Hierarchical Fault Tracing for VLSI Sequential Circuits from CAD Layout Data in the CAD-Linked EB Test System

Katsuyoshi MIURA Koji NAKAMAE Hiromu FUJIOKA

Department of Information Systems Engineering, Faculty of Engineering,
Osaka University
Yamada-Oka 2-1, Suita, Osaka, 565 Japan

Tel: +81-6-879-7813 Tel: +81-6-879-7811 +81-6-879-7810
Fax: +81-6-879-7812 Fax: +81-6-879-7812 +81-6-876-4599
e-mail: miura@ise.eng.osaka-u.ac.jp e-mail:nakamae@ise.eng.osaka-u.ac.jp e-mail: fujioka@ise.eng.osaka-u.ac.jp

Abstract—A previous hierarchical fault tracing method for combinational circuits which requires only CAD layout data in the CAD-linked electron beam test system is expanded as applicable to sequential circuits. The characteristics in the method remains unchanged that allows us to trace a fault hierarchically from the top level cell to the lowest primitive cell and from the primitive cell to the transistor-level circuit in a consistent manner independently of circuit functions. The applied results to the CAD layouts of some sequential CMOS benchmark circuits show our superiority to the guided-probe method where circuit logical functions are first extracted from the CAD layout data and then the guided-probe testing is executed.

I. Introduction

The electron beam (EB) test system linked to the CAD database [1] has been widely used for testing VLSIs. Through the fault localization process by using the method such as the guided probe diagnosis [2], [3] with the CAD-linked EB test system, a gate-level fault or a cell-level fault is determined. On the other hand, the performance faults such as the delay fault have become a serious issue [4]. In order to search the cause of the performance fault, it becomes necessary to trace the fault in the lowest level of the circuit, that is, in the transistor-level unlike the gate-level. In the CAD-linked EB test system, the CAD schematic or netlist and the mask layout data should be mutually linked. However, in order to prepare the CAD data and get the correspondence between the schematic or netlist and the mask layout, a great deal of labor is required.

To deal with the situation described above, we have proposed a hierarchical fault tracing method for a hierarchically structured CAD layout data of combinational circuits [5], [6], [7]. The method needs only CAD layout data: the tedious tasks for CAD linkage is greatly reduced. In the method, a fault is traced hierarchically from the top level cell to the transistor-level circuit in a consistent manner independently of the circuit function, as extracting circuit data successively.

Usually, VLSI includes some sequential circuits which contain memory elements and feedback lines. The output of the sequential circuit is a function of the present state and the input. To determine the output, one needs to know the present state of the memory elements and the logic values at the feedback lines. Thus, a fault in a sequential circuit is not detected within one time frame by simply applying an input vector and observing the output [8].

In this paper, we expand our previous hierarchical fault tracing method for combinational circuits as applicable to such sequential circuits. Then the expanded method is applied to the CAD layouts of some sequential CMOS benchmark circuits and compared with the guided-probed method where circuit logical functions are first extracted from the CAD layout data and then the guided-probe testing [9] is executed.

II. Overview of Previous Hierarchical Fault Tracing for Combinational Circuit

The previous hierarchical fault tracing method for combinational circuits uses an integrated algorithm that combines a successive circuit extraction from hierarchically structured CAD layout data and a hierarchical fault tracing for the combinational circuit [5], [6], [7].

The fault tracing system consists of a device under test (DUT), a good device to acquire reference waveforms, an LSI tester to stimulate these devices, an EB tester to observe an internal behavior, and a control program.

The procedure of the hierarchical fault tracing is as follows. At first, one of faulty external output pins specified by an LSI tester is selected to be a start point for tracing. The start point is on the top level cell. Next, partial circuit data around the start point are extracted from CAD layout of the top level cell. Then the interconnection to be measured is specified. The signal waveform on the interconnection is measured with the EB tester and is compared with the reference waveform. When the signal is faulty, the fault tracing proceeds upstream until all the upstream interconnections have good signal waveform. Otherwise, the faulty area is specified at the top level. If the pointed faulty area includes a cell, the fault tracing proceeds to the next lower level inside the cell. Repeating the procedure described above, the faulty tracing level goes down. Finally, the fault is localized in the transistor-level.

ASP-DAC '97
0-89791-851-7$5.00

We use two kind of labels called *DC PATH LABEL* and *MEASURED LABEL* in the fault tracing algorithm. The *DC PATH LABEL*s are put on the interconnections and the devices which constitute a DC path (an ELEMENTARY PATH) in order to make the fault tracing proceeds to the upstream. The DC path is defined as a path which connects a starting interconnection to the power supplies V_{DD} and V_{SS} or to an output terminal of a cell instance. The *MEASURED LABEL*s are put on the interconnections that have been tested in order to avoid measuring the same interconnection again.

III. Hierarchical Fault Tracing for Sequential Circuits

In the sequential circuits, a faulty signal may propagate through the same circuit node multiple times because of memory elements and some feedback lines: this requires to allow us to test the same circuit node more than once at different times. The means of solving this problem will be discussed in III-A.

To determine the output of the sequential circuit, one needs to know the present state of the memory elements as well as the logic values at the feedback lines. It follows that the fault in the sequential circuit can not be detected within one time frame by simply applying an input vector and observing the output [8]: the test of the sequential circuit results in a long test sequence. Thus the less number of probing is desirable because of a long tracing time due to a long test sequence. Reduction of probing points will be treated in III-B.

A. Additional information on DC PATH LABELs and MEASURED LABELs

Our previous fault tracing method prevents us from testing the same circuit node again. Let us take a circuit shown in Fig. 1 as an example. Interconnections are labeled 1, 2, $\cdots$, and 12. Arrows indicate the faulty signal flow. Boxes are flip-flops. We start fault tracing at the interconnection 8. After the *DC PATH LABEL*s are put on the interconnections 8 and 2 and the MOS transistors along the DC path drawn by thick lines in Fig. 1, fault tracing proceeds to the interconnection 2. Next, the *DC PATH LABEL*s are put on the interconnection 1 and 12. When the fault tracing proceeds to the interconnection 12, the *DC PATH LABEL* labeling and the fault tracing proceed to the interconnection 11, $\cdots$, 10, 9, and reach at the interconnection 7. At the next step, the fault tracing must proceed to the interconnection 1 through the interconnection 2. The *MEASURED LABEL* previously placed on the interconnection 2, however, put a stop to further fault tracing. There is also a problem on the *DC PATH LABEL*. When fault tracing proceeds to the interconnection 7, the *DC PATH LABEL* labeling is carried out. The *DC PATH LABEL*s have to be put on the interconnections and the MOS transistors along the path drawn by gray line in Fig. 1. On the interconnections 2 and 8 and the MOS transistors along the path drawn by thick line in Fig. 1, however, the *DC PATH LABEL*s have been already placed. If new *DC PATH LABEL*s are put over the old ones, the fault tracing may proceed to a wrong path, that is, to the interconnections 5, 6 and 8.

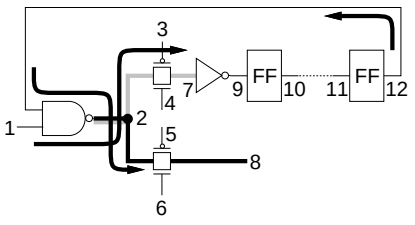

Fig. 1. Example of the fault tracing on the same circuit nodes at different times. The arrows indicate the faulty signal flow. The thick and gray lines show the DC paths in the first and second tracings, respectively.

In order to trace a fault along the correct path, we add the depth number of the fault tracing to the *DC PATH LABEL* and permit that the *DC PATH LABEL*s are overlapped if their depth numbers differ from each other. The depth number indicates the distance from the primary output, that is, the start node of the fault tracing on the traced path. The depth number is 1 at the beginning of the fault tracing, and is incremented when the fault tracing proceeds through the gate electrode of a MOS transistor and through a cell instance.

In order to permit the interconnection to be measured again, if the tracing time is different, we add an information about the tracing time to the *MEASURED LABEL*. In addition, the result of the waveform comparison (Good or Faulty) is recorded in the *MEASURED LABEL* to make the fault tracing efficient.

B. Reduction of probing points

In the sequential circuit, the less number of probing is desirable because of a long tracing time due to a long test sequence. When the same circuit is traced in the second time, making use of the previous information about the extracted circuit data, the tracing path and depth and the propagation delay can much improve the efficiency of the fault tracing. The extracted circuit data is recorded in the circuit data structure [6]. The traced path and depth and the propagation delay can be found from the *DC PATH LABEL* and the *MEASURED LABEL*.

We apply the binary search algorithm to the second tracing at the same circuit node. Let us take a circuit shown in Fig. 2 as an example. The interconnections are labeled 1, 2, $\cdots$, and 8. The numbers in parentheses attached to labels 1 to 6 are the depth numbers. The arrows indicate the faulty signal flow. We start the fault tracing at the interconnection 6. When the fault tracing

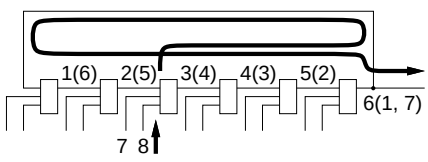

Fig. 2. Example of the second tracing for the feedback lines. In the second tracing, only the interconnections 3, 2, 7 and 8 are measured.

proceeds to the interconnection 6 through the interconnections 5, 4, $\cdots$, and 1, a *MEASURED LABEL* has already been placed on the interconnection 6. Thus we know that the feedback line is formed through the interconnection 6.

With the aid of the binary search algorithm, the interconnection 3 which is centered on the tracing path is tested at first. The measurement period of the interconnection 6 is advanced by the signal propagation delay between the interconnections 3 and 6 which can be calculated from the data recorded in the *MEASURED LABEL*s. When the interconnection 3 is faulty, again the binary search is applied to a left half of the path. Divisions are repeated until the cell with an unmeasured faulty input 8 is specified. From the interconnection 8, the usual back tracing is performed. In this fault tracing method, measurements are carried out only $\log_2(D_2 - D_1)$ times, where D_1 and D_2 are the depth numbers put on the interconnection 6 at the first time and at the second time, respectively.

C. Global flow of the hierarchical fault tracing

Fig. 3 shows the global flow of the hierarchical fault tracing in the sequential circuit. Circles labeled A, B, C and D indicate the processes. Arrows labeled a, b, $\cdots$, and g are the conditions to turn to another process. At first, go to the process A in which a back tracing is carried out. This process includes sub processes of selection of the upstream interconnection, decision of the measurement period, practice of the waveform measurement and the waveform comparison, and descent of the hierarchy.

When the fault tracing meets an unmeasurable interconnection [5] (the condition a), go to the process B. In this process, measurable upstream interconnections are tested. If a faulty waveform is acquired (the condition b), then return to the process A. Otherwise (the condition c), go to the process C where a forward tracing to the last faulty interconnection is carried out. Then return to the process A (the condition d). When the fault tracing meets the interconnection on which a faulty waveform has been already acquired (the condition e), go to the process D where the second tracing at the same circuit node is performed. When the second tracing process is finished (the condition f), then return to the process A. When good waveforms are acquired on all upstream interconnections in the transistor-level (the condition g), then the fault tracing is completed.

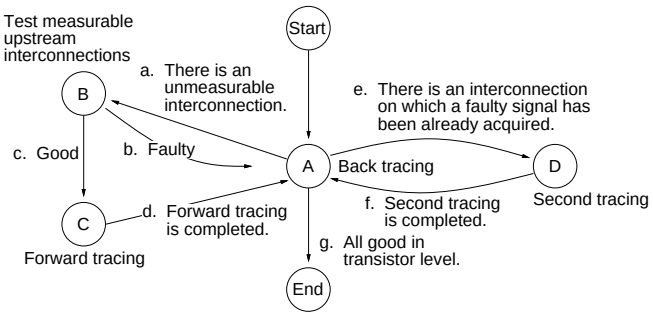

Fig. 3. The global flow of the hierarchical fault tracing in the sequential circuit. Circles labeled A, B, C and D, indicate the processes. Arrows labeled a, b, $\cdots$, and g are the conditions to turn to another process.

IV. Application

We have implemented our expanded hierarchical fault tracing method in the C language on a UNIX workstation where the layout data in GDS-II format is available. We applied the method to the CAD layouts of randomly selected five circuits in the ISCAS'89 sequential benchmark circuit set[10] as shown in Table 1. The layouts were made by using the CMOS3 cell library [11]. On each layout, we randomly generated a performance fault [4] ten times. The assumed performance fault is that the signal delay lies outside of a specified range due to the insufficient drive capability of the MOS transistor. The test pattern sequence was generated to propagate the faulty signal due to the performance fault to primary outputs.

Figs. 4(a) and (b) show the CAD layout and the gate-level schematic of the sequential benchmark circuit s27. Fig. 5 shows the transistor- level schematic of the cell "L" included in Fig. 4(b). The transistor M564 is assumed faulty. The expanded hierarchical fault tracing method was applied to the CAD layout shown in Fig. 4(a). The executed result is shown in Table 1. Each "measure" row in the table contains the probing point number, the interconnection number, the measurement period and the waveform comparison result. In case of faulty, the period in which the faulty signal is detected is also recorded as [208, 228], where the unit of time is ns. The row 10 shows the suggested faulty cell. Rows from 13 to 15 suggest the names of a possible faulty interconnection and MOS transistors. These results are shown in the crosshatched region

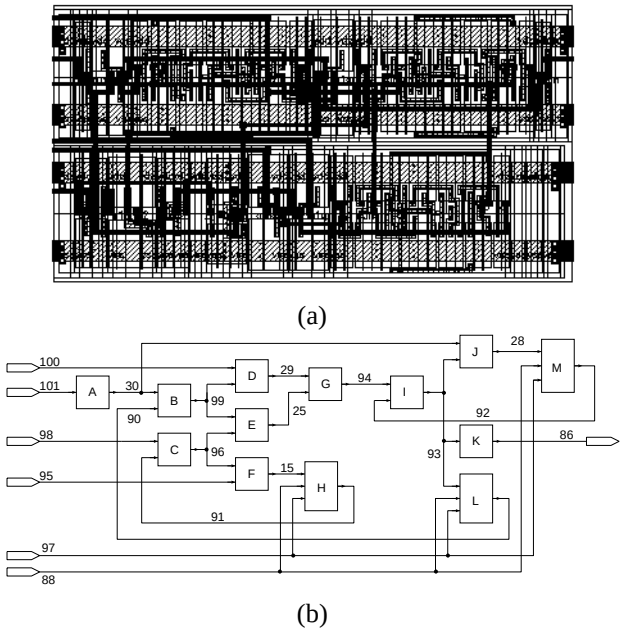

Fig. 4. CAD layout (a) and the gate-level schematic (b) of the sequential benchmark circuit s27.

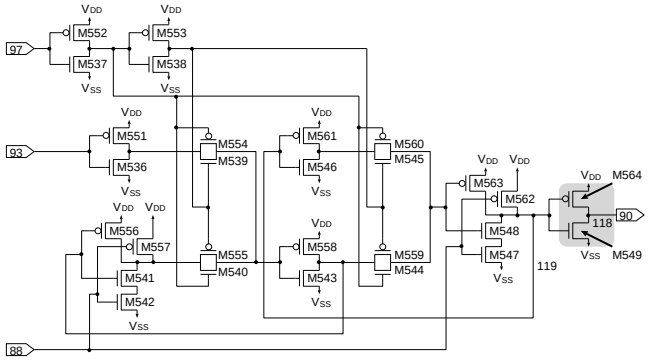

Fig. 5. Transistor-level schematic of the cell "L" included in Fig. 4(b). The transistor M564 is assumed faulty.

TABLE I
EXECUTED RESULT.

```
 1  Initial Fault Phase: [210-230]
 2  measure[1]:93, Phase: [110-230], Faulty: [208-228]
 3  measure[2]:92, Phase: [108-228], Good
 4  measure[3]:94, Phase: [108-228], Faulty: [206-226]
 5  measure[4]:25, Phase: [106-226], Faulty: [204-224]
 6  measure[5]:99, Phase: [104-224], Faulty: [202-222]
 7  measure[6]:90, Phase: [102-222], Faulty: [200-220]
 8  measure[7]:88, Phase: [100-220], Good
 9  measure[8]:97, Phase: [100-220], Good
10  There is a fault in the cell: L
11  measure[9]:119, Phase: [100-220], Good
12  There may be a fault around the interconnection 118
13  interconnection(118)
14  MOS-FET(564)
15  MOS-FET(549)
```

in Fig. 5. Thus, the intended fault was correctly specified.

We compared the number of probing points of our method with that of the guided-probe method where circuit logical functions are extracted from the CAD layout data [12] and then the guided-probe testing [9] is executed. In the guided-probe testing, to reduce the number of probing points, the speed-up technique is adopted where we probe first the control lines among the inputs that can influence the output with errors. The difference of the number of probing points of the present method from that of the guided-probe method is shown in Table 2. MIN, MAX, and AVE means the minimum, maximum and average differences for ten randomly generated performance faults, respectively. For reference, the average number of probing points (AVE probing points) of our fault tracing method is also described. It is seen that the number of probing points of the expanded hierarchical fault tracing method is reduced a little as compared with that of the guided-probe method. Thus the present method is superior to the guided-probe method when only CAD layout data is available, since our method does not need to recognize the circuit logical functions.

V. CONCLUSIONS

We expand a previous hierarchical fault tracing method for combinational circuits which requires only CAD layout data in the CAD-linked electron beam test system as applicable to sequential circuits. The characteristics in the method remains

TABLE II
DIFFERENCE OF THE NUMBER OF PROBING POINTS OF THE EXPANDED HIERARCHICAL FAULT TRACING METHOD FROM THAT OF THE GUIDED-PROBE METHOD.

circuit name	#trans.	difference			AVE probing points
		MIN	MAX	AVE	
S27	136	-4	0	-1.1	6.6
S208	676	-4	3	0	14.9
S510	1158	-5	4	-0.6	10.9
S953	2530	-4	3	-0.2	9.0
S5378	14276	-5	5	-1.0	17.2

unchanged that allows us to trace a fault hierarchically from the top level cell to the lowest primitive cell and from the primitive cell to the transistor-level circuit in a consistent manner independent of circuit functions The applied results to the CAD layouts of five sequential CMOS benchmark circuits show the superiority of our expanded hierarchical fault tracing method as compared with the guided-probe method where circuit logical functions are extracted from the CAD layout data and then the guided-probe testing is executed.

REFERENCES

[1] K. Ura and H. Fujioka, "Electron Beam Testing," *Advances in Electronics and Electron Physics*, ed. P. W. Hawkes, Academic Press, New York, Vol. 73, pp. 233-317, 1989.

[2] A. C. Noble, "IDA: A Tool for Computer-Aided Failure Analysis," *in Proc. of 1992 international Test Conference*, pp. 848-853, 1992.

[3] N. Yamaguchi, T. Sakamoto, H. Nishioka, T. Majima, T. Satou, H. Shinada, H. Todokoro and O. Yamada, "E-Beam Fault Diagnosis System for Logic VLSIs," *Microelectronic Engineering*, Vol. 16, pp. 121-128, 1992.

[4] S. W. Director, W. Maly and A. J. Strojwas, *VLSI Design of Manufacturing : Yield Enhancement*, Kluwer Academic Publishers, Boston, 1990.

[5] K. Miura, K. Nakamae and H. Fujioka, "Automatic Tracing of Transistor-Level Performance Faults with CAD-linked Electron Beam Test System," *IEICE Transactions on Fundamentals*, Vol. E77-A, No. 3, pp. 539-545, Mar. 1994.

[6] K. Miura, K. Nakamae and H. Fujioka, "Automatic Transistor-Level Performance Fault Tracing by Successive Circuit Extraction from CAD Layout Data for VLSI in the CAD-Linked EB Test System," *IEICE Transactions on Electronics*, Vol. E78-C, No. 11, pp. 1607-1617, 1995.

[7] K. Miura, K. Nakamae and H. Fujioka, "Hierarchical VLSI Fault Tracing by Successive Circuit Extraction from CAD Layout Data in the EB Test System," *Journal of Electronic Testing*, in Press.

[8] R. Rajsuman, *Digital Hardware Testing: Transistor-Level Fault Modeling and Testing*, Artech House (Boston), Chapter 8, 1992.

[9] M. Abramovici, M. A. Breuer, A. D. Friedman, *Digital Systems Testing and Testable Design*, IEEE PRESS (New York), Chapter 12, 1990.

[10] F. Brglez , D. Bryan, K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," *in Proc. ISCAS '89*, pp. 1929-1934, 1985.

[11] K. Nakamae, K. Miura, H. Fujioka, "LSI Testing with CAD-Linked Electron Beam Test System," *Microelectronic Engineering*, vol. 31, pp. 319-330, 1996.