IEEE Standard 1500 Compatible Interconnect Diagnosis for Delay and Crosstalk Faults

Katherine Shu-Min Li[1], Yao-Wen Chang[2], Chauchin Su[3], Chung-Len Lee[1], Jwu E Chen[4]

[1]Department of Electronics Engineering, National Chiao Tung University, Hsichu, Taiwan
[2]Department of Electrical Engineering & Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan
[3] Department of Department of Electronic Control, National Chiao Tung University, Hsichu, Taiwan
[4]Department of Electrical Engineering, National Central University, Chungli, Taiwan

Abstract – We propose an interconnect diagnosis scheme based on Oscillation Ring test methodology for SOC design with heterogeneous cores. The target fault models are delay faults and crosstalk glitches. We analyze the diagnosability of an interconnect structure and propose a fast diagnosability checking algorithm and an efficient diagnosis ring generation algorithm which achieves the *optimal diagnosability*. Two optimization techniques improve the efficiency and effectiveness of interconnect diagnosis. In all experiments, our method achieves 100% fault coverage and the optimal diagnosis resolution.

I. Introduction

Interconnect delays, rather than gate delays, dominate overall circuit performance in the nanometer era [1-2], especially for System-on-chip (SOC) ICs. *Interconnect diagnosis*, including the detection and location of faulty nets, plays a key role in enhancing circuit reliability and yield. It is not easy to directly apply those existing interconnect diagnosis techniques to SOC designs, and the diagnosis costs greatly increase for manufacturing and yield enhancement. Therefore, it is desired to develop an effective test scheme to reduce the costs of interconnect diagnosis. Interconnect diagnosis for various applications, such as printed circuit board (PCB) and multi-chip module (MCM) has been studied extensively in the literature [3-8]. However, their target fault models are mainly traditional stuck-at and bridging faults. The diagnosis algorithms include counting sequence, walking-0 and walking-1 sequence, maximal independent test set, and focus mainly on special interconnect structures, especially for bus-oriented or FPGA designs. On the other hand, in this paper we consider delay and crosstalk glitch faults, which are important in nanotechnology.

Oscillation ring based test is an efficient and effective method to detect faults in a circuit or a device [9-10]. An oscillation ring is a closed loop with an odd number of signal inversions. Once the ring is constructed, oscillation signal appears on the ring. For a circuit with faults, some rings will not oscillate correctly. Once a set of oscillation tests have been conducted, we can locate some or all of the faults according to the test outcome [11]. Whether each fault can be correctly identified, or *diagnosed*, depends on the interconnect structure and the test rings applied.

The advantage of oscillation ring based diagnosis for the interconnect structure is that, in addition to functional faults like stuck-at and open faults, it is also capable of identifying delay faults and crosstalk glitch faults, the main sources for the loss of signal integrity [2]. Therefore, the *oscillation ring* based technique is an ideal approach to interconnect diagnosis.

In this paper, we propose an oscillation ring based scheme to diagnose interconnect faults to reduce the test time for SOC interconnect diagnosis. This approach is compatible with the P1500 standard [12], providing structural support for core testing as well as interconnect testing in SOC. We analyze the diagnosability of an interconnect structure and propose a fast diagnosability checking algorithm and an efficient ring generation algorithm. We prove that the generation algorithm can find the optimal diagnosability for any interconnect structure. The predetermined diagnosis method achieves the optimal diagnosability (i.e. the maximum diagnosis resolution). We also propose two optimization techniques for test time reduction with no hardware overhead. The first one is an adaptive diagnosis method, which reduces test time by 1.54X-2.67X. The other is a concurrent diagnosis method, which improves test effectiveness by up to 9.66%. Experiments on the MCNC benchmark circuits show our methods achieve 100% fault coverage and the optimal diagnosis resolution. (Here, the diagnosis resolution is defined as the cardinality of the largest set of indistinguishable faults, and the maximum diagnosis resolution or the optimal diagnosability implies that the cardinality is 1.)

The proposed approach provides many advantages. First, it is applicable to arbitrary global interconnect. In contrast, previous diagnosis methods are more concentrated on special structures. Second, our approach can deal with faults that cause signal integrity problems, while it is difficult to handle such faults under traditional methods. We provide ring generation algorithms that achieve 100% fault coverage and the optimal diagnosis resolution for the modeled faults. A fast diagnosability checking methodoloty is given in this paper, which greatly reduce the execution time.

II. Oscillation Ring Test Scheme for Interconnect Detection and Diagnosis

A. The OR Test Architecture

We discuss the *interconnect oscillation ring test* (*IORT* for short) for SOC interconnects [11]. Figure 1 illustrates a counter-based test architecture for both delay and crosstalk glitch detection for SOC ICs with the compatible IEEE P1500 core test standard. In P1500, each input/output pin of a core is attached with a *wrapper cell*, and a centralized *test access mechanism* (*TAM*) is provided to coordinate all test processes. In addition to the normal input/output connections, all wrapper cells in a core can also be connected with a shift register, usually referred to as a scan path, to facilitate test access. A modified wrapper cell design has been proposed to provide extra connections and inversion control so that the oscillation rings can be constructed through the wires and the boundary scan paths in cores [11]. For example, the oscillation ring test architecture in Figure 1 consists of one oscillation ring and two neighboring nets.

The target fault models of this test architecture are stuck-at, open, delay and crosstalk glitch faults. In addition to fault detection, measuring the delay fault can also be achieved. If an oscillation ring fails to oscillate, there exists stuck-at or open fault(s) in the components of the oscillation ring. The period of the oscillation

0-7803-9451-8/06/$20.00

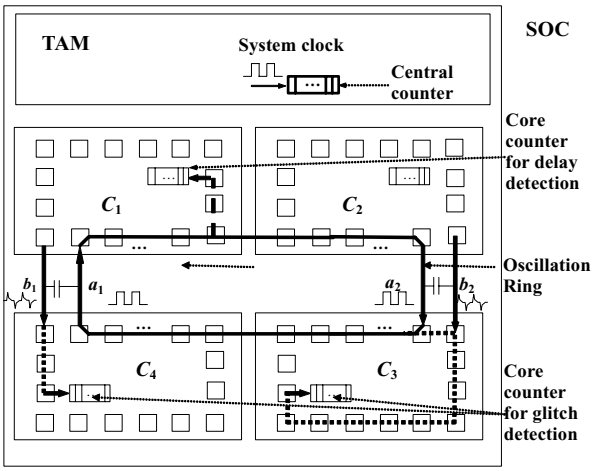


Figure 1. Test architecture for delay and crosstalk detection.

signal is measured by using a delay counter in a core to test delay faults, and a similar scheme is also applied for crosstalk glitch detection.

A local counter is included in each core, and a central counter is in the TAM of an SOC. The central counter in the TAM is enabled by the signal *OscTest*, and triggered by the system clock. A local counter is connected to one wrapper cell in each core; however, it can be accessed by every wrapper cell through the wrapper cell chain. When an oscillation ring passes a core, an internal scan path is formed to connect the oscillation signal to the local counter. For example, core C_1 is passed by the oscillation ring in Figure 1. The oscillation signal is fed to the local counter through a series of modified wrapper cells. When an oscillation test session starts (*OscTest* = 1), the TAM enables its own central counter as well as all local counters in cores. After the central counter in the TAM counts to a specific number n, the oscillation test session terminates and all local counters are disabled (*OscTest* = 0). Then all the local counter contents can then be scanned out to 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 i-th ring's oscillation frequency f_i can be approximated by

$$f_i = f \times n_i / n \qquad (1)$$

Since the frequency of each ring is predetermined during the design phase, a 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.

$$\varepsilon = \frac{1}{f_{\min} \times T_0} \le \zeta \qquad (2)$$

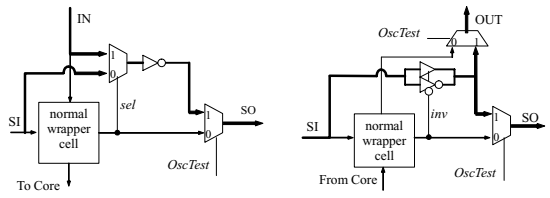
Let the frequency specification of the oscillation rings be 4 MHz to 400 MHz and ξ be 0.001, implying the counter content n_{min} is at least 1000. From (2), we have the required T_0 to be 250μs. This example illustrates the feasibility of the oscillation test scheme from a measurement prospect, and this frequency specification is actually compliant with ATE specifications.

B. P1500 Wrapper Cell Design

An oscillation ring consists of interconnect wires and part of the scan path in each core where the ring passes. Thus, a P1500 wrapper cell must provide necessary paths between input/output ports and scan in/scan out ports. If an oscillation test is used to test wires connected to pads, the boundary scan cells also have to be modified in a similar way. In order to facilitate the scheme, the P1500 boundary wrapper cells need to be modified.

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 internal core and the wire connected to the pin. Whenever an 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 the scan output (SO); while for an output pin, it should connect scan in (SI) to pin output (OUT) during an oscillation test session.

The modified wrapper cell designs are shown in Figure 2. 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 cell, the extra paths are SI→SO and SI→OUT. The added inverting and non-inverting buffers are used to generate oscillation signals for the OR test; however, in an input wrapper cell, only one type of buffer is provided due to the limited control signals. Two control signals are needed in each modified wrapper: signal *OscTest* is a global control signal; while the signal *sel* is only used in the input wrapper cell, and the signal *inv* is only used in the output wrapper cell to ensure the odd parity of each ring. Signals *sel* and *inv* are set individually and scanned into the wrapper cells before an oscillation ring test session starts.



(a) (b)

Figure 2. Modified wrapper cells with forced inversion; (a) input (b) output.

C. Interconnect Diagnosis for SOC

A circuit consisting of three cores (C_1, C_2, and C_3) and three nets (n_1, n_2, and n_3) is shown in Figure 3. The first ring consists of nets n_1 (and its right-hand side branch), n_2, and n_3, and it passes all three cores. The second ring consists of n_1 (and its left-hand side branch) and n_3, and scan paths in C_1 and C_3. The oscillation ring test scheme detects which line is faulty (n_1, n_2 or n_3), and the oscillation ring diagnosis scheme diagnoses which segment is faulty (n_{11}, n_{12}, n_{13}, n_2 or n_3).

In order to simplify the interconnect diagnosis problem, we model the SOC circuit in Figure 3 by a *hypergraph*, and model interconnects by a *hypernet* as shown in Figure 4.

Definition 1: A *hypergraph* $H = (V, L)$ consists of a vertex set V and an edge set L. A multi-terminal edge connects a set of vertices $V_i \subseteq V$, $| V_i | \ge 2$, and it is referred to as a *hypernet*.

This hypergraph model is not good enough for diagnosis, since different parts of the same net (i.e. different net segments) affect different rings. Consider the 5-terminal hypernet with seven edge segments e_1 to e_7 as shown in Figure 4(a). If edge e_1 is faulty, all four rings will not oscillate correctly. A faulty e_2 affects rings 1 and 2, a faulty e_3 affects rings 3 and 4, and faults on edges e_4, e_5, e_6 and

e_7 affect rings 1, 2, 3, and 4, respectively. For diagnosis purpose, all these seven segments are different.

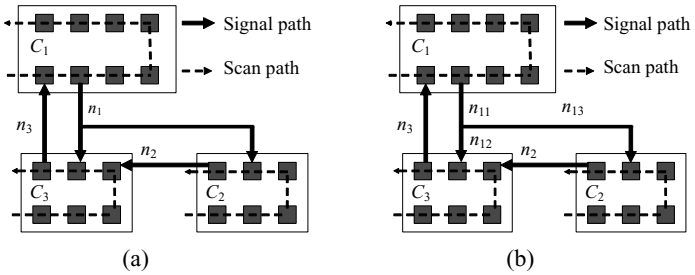


Figure 3. An example of SOC interconnect (a) interconnect detection for each net (b) interconnect diagnosis for each net segment.

Definition 2: A directed *graph* $G = (V, E)$ consists of a vertex set V and an edge set E, and each edge in E is an ordered pair (u, v), where $u, v \in V$.

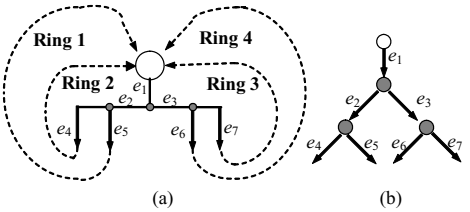


Figure 4. (a) a hypernet, and (b) the graph model for diagnosis.

III. Interconnect Diagnosability

A. Diagnosability Analysis

Given a circuit consisting of n edges $E = \{e_1, e_2, \ldots, e_n\}$ and a set of m oscillation rings $R = \{r_1, r_2, \ldots, r_m\}$. Once a ring is constructed, the test outcome is either "*pass*" (P) or "*fail*" (F). When an edge e_i is faulty, the test outcome of applying the m rings is said to be the *syndrome* of faulty e_i.

Definition 3: A fault on edge e_i and a fault on edge e_j are *distinguishable* under the test set R if the syndrome of faulty e_i and faulty e_j are different.

Definition 4: An edge is said to be *single-fault diagnosable* under the test set R if a faulty edge can be correctly identified, given that there is at most one fault in the interconnect structure.

Lemma 1: A fault on edge e_i and a fault on edge e_j are *distinguishable* under the test set $R \Leftrightarrow R_i \neq R_j$.

Proof: $\Leftarrow$ The fact $R_i \neq R_j$ implies that there exists a ring r such that either (1) $r \in R_i \wedge r \notin R_j$, or (2) $r \in R_j \wedge r \notin R_i$. Thus, the syndromes of faulty e_i and faulty e_j are different.

$\Rightarrow$ When $R_i = R_j$, both faulty e_i and faulty e_j fail the same set of rings, and thus they have the same syndrome. □

Theorem 2: Edge e_i is *single-fault diagnosable* $\Leftrightarrow R_i \neq R_j$ for all $1 \leq j \leq n$ and $j \neq i$.

The correctness of Theorem 2 follows the result of Lemma 1. It takes $\mathbf{O}(n^2m)$ time to verify Theorem 2, since each pair of edges have to be compared. In order to reduce the complexity for diagnosability check, the following theorems can be used.

Theorem 3: Edge e_i is single-fault diagnosable if $|E_i| = 1$.

Proof: Assume that edge e_i is not single-fault diagnosable. From Theorem 2, there must exist an edge e_j such that $j \neq i$ and $R_i = R_j$. Therefore, both e_i and e_j belong to E_i and thus $|E_i| > 1$.

Theorem 4: Let R_i' be any non-empty subset of R_i for an edge e_i, and $E_i' = \bigcap_{r \in R_i'} r$. Edge e_i is single-fault diagnosable $\Leftrightarrow \forall\, e_k \in E_i' - \{e_i\}$, e_i and e_k are distinguishable.

Proof: $\Leftarrow$ When at least one ring in R_i' oscillates correctly, e_i must be fault-free. On the other hand, when no rings in R_i' oscillate correctly, at least one edge in E_i' is faulty. Since all edges in $E_i' - \{e_i\}$ are distinguishable from e_i, we know whether e_i is faulty. Therefore, e_i is also single-fault diagnosable.

$\Rightarrow$ Assume that there is an $e_k \in E_i' - \{e_i\}$ and e_k is not distinguishable from e_i. When every ring in R_i' fails, it may be attributed to either e_k or e_i. Thus, e_i is not single-fault diagnosable. □

Theorem 4 shows that not all rings in R_i are necessary to diagnose e_i, and a subset R_i' is informative enough if and only if e_i is distinguishable.

Corollary 5: Let R_i' be any non-empty subset of R_i for an edge e_i, and $E_i' = \bigcap_{r \in R_i'} r$. If for each $e_k \in E_i' - \{e_i\}$, e_k is single-fault diagnosable, then edge e_i is also single-fault diagnosable.

An example for the above definitions, theorems and corollaries is shown in Figure 5. Let the edge under consideration be e_i, then $R_i = \{r_1, r_2, r_3, r_4\}$, and $E_i = \{e_i, e_j, e_k\}$. Since R_i' can be any non-empty subset of R_i, we may choose $R_i' = \{r_2, r_3\}$, and thus $E_i' = \{e_i, e_j, e_k\}$. It is not necessary to have both e_j and e_k diagnosable to make e_i diagnosable. For example, let faults on e_j and e_k be indistinguishable; if a fault on e_i is distinguishable with $\{e_j, e_k\}$, then e_i is diagnosable according to Theorem 4.

Note that the above analysis applies to all types of faults except crosstalk glitches since they can be located directly from the test results of each ring.

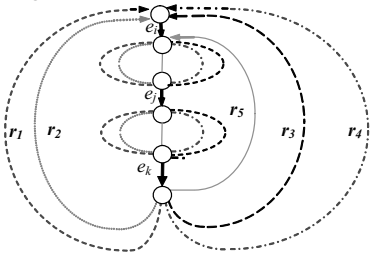


Figure 5. An interconnect diagnosis graph example.

B. Heuristic Diagnosability Check

In order to accelerate the process of diagnosability analysis, we propose a diagnosability check heuristic. Consider two edges e_i and e_j. According to Lemma 1, faults on these two edges are distinguishable if $|R_i| \neq |R_j|$. Thus, as the first step, we sort and partition all edges according to the number of rings passing them (i.e., $|R_i|$ for edge e_i). For example, in Figure 5, e_j and e_k are in the same group as $|R_j|=|R_k|=5$, distinguishable from $|R_i|=4$.

The second heuristic is to apply Theorem 3 first to check the diagnosability of an edge. Since the condition of Theorem 3, $|E_i|=1$, is only sufficient but not necessary to guarantee that e_i be single-fault diagnosable, it is still possible that e_i is single-fault diagnosable when $|E_i| \neq 1$. In this case, we need to compare R_i with R_j for each e_j in the same group as e_i. To avoid the aforementioned problem, a third heuristic is used. The most likely reason for diagnosable e_i with $|E_i| \neq 1$ is that there exists an e_j such that $R_j \supset R_i$. When the edge e_j has been checked and removed from the check list before edge e_i is processed, we shall not run into this problem by Corollary 5. The flowchart of the diagnosis checking heuristic is shown in Figure 6.

Finally, when two faults are indistinguishable, they are put into the same *equivalent class* so as not to be compared twice.

The interconnect diagnosis heuristic algorithm is illustrated as follows. Consider the graph shown in Figure 7. There are three rings in the figure: $r_1 = \{e_1, e_4\}$, $r_2 = \{e_2, e_5\}$, and $r_3 = \{e_1, e_2, e_3\}$.

The diagnosis matrix representation for Figure 7 is illustrated in Figure 8(a), where each column represents an edge and each row represents a ring. A "1" is put in cell (*i*,*j*) if ring *i* passes edge *j*. Note that the edges are sorted and partitioned into two groups that are separated by the broken line. The first group consists of edges e_1 and e_2, and each of them is passed by two rings (i.e., $|R_1|=|R_2|=2$). The second group consists of three edges, and each of them is passed by one ring only (i.e., $|R_3|=|R_4|=|R_5|=1$).

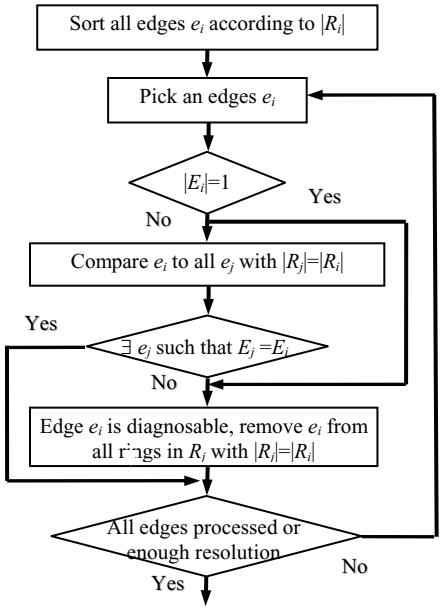


Figure 6. Flow chart of the heuristic for diagnosability checking.

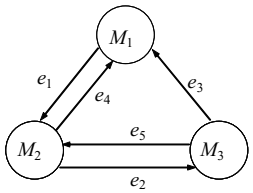


Figure 7. An illustrative diagnosability example.

The diagnosability checking process works as follows. First, apply Theorem 3 to edge e_1. We see that it is passed by rings r_1 and r_3, and the intersection of these two rings is $\{e_1\}$ (i.e., $|E_1|=1$). Thus, edge e_1 is single-fault diagnosable. Similarly, edge e_2 is also diagnosable as shown in Figure 8(b).

Syndrome of $e_1 = \{101\}$ indicates that the test results of r_1 and r_3 are incorrect and r_2 is correct when e_1 is faulty; syndrome of $e_2 = \{011\}$ indicates that r_2 and r_3 are incorrect and r_1 is correct when e_2 is faulty. Since the diagnosability analysis starts with the group with the highest $|R_i|$, we start with group $|R_i|=2$, including e_1 and e_2 and then group $|R_i|=1$, consisting of edges e_3, e_4, and e_5. Then, edges e_1 and e_2 are then marked and removed from the rings, as shown in Figure 8(b). There is only one edge remained in each ring, thus edges e_3, e_4 and e_5 are single-fault diagnosable due to Corollary 5.

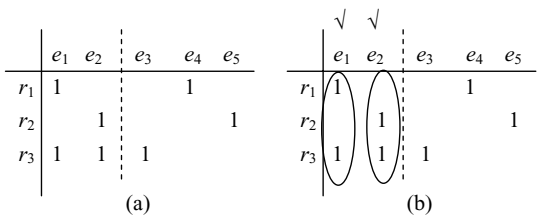


Figure 8. Matrices for the heuristic diagnosability checking.

IV. Interconnect Diagnosis Algorithm

In order to uniquely identify the faulty net segment, we need to ensure the optimal diagnosability or the maximum diagnosis resolution. The diagnosis resolution is defined as the largest number of nets with the same syndrome under a given set of test rings. Our goal is to diagnose every fault on every net segment, defined as the optimal diagnosability or the maximum diagnosis resolution.

We propose a heuristic to find a small set of rings for single fault diagnosis. The algorithm is a modified depth-first search. The SOC under test is modeled as a hypergraph *H*. This graph is then transformed into graph $G = (V, E)$ as outlined in Section 3.1. The vertex set *V* consists of cores and fanout points (intermediate nodes). The edge set *E* consists of edge segments partitioned from the original hypernets as explained in Figure 4(b). Our goal is to generate a predetermined set of rings to diagnose all edges in *E*. Since we need to detect the interconnect structure before diagnosis, the set of fault-detection test rings R_t should be applied first. A heuristic to find R_t is outlined below in Figure 9.

Algorithm: IORT (Interconnect Oscillation Ring Generation for Fault Detection)

Input: A hypergraph $H = (V, L)$ representing a circuit

Output: A list of rings R_t

```
1.  Transform hypergraph H into a new graph G = (V', E) with
    equivalent 2-pin nets;
2.  R_t; = ∅;
3.  for every e = (u, v) ∈ E and e is not visited
4.      R_t = R_t ∪ find_ring(G, e);
5.  reverse-order simulation for rings in R_t.

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
```

Figure 9. The ring generation for fault detection algorithm.

For interconnect detection in the IORT scheme, in order to find R_t, we propose a heuristic algorithm to find a minimum set of rings that cover all 2-pin nets under test. We generate a ring containing a 2-pin net $(u, v) \in E$ by starting from vertex *v*, an input pin. Then we 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. 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 containing this net.

Our goal for the interconnect diagnosis in the IORD scheme is to find a small set of rings R_d that can uniquely identify the faulty *edge* or *net segment*. The hypernet graph model for interconnect diagnosis (R_d) is the *2-pin net segment* model shown in Figure 4(b), different from the *2-pin net* model for interconnect detection (R_t). The set R_d is obtained by augmenting R_t as follows. We first apply the diagnosability checking techniques discussed in Section 3 to R_t to find out the net segments that are not diagnosable. For an edge *e* that is not single fault diagnosable, we try to find a new ring passing

it without going through the edges that are indistinguishable to e. If such a ring exists, it will be included in R_d. The diagnosability checking should be conducted for each added ring so that other edges that become diagnosable with the new ring will be found.

This algorithm can be adjusted to the required diagnosis resolution to reduce the number of diagnostic rings in Figure 10.

Algorithm: IORD (Interconnect Oscillation Ring Generation for Fault Diagnosis)

Input: A hypergraph $H = (V, L)$ representing a circuit
Output: A set of rings R_d

1. Transform hypergraph H into a new graph $G = (V'', E)$ with equivalent *2-pin net segments*;
2. Generate a set of rings R_t for fault detection;
3. $R_d = R_t$;
4. Conduct diagnosability check;
5. **for every** $e \in E$ {
6. **if** (e is not single-fault diagnosable)
7. Find a ring r to make e diagnosable;
8. $R_d = R_d \cup \{r\}$;
9. Modify the diagnosability of all edges in E;
   }
10. return R_d;

Figure 10. The ring generation for fault diagnosis algorithm.

The flowchart illustrating the process of diagnosis ring generation is given in Figure 11.

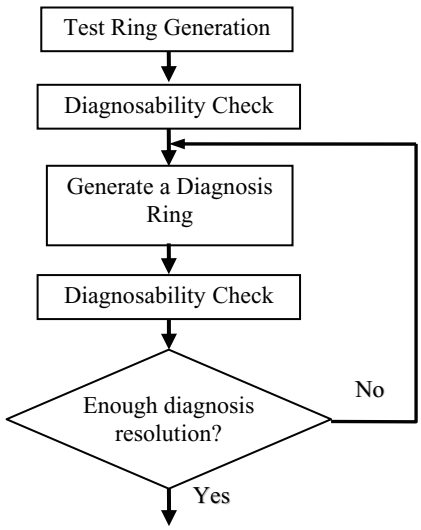


Figure 11. Diagnosis ring generation procedure.

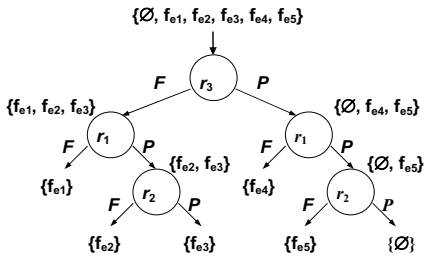


Figure 12. An adaptive diagnosis tree.

V. Optimization Techniques for Interconnect Diagnosis

Multiple oscillation rings cannot be applied simultaneously if they share some net segment (common edge constraint), or they go through the same scan path in a core (scan path conflict). In order to achieve the maximum concurrency (i.e., parallel test), we model all the constraints by a *conflict graph*, in which each ring is represented by a node, and two nodes are connected by an edge if they interfere with each other. The problem of finding the maximum concurrency tests can thus be reduced to the well-known *graph coloring problem*.

The number of test patterns can be greatly reduced whenever adaptive diagnosis is possible. In the adaptive diagnosis, a test pattern is selected according to the result of previous tests. An adaptive diagnosis tree, typically a binary tree, can be constructed according to the test patterns. For example, the adaptive diagnosis tree for the diagnosis example given in Figures 7 and 8 is illustrated in Figure 12.

For an n-net system, initially there are n+1 possible diagnosis results, namely fault-free (∅) and a single fault on net e_i (f_{ei}) for $1 \le i \le n$. Each node in the tree represents a test pattern (ring), and the test outcome can be either pass (P) or fail (F). If the tree is balanced, the minimum number of diagnosis patterns required is $\lceil \log_2(n+1) \rceil$.

In order to construct a balanced adaptive diagnosis tree, in each internal tree node we need to select the test pattern (i.e. test ring) that evenly partitions the possible outcomes into two groups: Fail (F) and Pass (P). For example, in Figure 15, we choose the test pattern r_3 as the first test, since it evenly partitions the six possible outcomes into Fail (f_{e1}, f_{e2}, f_{e3}) and Pass (∅, f_{e4}, f_{e5})**.** It can be seen that, in Figure 15, each test partitions possible outcomes into two groups whose cardinalities differ by at most 1.

The upper bound on the number of adaptive diagnosis test sessions needed in our method can be computed as follows. Let the number of test rings (without diagnosis) be $|R_t|$, and the length of the longest test ring be L_h. In the worst case, we need to apply $|R_t|$ rings to find out that there is a faulty net, and the last ring contains L_h net segments that are all passed by the ring only. It takes up to L_h–1 rings to distinguish these L_h possible faults, and thus the maximum number of diagnosis rings is $|R_t|$+ L_h–1.

VI. Experimental Results

We tested the diagnosis algorithm based on six benchmark circuits. In Table I, where the first column gives the circuit names, and the next four columns give the circuit statistics ("*Statistics*"), including the number of cores (*#core*), the number of pads (*#pads*), the number of hypernets (*#hyp*), and the number of net segments (*#net_segment*). The 5th column, *#net_segment*, lists the number of net segments to be diagnosed in each benchmark. The next three columns ("*Predetermined*") give the experimental results for predetermined diagnosis, including the number of rings required to detect all *2-pin nets* ($|R_t|$) and to diagnose all single faults ($|R_d|$). The last column, $|R_d|/|R_t|$, gives the ratio of rings from 1.25X to 2.81X for the maximum diagnosis resolution vs. for fault detection. This ratio means that we need extra test time of 1.25X to 2.81X to diagnose the single fault in each net segment under the predetermined diagnosis method, compared to the IORT scheme. In each case, we also give the estimated testing time (given in parenthesis), obtained by assuming only 4 MHz measuring period as discussed in Section 2.1 to estimate the *longest* test application time for each ring. The time needed to set up the rings should be roughly proportional to the testing time.

The next four columns ("*analysis*") give the diagnosis related information after applying R_t rings. The column *#OneRing* gives the number of nets passed by only one ring. Since the purpose of R_t is to detect faults with the minimum number of rings, it is not surprising that most nets are passed by one ring only. Most nets that are not diagnosable at this stage fall into this set. Columns "*#NoDiag*" and "*#EquClass*" give the number of nets that are *not* *diagnosable* and the number of *equivalence* *classes* after applying R_t, respectively. Two faults are in the same equivalence class if their syndromes for the tests are identical. The last column in this group ("$|R_d|$–$|R_t|$") gives the number of extra diagnosis rings required in each case to

make all nets single-fault diagnosable. Assume that there are m equivalence classes whose sizes are s_1, s_2, …, s_m, respectively. The upper bound on the number of additional diagnosis rings "$|R_d|$–$|R_t|$" can be expressed as follows:

$$\sum_{i=1}^{m}(S_i - 1) = \sum_{i=1}^{m} S_i - m = \#\,NoDiag - \#\,EquClass \qquad (3)$$

The upper bound on the required number of extra rings ($|R_d|$–$|R_t|$) is "(*#NoDiag*)–(*#EquClass*)". The empirical results "$|R_d|$–$|R_t|$" differs from the theoretical results "(#NoDiag)–(#EquClass)" given in Equation (3) by small differences of only up to 6.64%.

The last three columns ("*adaptive*") compare the number of rings required in both predetermined ($|R_d|$) and adaptive diagnosis ($|R_a|$). After applying R_t rings, the size of the largest equivalence class for each benchmark is given in the column "*max. EC*". In the worst case, the adaptive diagnosis needs to apply $|R_t|$ rings, and then *(max. EC)–1* rings for diagnosis. The number of the worst-case adaptive diagnosis rings is given in column "$|R_a|$". The last column ($|R_d|/|R_a|$) shows the ratio of rings for the predetermined vs. adaptive diagnosis schemes. For the results shown in the column, the adaptive algorithm obtains 1.23X to 2.67X improvements over the predetermined diagnosis scheme. Also, from the normalized $|R_a|$ and $|R_t|$, the test time of adaptive diagnosis is approximately equal to that for detection alone and this reveals the effectiveness of adaptive diagnosis.

The experimental results for the concurrent test are given in Table II. The 3rd column ($|R_c|$) lists the number of test sessions after applying the concurrency test under the assumed worst-case scenario of net directions, core lists, scan paths and boundary scan paths. When a set of rings are applied concurrently, we refer to these rings as a test session. The 4th column ($|R_d|$-$|R_c|$) gives the percentage of improvements. The improvement can be even better for general interconnect structures. The reduction in test time due to the concurrent test ranges from 0.27% to 9.66% *with no hardware overhead*.

VII. Concluding Remarks

We have presented an IORD scheme for interconnect faults in SOC. In addition to the *100% fault detection coverage for each net* achieved by the IORT scheme, we have shown that fault location or fault diagnosis can also be done by including some extra test rings to achieve the *optimal diagnosability (or the maximal diagnosis resolution) for each net segment*. We have also presented two heuristics, diagnosability check and diagnosis ring generation, with theoretical study and integrated them into the IORD algorithm. Finally, two optimization techniques for improving interconnect diagnosability are proposed and showed to be effective. We have further compared the predetermined, adaptive and concurrent diagnosis schemes. Experimental results have justified the efficiency and effectiveness of the proposed methods.

References

[1] M. Tehranipour, N. Ahmed, M. Nourani, "Testing SoC Interconnects for signal integrity using boundary scan", in *Proc VTS*, 2003.

[2] Semiconductor Industry Association (SIA), *International Technology Roadmap for Semiconductors 2003 Edition* (ITRS), 2003.

[3] W. K. Kautz, "Testing of faults in wiring interconnects," *IEEE Trans. Computers*, vol. C-23, no. 4, pp. 358-363, Apr. 1974.

[4] X.-T. Chen, F. J. Meyer, and F. Lombardi, "Structural diagnosis of interconnects by coloring," *ACM Trans. Design Automation Electronic Systems*, vol. 3, no. 2, pp. 249-271, Apr. 1998.

[5] Y. Kim, H.-D. Kim, and S. Kang, "A new maximal diagnosis algorithm for interconnect test," *IEEE Trans. VLSI*, vol. 12, no. 5, pp. 532-537, May 2004.

[6] J.-C. Lien and M. A. Breuer, "Maximal diagnosis for wiring networks," in *Proc. ITC*, pp. 71-77, 1991.

[7] W.-T. Chen, J.-L. Lewandowski, and E. Wu, "Optima diagnostic methods for wiring interconnects," *IEEE Trans. Computer-Aided Design*, vol. 11, no. 9, pp. 1161-1166, Sep. 1992.

[8] E.J. Marinissen, B. Vermeulen, H. Hollmann, and R.G. Bennetts, "Minimizing pattern count for interconnect test under a ground bounce constraint**,"** *IEEE Design & Computers*, Vol. 20, No. 2, pp. 8-18, Mar-April, 2003.

[9] M. Kaneko and K. Sakaguchi, "Oscillation fault diagnosis for analog circuits based on boundary search with perturbation model," in *Proc. ISCAS*, pp93-96, 1994.

[10] K. Arabi and B. Kaminska, "Oscillation-based test strategy for analog and mixed-signal integrated circuits," in *Proc. VTS*, 1996.

[11] K. S.-M. Li, C.-L. Lee, C. Su, J.E. Chen, "Oscillation ring based interconnect test for SOC" in *Proc. ASPDAC*, pp. 184-187, 2005.

[12] F. DaSilva, Y. Zorian, L. Whetsel, K. Arabi, R. Kapur, "Overview of the IEEE P1500 standard," in *Proc. ITC.*, pp. 988-997, 2003.

Table I: Experimental results for Interconnect Diagnosis both for Predetermined and Adaptive Methods.

Circuit	Statistics				Predetermined			Analysis				Adaptive		
	#core	#pad	#hyp	#net_ segment	$\|R_t\|$	$\|R_d\|$	$\|R_d\|/\|R_a\|$	#One Ring	#No Diag	#Equ Class	$\|R_d\|$–$\|R_t\|$	max. EC	$\|R_a\|$	$\|R_d\|/\|R_a\|$
ac3	27	75	211	416	133(33.3ms)	374(93.5ms)	2.81	389	323	68	241	8	140(35ms)	2.67
ami33	33	42	117	343	242(60.5ms)	303(75.8ms)	1.25	309	126	59	61	5	246(61.5ms)	1.23
ami49	49	22	361	475	156(39ms)	386(96.5ms)	2.47	406	337	88	230	9	162(40.5ms)	2.38
apte	9	73	92	136	73(18.3ms)	122(30.5ms)	1.67	127	94	40	49	4	76(19ms)	1.61
hp	11	45	72	195	81(20.3ms)	164(41ms)	2.02	176	145	51	82	7	87(21.8ms)	1.89
xerox	10	2	161	356	218(54.5ms)	342(85.5ms)	1.57	346	214	86	124	5	222(55.5ms)	1.54
Comp.					0.9679								1	

Table II: Concurrent Test Sessions.

Circuit	$\|R_d\|$	$\|R_c\|$ (worst case)	$\|R_d\|$-$\|R_c\|$
ac3	374	373	1 (0.27%)
ami33	303	290	17 (5.86%)
ami49	386	352	34 (9.66%)
apte	122	119	3 (2.52%)
hp	164	160	4 (2.50%)
xerox	342	327	15 (4.59%)
Comparison		1	4.57%

Table III: Comparison between Theoretical Bounds and Experimental Results.

Circuit	#NoDiag	#EquClass	Eq (3) (#NoDiag-#EquClass)	Extra Rings ($\|R_d\|$–$\|R_t\|$)	(#NoDiag-#EquClass) and ($\|R_d\|$–$\|R_t\|$)
ac3	323	68	255	241	14 (5.49%)
ami33	126	59	67	61	6 (8.96%)
ami49	337	88	249	230	19 (7.63%)
apte	94	40	50	49	1 (2.00%)
hp	145	51	94	82	12 (12.77%)
xerox	214	86	128	124	4 (3.13%)
Comparison				1	6.64%