JOURNAL OF ELECTRONIC TESTING: Theory and Applications 16, 147–155 (2000)

Oscillation Ring Delay Test for High Performance Microprocessors

WEN CHING WU, CHUNG LEN LEE AND MING SHAE WU
Department of Electronics Engineering, National Chiao Tung University, Hsin Chu, Taiwan, ROC
cllee@cc.nctu.edu.tw

JWU E. CHEN
Department of Electrical Engineering, Chung Hwa University, Hsin Chu, Taiwan, ROC

MAGDY S. ABADIR
Somerset Design Center, Motorola Inc., Austin, TX, USA

Received January 15, 1999; Revised July 6, 1999

Editor: A.P. Ambler

Abstract. This paper proposes a new test scheme, oscillation ring test, and its associated test circuit organization for delay fault testing for high performance microprocessors. For this test scheme, the outputs of the circuit under test are connected to its inputs to form oscillation rings and test vectors which sensitize circuit paths are sought to make the rings oscillate. High speed transition counters or oscillation detectors can then be used to detect whether the circuit is working normally or not. The sensitizable paths of oscillation rings cover all circuit lines, detecting all gate delay faults, a large part of hazard free robust path delay faults and all the stuck-at faults. It has the advantage of testing the circuit at the working speed of the circuit. Also, with some modification, the scheme can also be used to measure the maximum speed of the circuit. The scheme needs minimal simple added hardware, thus ideal for testing, embedded circuits and microprocessors.

Keywords: oscillation ring testing, delay fault testing, sensitized path, gate delay fault, robust path dealy fault, stuck at fault, hazard-free path delay fault, multiple reconvergent fanout, flunk lines

1. Introduction

Testing digital logic circuits, especially for high performance microprocessors which work at a high speed, is a difficult and expensive task. Due to the difficulty, most testing primarily focuses on stuck-at faults. However, with the increasing performance and complexity of logic circuits, stuck-at fault testing becomes insufficient to guarantee an acceptable quality level of proper system operation. Delay fault testing is one of possible exercises to improve the testing efficiency to guarantee the logic circuit quality. There is much research on delay fault testing [1–14]. For example, the fault models treated include gate delay fault models, path delay faults, robust delay faults, non-robust delay faults, and hazard-free delay faults [4–14]. However, for all the approaches treated, the testing setup is rather complicated, i.e., it needs two sets of latches, each with a separately precisely controllable clockings to apply the initialization-excitation test pattern pairs and read out test result respectively [4]. In addition, to implement the above delay testing schemes to microprocessors, a scan design for testability needs to be incorporated to shift patterns into and out of the circuit, making the testing be very inefficient, i.e., take much test application time.

An oscillation ring, cascaded by odd number of inverters, has been used to evaluate the speed of integrated circuits [15]. It has also been used to generate control voltage sets, to extract circuit transconductance, phase noise [16–18]. This paper proposes a new test scheme to test all the gate delay faults and most of hazard-free path delay faults, as well as stuck-at faults, of embedded circuits of a microprocessor [19, 20]. The scheme requires minimal extra hardware overhead but eliminates the use of a tester and can test the circuits at working speed of the microprocessor. In addition, the scheme, when modified a little bit, can be used to measure the working speed of the circuit. The scheme is based on the use of oscillation rings, i.e., the outputs of the circuit under test are connected to some of inputs of the circuit with odd inversion parity to form oscillation rings. Under appropriate input pattern to the un-connected inputs, the oscillation rings will oscillate. If faults exist on the paths of oscillation rings, no oscillation will occur. Observing the outputs of the oscillation rings, one can tell whether the circuit works normally or not. A similar approach was employed to test analog circuits [21], for which the circuit under test is converted to an oscillation circuit. During the test mode, the faults which cause a deviation of the oscillation frequency from its tolerance band are considered detected.

2. Oscillation Ring Test

To explain this oscillation ring test scheme, an example circuit, C17, of Fig. 1(a) is used for demonstration. For circuit C17, A, B, C, D, and E are primary inputs, P and Q are primary outputs, and there are 34 stuck-at faults, 34 gate delay faults, and 22 path delay faults. For this circuit, we first consider path CGIJLMP which is shown in bold line in Fig. 1(b). This path is first selected to be sensitized. Its output P is to be connected to the input C. To make the path oscillate, the values of off-path inputs must be set to be non-controlling values to sensitize the path. So B, D, and H are set

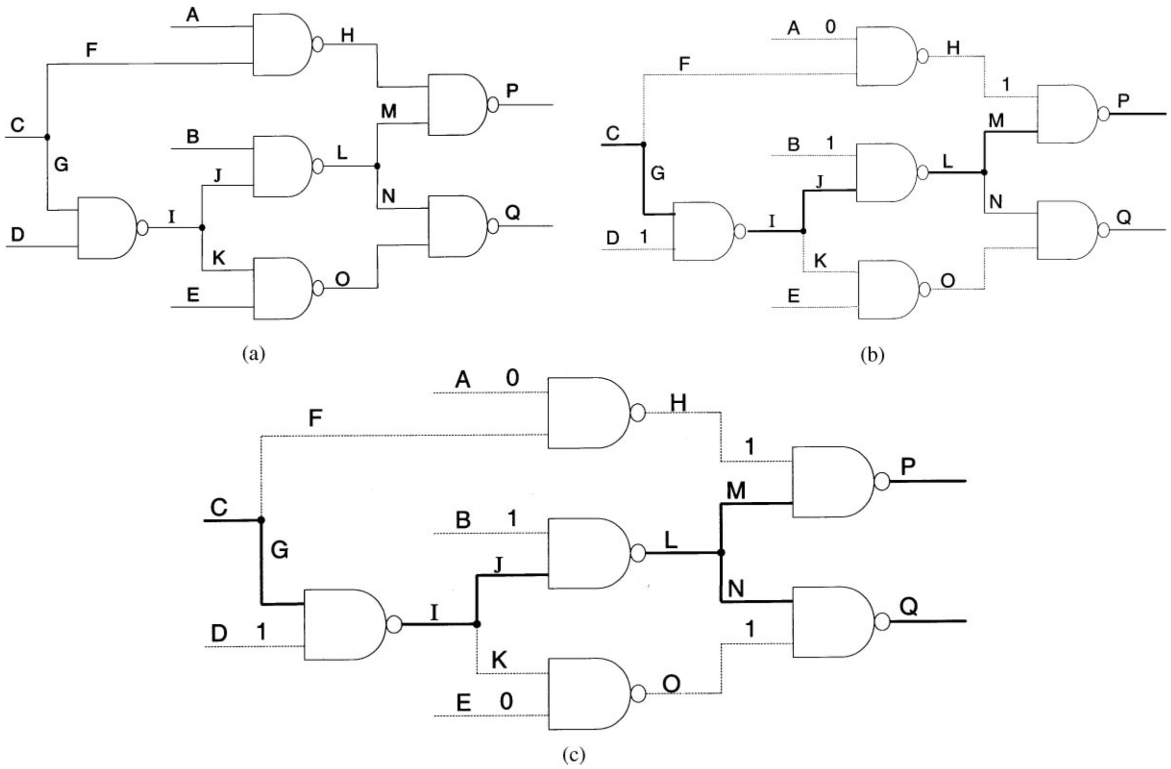

Fig. 1. (a) The example circuit C17; (b) sensitization of the path CGIJLMP and flunk lines (the dotted lines); (c) sensitization of multiple paths CGIJLMP and CGIJLNQ.

to 1. These non-controlling values must be justified at other primary inputs, so inputs (A, B, D, E) are set to (0, 1, 1, X) respectively, where X is “don’t care”. That is, under this pattern, we can form a ring with an odd inversion parity by connecting primary output P and primary input C. In the presence of any stuck-at fault at lines C, G, I, J, M, and P, i.e., C↑ (C stuck-at 1), C↓(C stuck_at 0), G↑, G↓, I↑, I↓, J↑, J↓, L↑, L↓, M↑, M↓, P↑, and P↓, the oscillation ring will not oscillate. By observing the output of the oscillation ring, P, one can tell whether the above faults exist or not.

It is easy to see that, in addition to the path CGIJLMP, there is another path CGIJLNQ, as shown in Fig. 1(c), which can be sensitized under the same test pattern. Observing the output at line Q, one can also tell whether additional faults, N↑, N↓, Q↑ and Q↓, exist. This path CGIJLNQ is called the *compatible path* to the path CGIJLMP. The first selected path is called the *primary path* and its compatible paths are called *secondary paths*. In fact, either of the output of the primary path or the secondary path can be connected to the input to form an oscillation ring. In Fig. 1(b), in order to sensitize the secondary path CGIJLNQ, E is set to 0 by justification, as shown in Fig. 1(c). The pattern is then (A, B, C, D, E) = (0, 1, P, 1, 0), where C = P means that output P is connected to input C. If the path CGIJLNQ is selected to be the primary path, the pattern is (A, B, C, D, E) = (0, 1, Q, 1, 0). To improve the test efficiency and reduce the size of test set, it is desirable to find as more secondary paths as possible for an oscillation ring.

In Fig. 1(b), in addition to faults C↑, C↓, G↑, G↓, I↑, I↓, J↑, J↓, L↑, L↓, M↑, M↓, N↑, N↓, P↑, P↓, Q↑ and Q↓, if there are stuck-at faults whose fault effects change the values of the off-path inputs, i.e., faults B↓, D↓, H↓ and O↓, the formed ring will not oscillate either. Hence, B↓, D↓, H↓ and O↓ are also detected by this ring. Furthermore, in the circuit, the existence of either A↑ or E↑ faults will dynamically block oscillation of the ring or the path NQ. This will be explained as follows: It is assumed that A↑ (or E↑) is present. When F (or K) is 0, the value of H (or O) is 1, then oscillation occurs. However, when F (or K) is 1, the value of H (or O) is 0, no oscillation occurs. So, A↑ and E↑ can be detected by oscillation test for the pattern (A, B, C, D, E) = (0, 1, P, 1, 0) or (0, 1, Q, 1, 0). However, it is interesting to note that patterns (A, B, C, D, E) = (0, 1, 0, 1, 0) and (0, 1, 1, 1, 0) can not detect A↑ and E↑.

In the above oscillation patterns, if there are gate delay faults or path delay faults on the selected sensitizing paths, the period of oscillation will exceed the clock cycle time. So the patterns can also detect gate delay faults: C↗ (slow-to-rise gate delay fault) , C↘ (slow-to-fall gate delay fault) , G↗, G↘, I↗, I↘, J↗, J↘, L↗, L↘, M↗, M↘, N↗, N↘, P↗, P↘, Q↗, and Q↘, and path delay faults CGIJLMP↗ (slow-to-rise path delay fault), CGIJLNQ↗, CGIJLMP↘ (slow-to-fall path delay fault), and CGIJLNQ↘. Because the values of the off-path inputs are set to static non-controlling values, the delay faults are hazard-free robustly detected.

In summary, when the pattern (A, B, C, D, E) = (0, 1, P, 1, 0) or (0, 1, Q, 1, 0) is applied to the circuit, the formed ring and sensitized path detect 26 stuck-at faults, 18 gate delay faults, and 4 path delay faults. The test efficiency is high.

3. Oscillation Ring to Measure the Working Speed of the Tested Circuit

The oscillation ring can be used to measure the working speed of the circuit under test. Fig. 2(a) shows a simple configuration where the working speed of the tested circuit can be measured by observing the period of the oscillation waveform of the connected rings. The oscillation pulse trains are fed into a counter to be counted the number of the pulses within a specified time slot of “*T*”. Counting the number of the pulse, one can tell the working speed of the tested path. If the tested path is the longest path of the circuit, the obtained speed is the working speed of the circuit under test. Also, due to the fact that the path delays for the rising waveform and the falling waveform of a path are different, the duty cycle of the oscillation waveform reveals the rising waveform delay and the falling waveform delay of the tested path respectively. Fig. 2(b) shows a circuit where a fast counter is used to measure the duty cycle of the oscillation waveform. The respective counter pulses in the “1” half cycle and the “0” half cycle can be counted with a similar counter setup of Fig. 2(a) respectively.

The above simple testing circuit can be either built-in along with the circuit under test or attached externally during testing.

4. Oscillation Rings Formed by Multiple Reconvergent Fanouts

In the above, the oscillation ring is formed with only one sensitized path. However, it is also possible that an

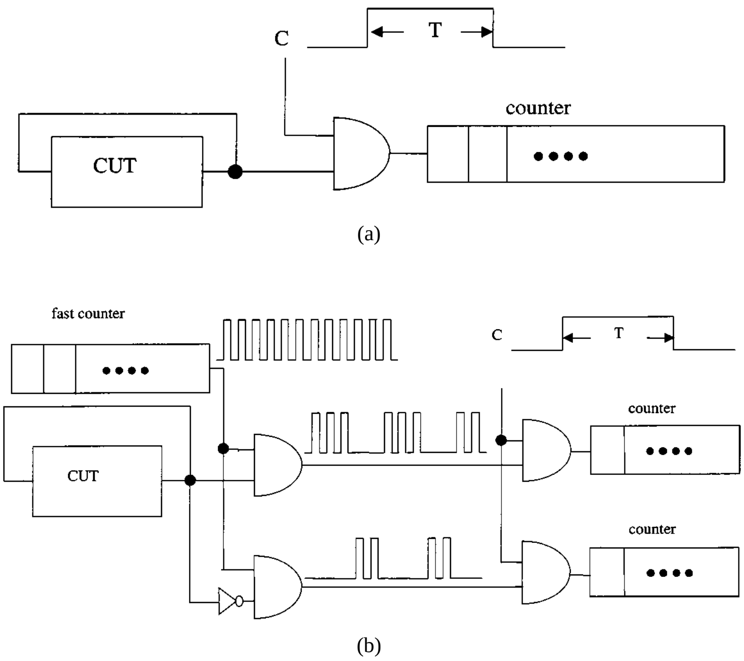

Fig. 2. (a) The configuration to measure the working speed of the tested circuit. (b) The circuit configuration to measure the duty cycle of the oscillation waveform to measure the rising and falling path delays of the test path.

oscillation ring is formed by more than one sensitized path. Fig. 3 show such cases that two reconvengent fanout branches form the oscillation ring. In Fig. 3(a), the two fanout branch paths reconverge at an AND gate. For this case, if the path delay, Δ_1, of *path 1* is less than the path delay, Δ_2, of *path 2*, but $\Delta_2 < 2\Delta_1$, the duration of the "0" (or "1") half cycle of the oscillation waveform at the output of the AND (or NAND) gate will be determined by Δ_2 and that of the "1" (or "0") half cycle will be determined by Δ_1. This is because

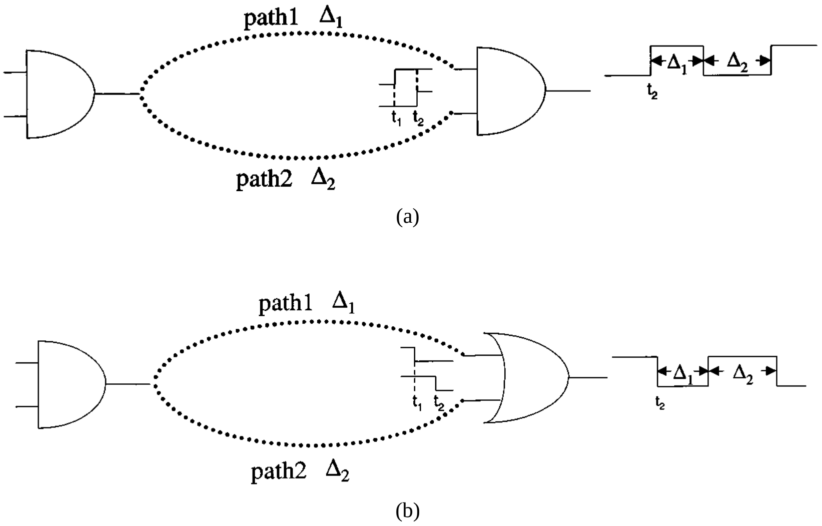

Fig. 3. The oscillation ring formed by two sensitized paths with the reconvergent gate to be (a) AND gate, and (b) OR gate.

for an AND (NAND) gate, "0" is the controlling value. However, if $\Delta_2 > 2\Delta_1$, the "1" ("0") duration of the waveform at the AND (NAND) gate output became irregular, because the oscillation waveform may travel around *path 1* more than twice. Similarly, if the two fanout branch paths reconverge at an OR (or NOR) gate, as shown in Fig. 3(b), the above results still exist except that the waveform polarities exchange. In the above, it has been assumed that the rising path delay and the falling path delay of *path 1* (or *path 2*) are the same.

For the oscillation rings formed by multiple sensitized paths, the gate delay faults and the path delay faults occurring at one of the above sensitized paths can not be detected since its effect will be masked by the other path, i.e., the oscillation ring still oscillates but with a changed oscillation waveform. To detect the faults, the duty cycle of the changed waveform needs to be measured. The circuit of Fig. 2(b) can be used to do the job.

For the stuck-at faults occurring at the sensitized paths, they can be detected if they stuck-at the logic values that make the sensitized path input to the reconvergent gate be at the controlling value, i.e., "0" if the reconvergent gate is an AND or NAND gate, and "1" if the reconvergent gate is an OR or NOR gate.

Because of the difficulty mentioned above caused by the multiple sensitized path oscillation ring, it is desirable to form the oscillation ring with only one sensitized path whenever it is possible in order to facilitate this test scheme.

5. Implementation and Test Organization

A. Flunk Lines

To completely test a circuit, we have to find a set of sensitizable paths to cover, at least once, all circuit lines. First we find a primary sensitizing path by finding the necessary input pattern. Then we attempt to find other compatible secondary paths for the same pattern. As mentioned previously, it is desirable to find as many secondary paths as possible to increase the detection efficiency. However, to find the optimum number of compatible paths is a non-trivial problem. In the following, *flunk lines* are first introduced.

When a pattern to sensitize a primary oscillation path is generated, the values on some circuit lines are fixed and they can not be selected to be secondary paths anymore. We define these lines to be *flunk lines* under the generated pattern. A compatible secondary path can not pass through any flunk line. Identifying flunk lines greatly reduces the search time for secondary paths.

There are three types of flunk lines. The first is the line whose value has been set to sensitize the selected path(s). In Fig. 1(b), A, B, D, and H are such lines. The second type of flunk line is the line whose neighboring inputs are oscillating lines implied by the selected oscillation rings. In Fig. 1(b), E and O are such lines. The third type lines are the lines whose succeeding exit lines are flunk lines. As an example, consider lines F and K in Fig. 1(b), their only succeeding exit lines are H and O respectively, which are flunk lines, so they are flunk lines.

B. The Selection of a Sensitization Path

The procedure to select the oscillation ring sensitized paths is as follows:

```
OSCILLATION RING SENSITIZATION

While(there is an unselected line)
{     Select a primary target line;
SECOND:      For all candidate paths
            {     Set values of off-path inputs;
If (sensitizable( ))
{    Update test conditions;
          Break;
}
          }
          Flunk_mark( );
/* Multiple oscillation ring */
            If (there is an unselected line which is
                not flunk line)
            {     Select a secondary target line;
                  Go to SECOND;
            }
            Record test conditions;
            Update selected line;
            Reset flunk lines;
     }
```

This procedure starts with first selecting an unselected line as a primary target line. All the paths, including the target line, which do not pass through any flunk line are candidate paths. They are potentially sensitizable. When a candidate primary path is selected, the values of off-path inputs are set to be non-controlling values to sensitize the path. These non-controlling values are justified to primary inputs by an algorithm like the "BACK" algorithm [22] which is commonly used in test pattern generation. If the selected candidate path is not sensitizable, another candidate path is selected until a sensitizing path is found or all the candidate paths fail to be sensitized. Once a sensitized path is found, the flunk lines are marked and the procedure tries to find other compatible secondary sensitizable paths. Once a test pattern for sensitizing multiple paths is generated, then the test conditions, including the test pattern, the primary input, the primary output, and inversion parities of paths, are recorded. This process continues until all lines have been selected.

In the above procedure, flunk line marking greatly reduces the unfruitful backtraking associated with finding compatible secondary paths. The procedure *Flunk_mark*( ) is described as follows:

```
Flunk_mark( )
    {   For all the lines $i$ whose value is 0 or 1
                Mark $i$ as a flunk line and Push
                    ( $i$, Stack_fl );
        For all the lines $i$ whose value is OSCI
                For all side-inputs $j$ of $i$
                        Mark $j$ as a flunk line and Push
                            ( $j$, Stack_fl );
        While (Stack_fl is not empty)
        {   $i = $ Pop (Stack_fl);
Backward:               If ( $i$ is a FOB of a FOS $j$ )
                        If ( all FOBs of $j$ are flunk lines )
                                Mark $j$ as a flunk line and
                                    Push ( $j$, Stack_fl );
                Else
                        For all gate inputs $j$ of $i$
                                Mark $j$ as a flunk line and
                                    Push( $j$, Stack_fl );
Forward:                If ( $i$ is a FOS )
                        For all FOBs $j$ of $i$
                                Mark $j$ as a flunk line and
                                    Push ( $j$, Stack_fl );
                Else If (gate output $j$ of $i$ is not a flunk
                        line)
                    If (all of gate inputs of $j$ are flunk
                        lines )
                        Mark $j$ as a flunk line and
                            Push( $j$, Stack_fl );

        }
    }
```

As mentioned previously, there are three types of flunk lines. Under the test pattern to sensitize selected paths, the first type of flunk lines whose values have been set are marked and pushed into *Stack_fl*. The second type of lines whose neighboring inputs are oscillating are also marked and pushed into the stack *Stack_fl*. After the two types of flunk lines are pushed into *Stack_fl*, one of flunk lines, line i, is removed from *Stack_fl* to find the third type of flunk lines. If a new flunk line is found, it is pushed into *Stack_fl*. The process to find the third type of flunk lines is divided into two phases, Backward and Forward. The Backward phase advances from i backward to its fanout stem or gate inputs. Suppose that i is a fanout branch, its fanout stem j is marked as flunk line if all of associated fanout branches of j are flunk lines. Otherwise (i.e., i is not a fanout branch), i is the only exit line of its gate inputs j, so all the associated gate inputs j are marked as flunk lines. The Forward phase advances from i forward to its fanout branches or gate output. Suppose that i is a fanout stem, all of its fanout branches j are marked as flunk lines. Otherwise (i.e., i is not a fanout stem), its gate output j is marked as flunk line if all the associated gate inputs j are flunk lines.

When a compatible secondary sensitized path is found, the test conditions, including the revised test pattern, the additional associated primary input, the primary output and the inversion parity of the secondary selected path, are updated. At the end of the procedures, all the circuit lines are either marked as sensitized paths or flunk lines.

C. Oscillation Ring Test Organization

A simple implementation of the oscillation ring test scheme is shown in Fig. 4. The purpose of the input stage is to feed the test patterns to the circuit-under-test. The output stage is to generate a detection signal when faults are detected. Both the input stage and the

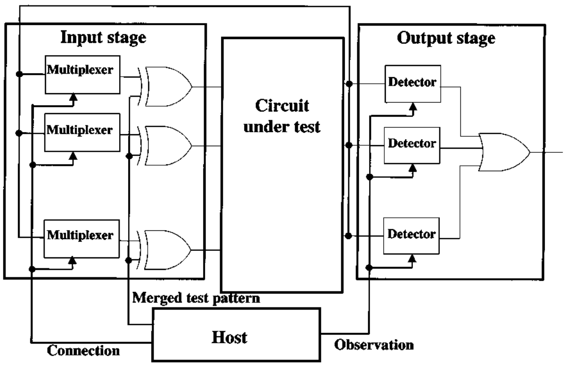

Fig. 4. Test organization of oscillation ring.

output stage are controlled by a test controller host. The input stage basically is a multiplexer network with XOR gates which are both controlled by the test controller. The multiplexer network connects the primary outputs with primary inputs and the XOR gates provide the required odd inversion parity. This input stage can be implemented by a PLA. The output stage is an "OR" function of all detectors that are activated by the controller according to the test pattern. For a test pattern, only the detectors whose associated primary outputs connected to primary inputs are activated. The detector could be a simple transition counting circuit to detect the oscillation of primary outputs. The input stage/output stage setup could be external to the circuit-under-test or be built-in with the circuit-under-test. However, for a microprocessor for which the scan design is used, it introduces additional delay of a multiplexers and an XOR gate to the scan cell.

6. Experimental Results and Discussion

Fig. 5 shows the complete oscillation ring test for the C17 circuit, where the feedback connections, inversion parities of the feedback paths, test patterns, and observation outputs are listed. The test consists of four patterns to detect all stuck-at faults and all gate delay faults. These four patterns can also detect 14 path delay faults which is 64% of the total path delay faults. For the first three patterns, two sensitized paths are found and both outputs P and Q are observation outputs. For the first pattern, both P and Q are connected to inputs A and E, respectively to form two separate oscillation rings.

The procedures outlined in Section 3B have been implemented as a C program called "Ortest". It was applied to the ISCAS benchmark circuits [23, 24] to generate complete oscillation ring tests. Table 1 shows the results for this experiment. The results were generated on a SUN 4/75 Sparc 2 workstation. In the table,

Connection	Inversion parity	Test pattern	Observation
ABCDE	ABCDE	ABCDE	PQ
P---Q	0---0	-010-	yes yes
-P---	-0---	0-x00	yes yes
--Q--	--1--	10-11	yes yes
---P-	---1-	011-x	yes no

Fig. 5. The complete oscillation ring tests for circuit C17.

Table 1. Experimental oscillation ring test results.

Circuit name	*#line*	*#line_u*	*#line_a*	*#pattern*	CPU time (sec)
c17	17	0	0	4	0.00
s27c	26	0	0	6	0.02
c74181	192	0	0	41	0.40
s208c	208	0	0	38	0.39
s298c	298	1	0	35	0.93
s344c	335	0	0	32	1.06
s349c	340	4	0	31	1.12
s382c	382	3	0	35	1.14
s386c	386	0	0	89	1.35
s400c	400	16	0	35	1.47
s420c	420	0	0	63	1.19
c432	419	1	0	86	527.41
s444c	444	67	0	38	1.88
s499c	477	0	0	115	1.57
c499	491	137	0	93	280.19
s510c	510	2	0	92	2.15
s526c	526	5	0	74	2.48
s635c	635	0	0	64	2.02
s641c	637	3	0	51	3.27
s713c	713	79	0	52	57.56
s820c	820	8	0	182	6.06
s838c	838	0	0	110	4.12
c880	880	0	0	48	5.97
s938c	938	0	0	187	13.69
s953c	953	2	0	125	5.87
s967c	964	3	0	133	6.13
s991c	974	31	0	76	87.92
s1196c	1196	69	0	178	14.46
s1238c	1238	262	0	107	12.44
s1269c	1261	24	0	71	590.43
c1355	1347	727	170	68	*
s1423c	1423	58	0	67	230.67
s1488c	1488	24	0	228	16.79
s1494c	1494	40	0	230	16.58
s1512c	1494	37	0	102	58.65
c1908	1896	20	187	159	*
c2670	2046	36	0	155	296.29
s3271c	3252	134	0	67	7,446.70
s3330c	3330	1	0	275	123.65
s3384c	3375	105	0	96	2,080.15
Sum	39063			3738	

*More than one day.

the circuits with the postfix c are the combinational part of the ISCAS sequential benchmark circuits. For each circuit, the number of total lines (*#line*), the number of lines which are hazard free robustly path delay fault untestable (*#line_u*), the number of lines which were aborted during the test generation (*#line_a*), and the number of patterns (*#pattern*), are listed. The condition for hazard free robustly path delay fault testing is so strict that many paths are untestable. Due to the strict condition of robustly hazard free path delay fault testing, finding sensitized paths sometimes involve extensive backtracking and can exceed a certain limit of CPU time. In those cases the line is aborted. This is the reason, the computation time for circuits c1355 and c1908 exceeded one day. The *#pattern* listed are the number of patterns which cover all the lines, except the untestable and aborted lines. The number of test patterns required to all the stuck-at faults and gate delay faults are much smaller. For example, to detect stuck-at faults for circuit c74181, only 15 patterns are needed. However, it takes 41 patterns to detect all path delay faults which cover all circuit lines. Note that, in general, it only takes a significantly smaller number of patterns, approximately one tenths of the total circuit lines, to cover all the circuit lines for a circuit. In the scan design of a high performance microprocessor, this means that the number of patterns needed to be shifted into inputs of the scanned combinational circuits is small. Also, this number of test patterns only hazard- free robustly tested a subset of the total number of delay paths. In practice, it can choose a set of critical paths of the circuits to form oscillation rings to test.

7. Conclusion

In this paper, a new test scheme, based on the concept of oscillation rings to detect faults for combinational circuits (or scan type of circuits) is proposed and demonstrated. The scheme connects outputs of the circuit-under-test to its inputs with odd inversion parity and applies an appropriate input pattern to the unconnected inputs to sensitize paths of the circuit, making them oscillation rings. By observing whether the outputs oscillate or not at the target frequency one can tell whether the circuit is working properly or not. The testing scheme needs simple added hardware which can be applied externally or built internally in the circuit when employed to test microprocessors. The testing efficiency of the scheme is high. It can detect all stuck-at faults, all gate delay faults and part of the hazard-free robust path delay faults at the working speed of the circuit. A procedure to find multiple number of sensitized paths has also been presented. Experimental results show that the number of patterns to cover all the lines of a circuit is approximately only one tenths of the total number of the circuit lines. One biggest advantage of this test scheme is that it can test the circuit at speed. It can also be used to measure the maximum speed of a given microprocessor. This is done by measuring the maximum frequency of the set of the critical paths (longest paths) of the microprocessor.

References

1. A.K. Pramanick and S.M. Reddy, “On the Detection of Delay Faults,” *Proc. of International Test Conference*, 1988, pp. 845–856.
2. V.S. Iyengar, B.K. Rosen, and I. Spillinger, “Delay Test Generation 1-Concepts and Coverage Metrics,” *Proc. of International Test Conference*, 1988, pp. 857–866.
3. V.S. Iyengar, B.K. Rosen, and J.A. Waicukauski, “On Computing the Sizes of Detected Delay Faults,” *IEEE Trans. On CAD-9*, pp. 299–312, 1990.
4. G.L. Smith, “Model for Delay Faults Based upon Paths,” *Proc. of International Test Conference*, 1985, pp. 342–349.
5. W.N. Lee, S.M. Reddy, and S.K. Sahni, “On Path Selection in Combinational Circuits,” *IEEE Trans. on CAD-8*, pp. 56–63, 1989.
6. S.M. Reddy, C.J. Lin, and S. Patil, “On Automatic Test Pattern Generation for the Detection of Path Delay Faults,” *Proc. of International Conference on Computer Aided Design*, 1987, pp. 284–287.
7. K.T. Cheng and H.C. Chen, “Classification and Identification of Nonrobust Untestable Path Delay Faults,” *IEEE Trans. on CAD-15*, pp. 1027–1034, 1996.
8. E.S. Park and M.R. Mercer, “Robust and Nonrobust Tests for Path Delay Faults in a Combinational Circuit,” *Proc. of International Test Conference*, 1987, pp. 1027–1034.
9. M.C. Lin, J.E. Chen, and C.L. Lee, “TRANS: A Fast and Memory Efficient Path Delay Fault Simulator,” *Proc. of European Conference on Design Automation and Test*, 1994, pp. 508–512.
10. K. Fuchs, M. Pabst, and T. Rossel, “RESIST: A Recursive Test Pattern Generation Algorithm for Path Delay Faults Considering Various Test Classes,” *IEEE Trans. on CAD-13*, pp. 1550–1562, 1994.
11. J. Savir and W.H. McAnney, “Random Pattern Testability of Delay Fault,” *Proc. of International Test Conference*, 1986, pp. 163–173.
12. A.K. Pramanick and S.M. Reddy, “Sufficient Multiple Path Propagating Tests for Delay Faults,” *Journal of Electronic Testing: Theory and Applications*, Vol. 7, pp. 157–167, 1995.
13. S. Kundu, S.M. Reddy, and N.K. Jha, “Design of Robustly Testable Combinational Circuits,” *IEEE Trans. on CAD-10*, pp. 1036–1048, 1991.
14. I. Pomeranz, S.M. Reddy, and P. Uppaluri, “EST: A Non-enumerative Test Generation Method for Path Delay Faults in

Combinational Circuits," *Proc. of International Conference on Design Automation*, 1993, pp. 439–445.

15. S.D. Lohit and H.R. Thomas, "Application of Ring Oscillators to Characterize Transmission Lines in VLSI Circuits," *IEEE Trans on Component, Packaging, and Manufacturing Technologies*, part-B, Vol. 18, pp. 651–657, 1995.
16. K. Kurita, T. Hotta, T. Nakano, and N. Kitamura, "PLL-Based BiCMOS on-Chip Clock Generator for Very High-Speed Microprocessor," *IEEE JSSC-26*, pp. 585–589, 1991.
17. M. Kaneko and K. Sakaguchi, "Oscillation Fault Diagnosis for Analog Circuits Based on Boundary Search with Perturbation Model," *IEEE Int. Symp. on Circuits and Systems*, pp. 93–96, 1994.
18. B. Razavi, "A Study of Phase Noise in CMOS Oscillators," *IEEE JSSC-31*, pp. 331–343, 1995.
19. W.C. Wu, "Delay Testing and Fault Simulation for Digital Circuits," Ph.D. Thesis, Department of Electronics Engineering, National Chiao Tung University, June 1997.
20. W.C. Wu, C.L. Lee, J.E. Chen, and M. Abadir, "Oscillation Ring Delay Test for High Performance Microprocessor," presented in 1st Int. Workshop on Microprocessor Test and Verification, 1998.
21. K. Arabi and B. Kaminska, "Oscillation-Based Test Strategy for Analog and Mixed-Signal Integrated Circuits," *Proc. of VLSI Test Symp.*, 1996.
22. W.T. Cheng, "The BACK Algorithm for Sequential Test Generation," *International Conference on Computer-Aided-Design*, 1991, pp. 214–218.
23. F. Berglez and H. Fujiwara, "Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in FORTRAN," Special Session on ATPG and Fault Simulation, *IEEE Int. Symp. On Circuits and Systems*, 1985.
24. F. Berglez, D. Brglez, and K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," *IEEE Int. Symp. On Circuits and Systems*, pp. 1929–1934, 1989.

Wen Ching Wu was born in Ilan, Taiwan on October 15, 1966. He received the BS, MS, and Ph.D. degrees in electronic engineering from National Chiao Tung University, HsinChu, Taiwan. Since 1997 he has been with the Electronics Research & Service Organization/Industrial Technology Research Institute, Chutung, Taiwan, where he is currently a testing engineer.

His principal areas of interest include fault simulation, delay testing, mixed-signal testing, and design for testability.

Chung Len Lee obtained his B.S. from National Taiwan University in 1968 and M.S. and Ph.D. from Carnegie Mellon University in 1971 and 1975 respectively, all in Electrical Engineering. He has been with Department of Electronics Engineering, National Chiao Tung University since 1975, engaging in teaching and research in the fields of semiconductor devices, integrated circuits, VLSI, computer aided design and testing. He has supervised over 100 M.S. and Ph.D. students to complete their thesis and has published over 200 papers in the above areas. He has been involved in various technical activities in the above areas in Taiwan as well as in Asia. He is on the editorial board of JETTA.

Ming Shae Wu was born in 1969 in Taiwan. He received his B.S. and M.S. in Electronics Engineering at National Chiao Tung University. Now he is a Ph.D. student at National Chiao Tung University and mainly studying in the testing of VLSI circuits.

Jwu E. Chen is an associate professor in the Department of Electrical Engineering, Chung-Hua University, Taiwan. His research interests include VLSI testing, yield analysis, test management and behavior and psychology of testing. He received B.S., M.S. and PhD degrees in electronics engineering from the National Chiao Tung University, Taiwan. He is a member of the IEEE and the computer society.

Magdy S. Abadir received the B.S. degree with honors in Computer Science from the University of Alexandria, Egypt in 1978, the M.S. degree in Computer Science from the University of Saskatchewan, Saskatoon, Canada, in 1981, and the Ph.D. degree in Electrical Engineering from the University of Southern California, Los Angeles, in 1986. Currently he is the Chief Technologist for verification and Manager of the Test and Logic Verification Methodology and Tools group at Motorola's PowerPC Design Center (Somerset) in Austin, Texas. Prior to that he was the General Manager of Best IC Labs in Austin Texas (a Burn-in and Test Engineering firm). From 1986 to 1994 he worked at the Microelectronics and Computer Technology Corporation (MCC) as a senior member of the technical staff. Dr. Abadir has co-founded and chaired a series of international workshops on the economics of design, test and manufacturing and on microprocessor test and verification. He has co-edited several books on those subjects, and he also published over 70 technical journal and conference papers in the areas of test economics, design for test, computer-aided design, high-level test generation, and design verification and economics.