An Approach to Reduce Over-testing of Path Delay Faults in Data Paths Using RT-level information

Yuki Yoshikawa, Satoshi Ohtake and Hideo Fujiwara
Graduate School of Information Science, Nara Institute of Science and Technology
Kansai Science City 630-0192, Japan
Email:{*yuuki-y, ohtake, fujiwara*}@*is.naist.jp*

Abstract

In this paper, we present an approach to reduce over-testing of path delay faults (PDFs). To reduce test generation complexity, design-for-testability (DFT) techniques are used in order to make a large number of untestable PDFs in a circuit testable. For a circuit modified by a DFT technique, even if faults corresponding to untestable PDFs in the original circuit are excluded from the fault list targeted by test generation, the faults may still be tested by the generated tests. In our previous work, we proposed a method which identifies a subset of sequentially untestable PDFs for circuits described at register-transfer level (RTL). In this paper, we show that, for a circuit modified by some DFT technique, tests generated by our test generation scheme do not test any fault corresponding to an untestable PDF, which are identified by our method, in the original circuit.

1. Introduction

The speed of VLSI circuits has increased in recent years. High speed circuits need delay testing to verify that a given logic operates correctly at the desired clock speed. There are several delay fault models, path delay fault (PDF) model, transition fault model, segment delay fault model and so on [1]. In this paper, we target the PDF model. PDF can model a defect as cumulative propagation delays along a path in a circuit, hence can model the delay between two flip-flops (FFs). To test a PDF, a consecutive vectors (two-pattern test) are required for FFs which are the starting points of the target path and other related paths.

Consider a sequential circuit, PDFs in the circuit are testable PDFs or untestable PDFs. The untestable PDFs are classified into combinationally untestable PDFs and sequentially untestable PDFs. In general, sequential test generation takes a lot of time to identify PDFs as sequentially untestable. Hence to reduce test generation complexity, design-for-testability (DFT) techniques are used in order to make a large number of sequentially untestable PDFs in a circuit testable [8]. However this induces over-testing problem because faults corresponding to sequentially untestable PDFs, which are never activated in normal operation, in the original circuit are tested. Over-testing causes yield loss. To reduce over-testing, sequentially untestable PDFs have to be removed from target of testing. Therefore not only DFT techniques but also techniques to identify sequentially untestable PDFs are important.

In terms of combinationally untestable PDFs, in [10], a method to select testable PDFs efficiently has been proposed. We also utilize the method to extract combinationally untestable PDFs. The work in [9] is to find sequentially untestable PDFs by using time expansion model at gate-level. For small scale circuits the method succeeds to identify sequentially untestable PDFs within practical time. However it is conceivable that the method takes long CPU time for large scale circuits. In our previous work [7], we extracted a large number of sequentially untestable PDFs at RTL in reasonable time. We referred to the extracted untestable PDFs as control-dependent untestable paths (CUPs). CUPs are identified by analyzing control signals such as load-enable signals of registers and select signals of multiplexers (MUXs). Gate-level paths between two registers are regarded as a bundle of paths at RTL, which is referred to as an *RTL path*. An RTL path is a path passing through only combinational module, which starts at a primary input (PI) or a register and ends at a register or a primary output (PO). Since the number of RTL paths in a circuit is much smaller compared to the number of gate-level paths, the extraction of CUPs is very quick.

Scan design is widely accepted in industry as an effective DFT method. We can enhance two-pattern testability for FFs by using standard scan technique such as *skewed-load testing* [2] and *broadside testing* [3], or enhanced scan technique [4, 5]. However these DFT techniques cause over-testing because sequentially untestable

PDFs are changed into testable PDFs. In order to reduce over-testing for a circuit modified by scan design, we remove sequentially untestable PDFs identified as CUPs from fault list targeted by test generation. However two-pattern tests generated to test PDFs in the fault list still may test the removed PDFs because the generated patterns happen to activate the removed PDFs. We propose a method to guarantee that sequentially untestable PDFs judged as CUPs are never tested by utilizing the concept of single-port-change (SPC) two-pattern testability proposed in our previous work [7]. A port means the output port of a register and has bit width. An SPC two-pattern test can launches transitions at the starting point of paths corresponding to a port and sets stable for the other ports. In this paper, to compare reduction of over-testing for SPC two-pattern tests with that for any two-pattern tests, we adopt enhanced scan technique as DFT.

Experimental results show that our proposed method never activate PDFs which are identified as CUPs. Moreover, we show that if we generate any two-pattern tests without any constraint for the same fault list, about 60 % of PDFs which are identified as CUPs are tested.

2. Target circuits and faults

An RTL design generally consists of a controller, represented by state transition graph, and a data path, represented by hardware elements such as registers, multplexers and operational modules. They are connected each other by control signal lines and status signal lines. A controller controls load-enable signals of registers and select signals of MUXs. On the other hand, the status signals from a data path determine a next state and/or output values of the controller. Our target circuit is a data path. Note that information of control signals from a controller is analyzed in order to extract control-dependent untestable paths. Our target faults are path delay faults in the data path. Note that path delay faults on control signal lines and status signal lines are not targeted in this paper.

3. Control-dependent untestable path

From an RTL description of a circuit, we can obtain information about state transitions of the controller and the control signals such as the load-enable signals of registers and the select signals of MUXs. The select signals of MUXs decide paths to transfer data from a register to next register. The load-enable signals decide the timing of data transfer between registers. We identify RTL paths as control-dependent untestable paths(CUPs) by analyzing the information.

Let P be a set of RTL paths in a data path and now we consider whether $p \in P$ is CUP or not. Let R_s be the register which is the starting point of p, and let R_e be the register which is the ending point of p. Let C_{Rs} and C_{Re} be load enable signals of registers R_s and R_e, respectively. If the load enable signal of a register is equal to '1', the register loads a value, otherwise, holds its value. Note that if the register does not have *hold* function, we assume that a load enable signal line is connected to the register, and the value of that signal is always '1'. If the starting point of p is a PI or the ending point of p is a PO, the PI or the PO is treated as a register with no *hold* function. Let M_i and C_{M_i} $(1 \leq i \leq n)$ be a MUX on p and its select signal, respectively. When M_i selects the input on p, the value of the select signal is denoted as $C_{M_i} = p_{M_i}$. For example, we assume that p is the RTL path R_1-M_1-$Adder$-M_2-R_3 in Figure 1. When M_1 and M_2 select p, $p_{M_1} = 0$ and $p_{M_2} = 0$. Let S_i and S_j be states of the controller. S_i and S_j is said to be consecutive if there exists a direct transition from S_i to S_j. Let $C_{M_k} = (a_i, a_j)$ be a select signal pair of M_k at S_i and S_j.

Definition [7]: An RTL path p is control-dependent untestable path (CUP) if either of following two conditions is satisfied for any consecutive two states.

1. $C_{Rs} = (0,-) \vee C_{Re} = (-,0)$ $\quad -$: don't care
2. $\vee_{i=1}^{n}\{C_{M_i} \neq (-, p_{M_i})\}$

$C_{Rs} = (0,-)$ shows Rs does not load a value at the first state of consecutive two states and the second state is don't care. $\vee_{i=1}^{n}\{C_{M_i} \neq (-, p_{M_i})\}$ means that there exist at least one MUX on p which are not selecting p at the second state.

Theorem [7]: All the gate-level paths corresponding to an RTL path p are non-robust untestable if p is CUP.

proof [7]: For the first condition of Definition [7], R_s does not launch a transition at S_i, or R_e does not capture the response at S_j even if R_s launched a transition at S_i. For the second condition, p is not selected at S_j and this prevents propagation of transitions from R_s to R_e. Therefore, p is non-robust untestable. □

We consider the RTL path R_1-M_1-$Adder$-M_2-R_3 and R_1-M_1-$Adder$-M_2-R_4 in Figure 1 as an example. First we consider R_1-M_1-$Adder$-M_2-R_3. R_3 does not have hold function. We assume that R_3 has a load-enable signal line, and the signal is always '1'. The controller has three consecutive two states (S_0,S_1), (S_1,S_2) and (S_2,S_0). For (S_0,S_1) and (S_1,S_2), the first condition of Definition [7] are satisfied because C_{R_1} is zero at S_0 and S_1. For (S_2,S_0), both the first condition and the second condition are not satisfied. Hence R_1-M_1-$Adder$-M_2-R_3 are not CUP.

Next we consider R_1-M_1-$Adder$-M_2-R_4. For (S_0,S_1) and (S_1,S_2), the first condition are satisfied. For (S_2,S_0), the first condition are not satisfied but the second condition are satisfied. Therefore R_1-M_1-$Adder$-M_2-R_4 is CUP.

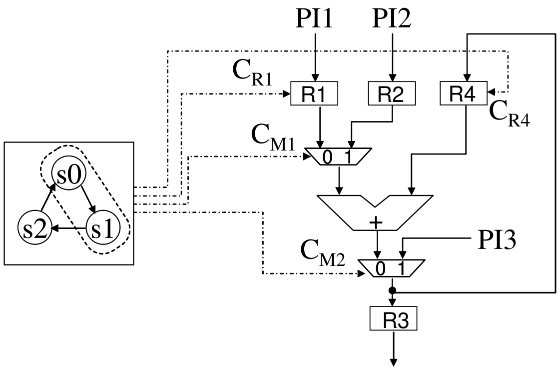

Figure 1. An RTL circuit.

Table 1. State transition table.

	s0	s1	s2
C_{R_1}	0	0	1
C_{R_4}	1	1	0
C_{M_1}	0	1	1
C_{M_2}	0	0	1

4. Reduction of over-testing

We expect reduction of over-testing by removing sequentially untestable (PDFs) corresponding to CUPs from a fault list targeted by test generation. For a circuit modified by enhanced-scan technique, any two-pattern test can be applied to FFs which are the starting points of paths. Hence we usually generate arbitrary two-pattern tests for the combinational logic block without considering any input constraint. However such two-pattern tests still may test the removed untestable PDFs because transitions may be launched at the starting points of the PDFs.

We reduce over-testing by using the concept of single-port-change two-pattern testability.

4.1. Single-port-change two-pattern tests

In our previous work [7], we have introduced single-port-change(SPC) two-pattern tests. SPC two-pattern tests are defined at RTL. Here a combinational logic block which forms an input cone to a register is considered at RTL. Suppose an RTL path passing through the cone is a target of testing. The target RTL path is referred to as *on-path*. An RTL path which supports propagation of transitions on the on-path is called *off-path*. For the input port of an operational module on an on-path, one of the RTL path passing through the other input port is an off-path (See the left picture of Figure 2). If an on-path passes through some

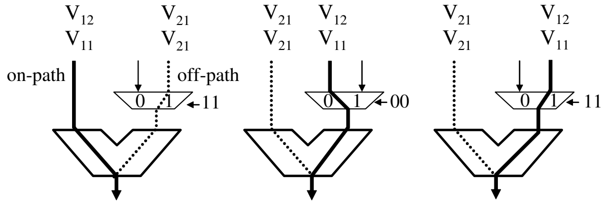

Figure 2. SPC two-pattern test.

operational modules, each operational module has an off-path. An SPC two-pattern test is a pair of two consecutive vectors which launches transitions at only the starting point of an on-path, and sets stable two consecutive vectors for the starting point of an off-path (or off-paths if there exist off-paths).

The select signal of each MUX is selecting an on-path or an off-path during test application as shown in Figure 2. In our previous method [6], we showed that while the select signal of a MUX is fixed, propagation of signals from the selected input port to the output port is independent of signals at the other input port. Therefore the on-path is testable if SPC two-pattern tests can be applied to starting point of the on-path and the off-path. Test generation of SPC two-pattern tests for a combinational logic block are performed by using a combinational test generation algorithm with constraints. With regard to quality of SPC two-pattern tests, we showed that there always exists an SPC robust (resp. non robust) test for a robust (resp. non-robust) testable path delay fault. For functionally sensitizable path delay faults, SPC two-pattern tests can test a subset of functionally sensitizable path delay faults. The detail of test generation and quality of SPC two-pattern tests are shown in [7].

4.2. test generation and test application

Here we first explain a case where sequentially untestable PDFs removed from a fault list are tested by generating two-pattern tests. Figure 3 shows a simple RTL structure and there are two RTL paths, $R1$-$Adder$-$R3$ and $R2$-$Adder$-$R3$. We assume that $R1$-$Adder$-$R3$ is not CUP and $R2$-$Adder$-$R3$ is identified as CUP. Then gate-level PDFs corresponding to $R2$-$Adder$-$R3$ are removed from a fault list. For the fault list, we usually generate a set of two-pattern tests without input constraints i.e. any two-pattern tests can be generated. Let the first vectors be V_{11} and V_{12}. Let the second vectors be V_{21} and V_{22}. During test application, transitions may be launched at the starting point of $R2$-$Adder$-$R3$ because V_{21} and V_{22} can be different vectors. Therefore unexpected PDFs may be tested.

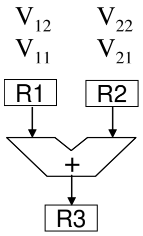

Figure 3. A simple RTL structure.

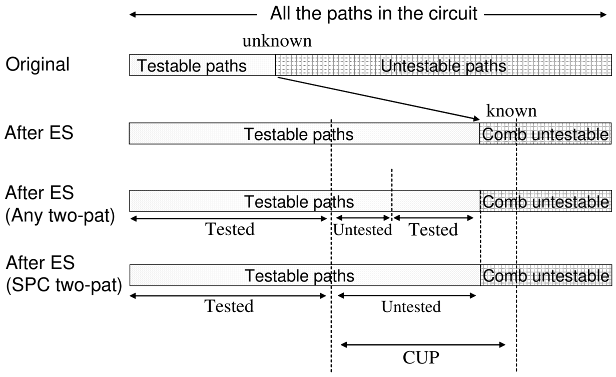

Figure 4. The ratio of CUPs to tested paths.

Next, we show that the removed untestable PDFs are never tested by generating SPC two-pattern tests. We consider the same fault list as the above instance. For the fault list, we generate a set of SPC two-pattern tests. Let the first vectors be V_{11} and V_{12}. Let the second vectors be V_{21} and V_{22}. Then During test application, transitions are never launched at the starting point of $R2$-$Adder$-$R3$ because V_{21} and V_{22} are the same vectors. Therefore the removed PDFs are never tested.

In this paper, we adopt enhanced scan design. Note that if we consider another DFT techniques such as non-scan DFT techniques, broadside and skewed-load techniques with standard scan, we have to consider incidental activation during justification of test patterns.

5. Experimental results

In this section, we evaluate the effectiveness of extraction of CUPs and the ratio of reduction of over-testing by using the concept of SPC two-pattern testability. In this experimental results, we adopt enhanced scan design because we evaluate the difference between the ratio of reduction of over-testing by using SPC two-pattern tests and that by using any two-pattern tests. The circuit characteristics of RTL benchmarks used in the experiments are shown in Table 2. Paulin, LWF, Tseng, JWF are widely used benchmark circuits. MPEG is more practical and larger circuit provided by industry. Generally if bit width of signal lines in a circuit increase, the circuit scale becomes large, i.e., the number of paths in the circuit exponentially increases. In this paper, we need the total number of paths to analyze the ratio of reduction of over-testing. Hence we use 8 bit width benchmarks. However the method to identify CUPs and the reduction of over-testing by using SPC two-pattern tests are independent of the bit width of signal lines and we can also apply them to larger circuits.

Table 2. Circuit characteristics.

Circuit	BW	#PIs	# POs	# REGs	# RTL paths	# Gate paths
Paulin	8	2	2	7	29	58,788
LWF	8	2	2	5	19	1,383
Tseng	8	3	3	6	20	8,266
JWF	8	5	5	14	153	13,000
MPEG	8	7	16	241	651	-

Figure 4 shows that the ratio of PDFs corresponding to CUPs to tested PDFs among CUPs as an example. Since for all the PDFs in an original circuit, we cannot perform sequential test generation within practical time. Hence we do not know the ratio of testable PDFs to untestable PDFs in the circuit exactly. However we can know the ratio of testable PDFs to combinationally untestable PDFs after modification by enhanced scan design. The ratio of testable PDFs increases because sequentially untestable PDFs become testable. CUPs are a subset of sequentially untestable PDFs in the original circuit. Note that the sequentially untestable PDFs judged as CUPs may overlap with combinationally untestable PDFs. If we generate any two-pattern tests, some PDFs corresponding to CUPs are tested. However our proposed method never test all the PDFs corresponding to CUPs.

Table 3 shows analysis of CUPs and reduction of over-testing. The second column is the number of RTL paths judged as CUPs in benchmarks. The third column is the number of gate-level paths corresponding to CUPs. The forth column is the ratio of the number of paths identified as CUPs to the number of total paths. The fifth column is the number of PDFs corresponding to CUPs except for combinational untestable PDFs. PDFs include both rising transition faults and falling transition faults. The sixth is CPU time to extract CUPs. The seventh and the last column are the ratio of the number of testable PDFs corresponding to CUPs to untested testable PDFs (This is the ratio of Untested to CUP in Figure 4). For JWF, 80 % of sequentially untestable PDFs are judged as CUPs within 0.1 second. For Paulin and Tseng, about 50 % of sequentially

Table 3. Analysis of CUPs and reduction of over-testing.

Circuit	#CUPs at RTL	#Gate-level paths(CUPs)	Ratio of CUPs	#PDFs except for C_untestable (CUPs)	CPU time (sec)	Over-tested any two-pat	Over-tested SPC two-pat
Paulin	12	32,715	55.7%	48,213	0.046	26,655 (55.2%)	0 (0%)
LWF	3	268	19.3%	536	0.030	339 (63.2%)	0 (0%)
Tseng	6	3,989	48.3%	5,779	0.031	3,413 (59.1%)	0 (0%)
JWF	117	10,486	80.7%	20,630	0.046	12,543 (60.8%)	0 (0%)
MPEG	32	256	-	512	0.062	-	-

untestable PDFs are identified as CUPs within 0.1 second. For only 100 PDFs in Paulin, if we perform sequential test generation, it takes 498 second to judge 51 PDFs as sequentially untestable. In terms of reduction of over-testing, if we generate SPC two-pattern tests for fault lists removed sequentially untestable PDFs, the removed PDFs are never tested. However if we generate any two-pattern tests for the same fault list, about 60 % of the removed PDFs is tested (over-testing). For MPEG, we just examined that time required for identifying CUPs.

6. Conclusion

In our previous work, we have proposed a method to extract a large number of sequentially untestable PDFs at RTL. However even if the untestable path delay faults are removed from a fault list targeted by test generation, the generated test patterns incidentally test the removed faults. In this paper, we have proposed a method which never test PDFs identified as control-dependent untestable paths by using single-port-change (SPC) two-pattern testability. In experimental result, we showed that SPC two-pattern tests can reduce about 60 % of over-testing compared to that for any two-patten tests.

acknowledgment The authors would like to thank Dr. Tomokazu Yoneda (Nara Institute of Science and Technology) for their valuable discussion and their cooperation. This work was supported in part by 21st Century COE (Center of Excellence) Program (Ubiquitous Networked Media Computing) and in part by Japan Society for the Promotion of Science (JSPS) under Grants-in-Aid for Scientific Research B(2)(No. 15300018) and for Young Scientists (B) (No.17700062).

References

[1] Angela Krstic and Kwang-Ting(Tim)Cheng, *Delay Fault Testing for VLSI Circuits*, Kluwer Academic Publishers, 1998.

[2] J. Savir and S. Patel, “Scan-based transition test,” IEEE Trans. on CAD, Vol. 12, No.8, pp. 1232-1241, Aug. 1993.

[3] J. Savir and S. Patel, “Broad-side delay test,” IEEE Trans. on CAD, Vol. 13, No.8, pp. 1057-1064, Aug. 1994.

[4] B. I. Devadas and G. E. Stong, “Design for testability : Using scanpath techniques for path-delay test and measurement,” Proceeding of International Test Conf, pp. 365-374, 1991.

[5] Tapan J. Chakraborty, Vishwani D. Agrawal, Michael L. Bushnell : “Design for testability for path delay faults in sequential circuits,” Proc. DAC 93, pp. 453-457, 1993.

[6] Md. Altaf-Ul-Amin, S. Ohtake and H. Fujiwara, “Design for hierarchical two-pattern testability of data paths,” IEICE Trans. on Information and Systems, Vol. E85-D, No. 6, pp. 975-984, Jun. 2002.

[7] Y. Yoshikawa, S. Ohtake, M. Inoue and H. Fujiwara, “Design for Testability Based on Single-Port-Change Delay Testing for Data paths,” Proc. Asian Test Symp., (be presented), 2005.

[8] T. J. Chakraborty, V. D. Agrawal, M. L. Bushnell, “Design for Testability for Path Delay Faults in Sequential Circuits,” Proc. 30th ACM/IEEE Design Automation Conference. pp. 453-457, 1993.

[9] A. Krstic. S. T. Chakradhar, K. T. Cheng, “Testable Path Delay Fault Cover for Sequential Circuits,” Proc. European Design Automation Conference. pp. 220-226, Sep. 1996

[10] M. Reddy, I. Pomeranz, S. Kajihara, “On Selecting Testable Paths in Scan Designs,” Proc. European Test Workchop. 2002.