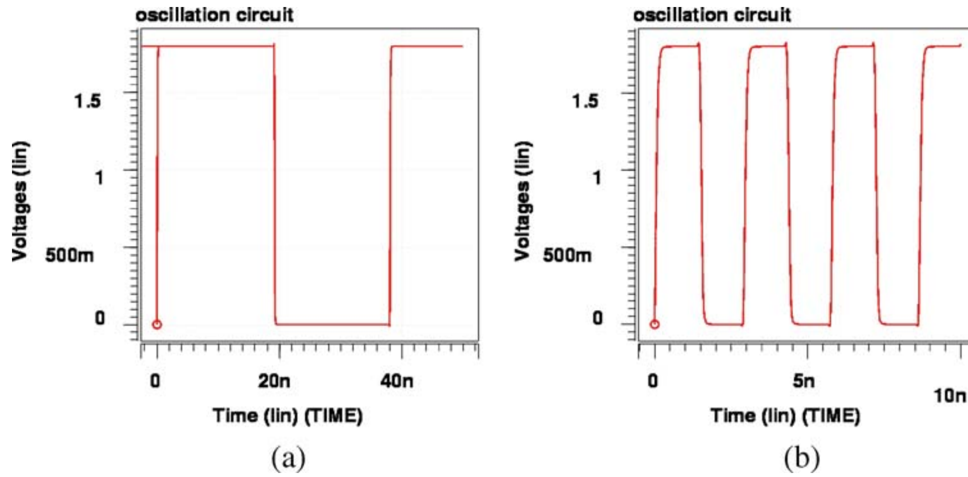

Fig. 12. Simulated waveforms of the longest **a** and shortest rings **b** of benchmark circuit *hp*.

the number of rings formed to cover all 2-pin nets (#rings) for complete detection of 100% fault coverage. In column 7, we give the estimated testing time (given in msec), which is obtained by assuming a 4 MHz measuring period. The time needed to set up the rings should be roughly proportional to the testing time. The last column (L.B.) gives the *lower bound* on the number of rings required for complete detection, calculated according to Corollary 1, for each circuit. From the last two columns, it can be seen that the lower bound is not very tight for some cases. If we assume that the ratio for the test setup time and the test application time is 1:1, then the longest time required among all MCNC benchmarks to finish the OR test is around 100 ms, which is for the circuit ***xerox***.

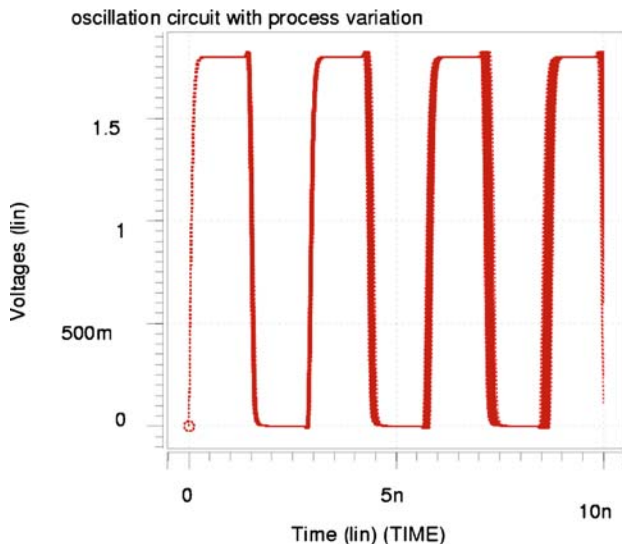

Fig. 13. Simulation waveform with process variation effects on the oscillation ring test scheme.

Since in these benchmark circuits the net directions are not known, we assume that: (1) Cores are listed in an order. For a hypernet formed, the pin corresponding to a core first in the core list is assumed to be the source while others are sinks. (2) Since the order on internal scan paths is not known, all output pins are conservatively assumed to be placed in consecutive positions. (3) All pads are connected through the boundary scan path, while positions of the pads are unknown. The above assumptions represent the worst-case scenario. (4) The inversion control signal *inv* is used to control the number of inversions in a ring. Thus, a ring consists of even of odd number of inverters is acceptable. Under assumption (2), none of the output 2-pin nets of a core can be tested in a single ring, and thus each ring may pass any core only once. The assumption (3) makes the boundary scan path appear only once in each ring. Thus, the results obtained could be treated as the *upper bound* on the rings required; the actual number of rings required would be smaller.

If test rings form a partition of all the 2-pin nets, a 2-pin net will appear in only one ring. In this case, the average ring length is minimum: (#2-pin net)/(#ring), and this number is also given in Table 4. However, such a partition is usually not possible, and most 2-pin nets

Table 3. Experimental results for MCNC benchmarks.

Circuit	#cores	#pads	#hypernet	#2-pin	#ring	Time (ms)	L.B.
ac3	27	75	211	416	133	33.3	69
ami33	33	42	117	343	242	60.5	214
ami49	49	22	361	475	154	38.5	35
apte	9	73	92	136	73	18.3	38
hp	11	45	72	195	82	20.5	68
xerox	10	2	161	356	218	54.5	174

Table 4. Analysis of ring lengths.

Circuit	#2-pin	#ring	Average ring length	#2-pin/#ring
ac3	416	133	6.99	3.13
ami33	343	242	8.93	1.33
ami49	475	154	16.54	3.08
apte	136	73	3.75	1.86
hp	195	82	4.7	2.38
xerox	356	218	3.73	1.63

appear in many different rings in circuits. Thus, the average ring length in our experiments is larger than (#2-pin net)/(#ring), which is shown in column 4 and column 5 in Table 4.

In Fig. 14, the distributions of ring lengths for each simulated benchmark are also shown. These distributions are quite different because of the different circuit structure of interconnects for each circuit.

The relationship between the number of OR testing rings and the achieved interconnect coverage, which consequently reflects the stuck-at fault, open fault, and delay fault coverage, is shown in Fig. 15 for all the simulated circuits. As to crosstalk fault, it can be observed by scanning out the contents of all the local counters to check whether it exist any crosstalk faults between the target nets in the oscillation ring and all the adjacent victim nets. It can be seen that the fault coverage in Fig. 15 increases roughly *linearly* with the number of test rings applied. This is in contrast with logic testing, in which a small number of test vectors usually account for the detection of most of the faults. There are some observations for the OR test methodology. First, the difficulty for the fault detection is almost the same for all interconnects in one circuit since the relationship between fault coverage and number of rings is approximately a straight line. Second, the difficulties which are in detecting faults are approximately proportional to the circuit size and interconnect structure, and thus determine the number of rings.

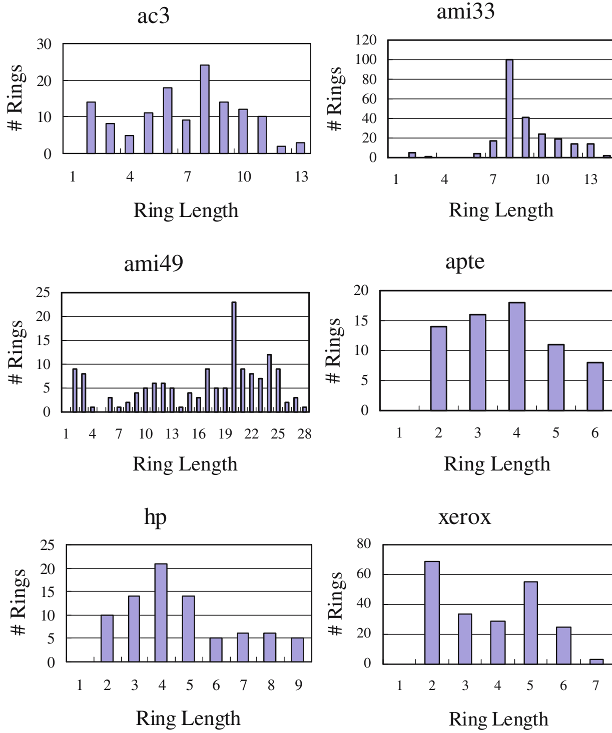

Fig. 14. Distribution of the ring lengths for the benchmark circuits by applying OR testing.

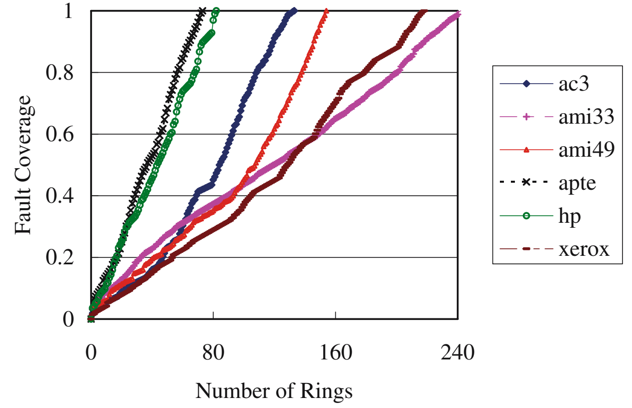

Fig. 15. Fault coverage versus number of rings.

If assumption (2) given in this section is lifted, then a core can be passed by a ring several times, or a core can be passed by several rings concurrently. As a result, the number of required rings can be reduced. A study on pin order and concurrent test can be found in Ref. [30].

5.4. Hardware Overhead

The hardware overhead of the oscillation test scheme can be estimated as follows. Let the circuit size of a two-input NAND be an equivalent gate. In order to implement the proposed method, each wrapper cell must be enhanced to provide extra paths, as shown in Fig. 5. Besides, each core must be provided with an embedded counter. For an enhanced input wrapper cell, the area penalty is roughly 3.5 equivalent gates, which include two 2-to-1 multiplexers, one inverter, and a pulse detector (not shown in Fig. 5a). For an enhanced output wrapper cell, the size of extra hardware is 4 equivalent gates, which include two 2-

Table 5. Hardware overhead.

Type	Enhanced wrapper cell		Embedded counter
	Input	Output	
Area (in Equivalent gate counts)	3.5	4	$\left\lceil \log_2 \left(\frac{f_{\max}}{f_{\min}} \times \frac{1}{\zeta} \right) \right\rceil \times 5.5$

An equivalent gate stands for a two-input NAND gate.

to-1 multiplexers, one buffer, and an inverting tri-state buffer.

A simple counter is constructed by cascading a number of T flip–flops (TFFs). There are many possible ways to design a TFF. For example, the design shown in [31] uses 5.5 equivalent gates to construct a resetable TFF. The length of the embedded counters is decided by the largest counter content $n_{\max}$, since the counters must be large enough to accommodate $n_{\max}$. Thus, the size of the counter should be at least $\lceil \log_2 n_{\max} \rceil$. Since $n_{\max} = f_{\max} \times T_0$, by rewriting Eq. 2, we have

$$n_{\max} = f_{\max} \times T_0 \geq \frac{f_{\max}}{f_{\min}} \times \frac{1}{\zeta} \quad (5)$$

From Eq. 5, we know that the length of the counters should be at least $\left\lceil \log_2 \left(\frac{f_{\max}}{f_{\min}} \times \frac{1}{\zeta} \right) \right\rceil$.

In our experiment, we assume $f_{\max}$=400MHz, $f_{\min}$= 4MHz, and ζ=0.001. Thus, the counter length should be 17, and the size of a counter is 93.5 equivalent gates. The area penalty is summarized in Table 5.

Since only one local counter is needed in each core, the overhead due to the counter is relatively small if the core is large enough. The extra hardware provides a mechanism for delay and crosstalk fault detection. Besides, it can be used for on-chip measurement [28]. In contrast, the delay and crosstalk issues are not considered in the IEEE Std. 1500. To detect timing related faults, it is necessary to launch signal transitions as well as to capture result at specific time. To achieve these requirements, the wrapper cells also have to be enhanced and more complex timing control has to be designed. Therefore, the oscillation test scheme actually provides a cost-effective way for delay and crosstalk fault detection.

6. Conclusion

We have presented a novel *o*scillation *r*ing (OR) test methodology as an effective DFT technology for interconnects among intellectual property (IP) modules in an SOC under the IEEE Std. 1500 environment. The target interconnect fault models include stuck-at, open, delay faults, and crosstalk glitches, which are difficult to test under traditional test architectures. Simulation on an MCNC benchmark circuit implemented with the TSMC 0.18 μm technology has shown the feasibility of the methodology. The path delay of the interconnection wires can also obtained by measuring period of the oscillation signals, thus the proposed scheme can be used for the circuit parameter measurement. The IEEE Std. 1500 wrapper cells and boundary scan wrapper cells are modified to accommodate the test scheme. We adopted a graph model approach to propose an efficient ring-construction algorithm to construct to achieve 100% interconnect coverage, consequently fault coverage. The OR scheme as an oscillation DFT gives high testability, and can be viewed as a variance of BIST with the IEEE Std. 1500 design. The scheme can also be extended to diagnosis the interconnect faults since the fault on an edge common to two or more rings will fail all oscillation signals passing through the shared common edge(s). This will be the topic of the followed study for this scheme.

Acknowledgements

The work of Katherine Shu-Min Li was supported in part by National Science Council of Taiwan under Grant No. NSC 95-2218-E-110-009. The work of Chauchin Su was partially supported by National Science Council of Taiwan under Grant Nos. NSC 95-2221-E-009-328 MY3 and NSC 95-2221-E-009-334.

References

1. Semiconductor Industry Association (SIA), *International Technology Roadmap for Semiconductors (ITRS)*, 2001.
2. W. Chen, S.K. Gupta, and M.A. Breuer, “Analytic Model for Crosstalk Delay and Pulse Analysis under Non-Ideal Inputs,” *Proc. Int'l Test Conf.*, 1997, pp. 809–818.
3. S. Irajpour, S. Nazarian, L. Wang, S.K. Gupta, and M.A. Breuer, “Analyzing Crosstalk in the Presence of Weak Bridge Defects,” *Proc. VLSI Test Symp.*, 2003, pp. 385–392.
4. K.-T. Cheng, S. Dey, M. Rodgers, and K. Roy, “Test Challenges for Deep Sub-Micron Technologies,” *Proc. Design Automation Conf.*, 2000, pp. 142–149.
5. K.T. Lee, C. Nordquist, and J.A. Abraham, “Automatic Test Pattern Generation for Crosstalk in Digital Circuits,” *Proc. VLSI Test Symp.*, 1998, pp. 34–39.
6. P.B. Sabet and F. Ilponse, “A Model for Crosstalk Noise Evaluation in Deep Submicron Processes,” *Proc. Int'l Symp. on Quality Electronic Design*, 2001, pp. 139–144.
7. J. Cong, T. Kong, and Z.D. Pan, “Buffer Block Planning for Interconnect-Driven Floorplanning,” *Proc. Int'l Conf. Computer-aided Design*, 1999, pp. 358–363.
8. J. Cong, D.Z. Pan, and P.V. Srinivas, “Improved Crosstalk Modeling for Noise Constrained interconnect Optimization,” *Proc. Asia South Pacific Design Automation Conf.*, 2001, pp. 373–378.
9. K.S.-M. Li, Y.-H. Cherng, and Y.-W. Chang, “Noise-Aware Buffer Planning for Interconnect-Driven Floorplanning,” *Proc. Asia South Pacific Design Automation Conf.*, Jan. 2003, pp. 423–426.
10. J.A. Sainz, M. Roca, R. Munoz, J.A. Maiz, and L.A. Aguado, “A Crosstalk Sensor Implementation for Measuring Interferences in Digital CMOS VLSI Circuits,” *Proc. On-Line Testing Workshop*, 2000, pp. 45–51.
11. F. Caignet, S.D.-B. Dhia, and E. Sicard, “On the Measurement of Crosstalk in Integrated Circuits,” *IEEE Trans. VLSI Systems*, vol. 8, no. 5, pp. 606–609, Oct. 2000.
12. S. Bose, P. Agrawal, and V.D. Agrawal, “A Rated-clock Test Method for Path Delay Faults,” *IEEE Trans. on VLSI Systems*, vol. 6, no. 2, pp. 323–331, 1998.
13. K.-T. Cheng and H.-C. Chen, “Classification and Identification of Nonrobust Untestable Path Delay Faults,” *IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems*, vol. 15, no. 8, pp. 845–853, 1996.

14. H.B. Bakoglu, *Circuit, Interconnections, and Packaging for VLSI*, Reading, MA: Addison-Wesley, 1990.
15. W.K. Kautz, "Testing of Faults in Wiring Interconnects," *IEEE Trans. Computers*, vol. C-23, no. 4, pp. 358–363, Apr. 1974.
16. P. Goel and M.T. McMahon, "Electronic Chip in Place Test," *Proc. Int'l Test Conf.*, 1982, pp. 83–90.
17. P.T. Wagner, "Interconnect Testing with Boundary Scan," *Proc. Int'l Test Conf.*, 1987, pp. 52–57.
18. C.W. Yau and N. Jarwala, "A Unified Theory for Designing Optimal Test Generation and Diagnosis Algorithms for Board Interconnects," *Proc. Int'l Test Conf.*, 1989, pp. 71–77.
19. F. DaSilva, Y. Zorian, L. Whetsel, K. Arabi, and R. Kapur, "Overview of the IEEE P1500 Standard," *Proc. Int'l Test Conf.*, 2003, pp. 988–997.
20. E.J. Marinissen, R. Kapur, M. Lousberg, T. McLaurin, M. Ricchetti, and Y. Zorian, "On IEEE P1500 Standard for Embedded Core Test," *Journal of Electronic Testing (JETTA)*, vol. 18, no. 4–5, pp. 365–383, Aug. 2002.
21. M. Kaneko and K. Sakaguchi, "Oscillation Fault Diagnosis for Analog Circuits Based on Boundary Search with Perturbation Model," *Proc. Int'l Symp. Circuits and Systems*, 1994, pp. 93–96.
22. K. Arabi and B. Kaminska, "Oscillation-Based Test Strategy for Analog and Mixed-Signal Integrated Circuits," *Proc. VLSI Test Symp.*, 1996, pp. 476–482.
23. Z.M. Santo, F. Novak, and S. Macek, "Design of Oscillation-Based Test Structures of Active RC Filters," *IEE Proc. Circ. Devices Syst.*, vol. 147, no. 5, pp. 295–302, 2000.
24. G. Huertas, D. Vazquez, E.J. Peralias, A. Rueda, and J.L. Huertas, "Practical Oscillation-Based Test of Integrated Filters," *IEEE Design & Test Computers*, vol. 19, no. 6, pp. 64–72, 2002.
25. W.C. Wu, C.L. Lee, M.S. Wu, J.E. Chen, and M. Abadir, "Oscillation Ring Delay Test for High Performance Microprocessor," *J. Electron. Test. (JETTA)*, vol. 16, no. 1–2, pp. 147–155, 2000.
26. M.-S. Wu, C.L. Lee, C.P. Chang, and J.E. Chen, "A Testing Scheme for Crosstalk Faults Based on the Oscillation Test Signal," *Proc. Asian Test Symp.*, 2002, pp. 170–175.
27. C. Su and W. Tseng, "Configuration Free SoC Interconnect BIST Methodology," *Proc. Int'l Test Conf.*, Nov. 2001, pp. 1033–1038.
28. C. Su, Y.-T. Chen, M.-J. Huang, G.-N. Chen, and C.-L. Lee, "All Digital Built-In Delay and Crosstalk Measurement for On-Chip Buses," *Proc. Design, Automation, and Test in Europe*, Mar. 2000, pp. 527–531.
29. K.S.-M. Li, C.-L. Lee, C. Su, and J.E. Chen, "A Unified Detection Scheme and its Fault Effects of Interconnection Bus Crosstalk Faults in Deep Submicron," *Proc. Asia Test Symp.*, Nov. 2004, pp. 145–150.
30. K.S.-M. Li, C. Su, Y.-W. Chang, C.-L. Lee, and J.E. Chen, "IEEE Standard 1500 Compatible Interconnect Diagnosis for Delay and Crosstalk Faults," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, (in press).
31. N.H.E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, 2nd Ed. Reading, MA: Addison-Wesley, 1992.

Katherine Shu-Min Li received the B.S. degree from Rutgers University, New Brunswick, NJ, and the M.S. and Ph.D. degrees from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 2001 and 2006, respectively. She is currently an Assistant Professor of the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, R.O.C. Her research interests focus on crosstalk effects, signal integrity, systems-on-chip (SOC) testing, floor planning and routing, testability and yield enhancement, transition fault, scan reordering, scan routing, low-power scan techniques, especially on oscillation ring (OR) test scheme, and interconnect optimization in deep submicrometer and nanotechnology. She is a member of IEEE, ACM, and ACM/SIGDA.

Chung Len Lee received the B.S. degree in Electrical Engineering from National Taiwan University, Taiwan, Republic of China in 1968, and the M.S. and Ph.D. degrees in Electrical Engineering from Carneigie-Mellon University, Pittsburgh, PA, U.S.A. in 1971 and 1975, respectively. In 1975, he joined National Chiao Tung University as a faculty member, engaging in research on various topics on optoelectronics, semiconductor devices, integrated circuits, and processes. He has published more than 300 papers in the above areas. Dr. Lee is a member of Phi Kappa Phi and Phi Tau Phi.

Chauchin Su received the B.S. and M.S. degrees in Electrical Engineering 1979 and 1981, respectively from National Chiao Tung University, Taiwan and Ph.D. degree in Electrical Engineering from University of Wisconsin at Madison, Wisconsin, U.S.A., in 1990. Since then, he joined the Department of Electrical Engineering, National Central University, Taiwan. In 2000, he transferred to the Department of Electrical and Control Engineering, National Chiao Tung University, Taiwan, where he is currently a Professor. His current research interests focus on the design and test of high speed serial links and mixed signal circuits. He is actively involved in various academic activities. He has served as a TPC member for ICCAD, ASP-DAC, ATS, IMSTW, and etc. He has also served as the Technical Program Cochair for ATS 2000 and General Co-chair for ATS 2004. Presently, he also serves as the Executive Director for The National SoC Program-Taiwan.

Jwu E Chen received the B.S., M.S. and Ph.D. degrees in Electronics Engineering from National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., in 1984, 1986 and 1990, respectively. He was a Member of the Department of Electrical Engineering faculty at the Chung Hua University, Hsinchu, Taiwan, from 1990 to 2004. He is currently a Member of the Department of Electrical Engineering faculty at the National Central University, Jhungli, Taiwan. His research interests include multiple-valued logic, VLSI Testing, reliable computing, yield analysis, and Humanoid Robotics. He is a member of IEEE and the Computer Society.