IEEE/ACM International Conference on Computer-Aided Design (ICCAD'07), pp. 418-423, Nov. 2007.

Efficient Path Delay Test Generation Based on Stuck-at Test Generation Using Checker Circuitry

Tsuyoshi Iwagaki†, Satoshi Ohtake‡, Mineo Kaneko† and Hideo Fujiwara‡
†School of Information Science, Japan Advanced Institute of Science and Technology (JAIST)
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
E-mail: {iwagaki, mkaneko}@jaist.ac.jp
‡Graduate School of Information Science, Nara Institute of Science and Technology (NAIST)
Kansai Science City 630-0192, Japan
E-mail: {ohtake, fujiwara}@is.naist.jp

***Abstract*— This paper proposes an approach to non-robust and functionally sensitizable path delay test generation through stuck-at test generation. In this approach, to generate two-pattern tests for path delay faults in a combinational circuit, checker circuitry is constructed which is composed of logic gates corresponding to the mandatory assignments for detecting the faults. This checker circuitry allows us to use any existing combinational stuck-at test generation tool. Since today's stuck-at test generation tools reach a mature level, the proposed approach can efficiently solve the path delay test generation problem for combinational circuits. Experimental results show that the approach can speed up path delay test generation and can improve fault efficiency. This paper also discusses how a scan circuit and the issues of over-testing and test power are handled in the proposed test generation framework.**

I. Introduction

Testing of path delay faults (PDFs) as well as stuck-at faults (SAFs) is an essential task for today's high speed circuits. Automatic test pattern generation (ATPG) techniques for SAFs have been well-studied over the decades, and those techniques for combinational circuits have reached a mature level. For PDFs, some efficient techniques have also been investigated [1]–[4]. These techniques adopted new multi-valued logic systems for detecting PDFs in a combinational circuit, and considered algorithms adapted to the logic systems. In this way, on the one hand, there have been methods using new multi-valued logic systems. But, on the other hand, there have been methods using existing ATPG techniques for SAFs to handle PDFs [5]–[8]. All of the methods of [5]–[8] is based on a test generation framework using a modified circuit. In this framework, to generate two-pattern tests for PDFs in a combinational circuit, the circuit is first transformed into a different one only during test generation. Then, a stuck-at test generation algorithm is applied to the corresponding SAFs in the transformed circuit. The generated test patterns for the corresponding SAFs are finally transformed into the two-pattern tests for the original PDFs. This framework can efficiently solve the problem of path delay test generation for a combinational circuit by using a mature ATPG tool for SAFs. Our work is also based on this framework. Next, we review each of the above related works further.

Which class of PDFs to be targeted and the circuit size after transformation are main concerns in the above test generation framework. A modified circuit based on a leaf-dag, in which a fanout and an inverter are only permitted at the primary inputs and the output of an inverter is not allowed to have a fanout, is used in [5]. The circuit size depends on the number of PDFs targeted. Although this method can efficiently handle the problem of path delay test generation for a combinational circuit, only robust PDFs [9] are targeted. There has been a method to handle larger classes of PDFs: singly-testable (ST) and multiply-testable (MT) PDFs [6]. Since this method can handle ST and MT PDFs, a large part of delay defects can be covered compared to [5]. However, this method is not applicable to large circuits because the size of the modified circuit used in the method depends on the total number of paths in a given combinational circuit. Recently, two methods to overcome the disadvantages of [5] and [6] have been proposed in [7] and [8]. Both methods are capable of handling non-robust (NR) PDFs [9], and the size of the modified circuits used in the methods depends on the number of PDFs targeted rather than the total number of paths in a given combinational circuit. Although, indeed, the disadvantages of [5] and [6] are alleviated in [7] and [8], other testable classes of PDFs such as functionally sensitizable (FS) [9] PDFs are not handled yet.

This paper presents a test generation method for NR and FS PDFs in a combinational circuit. In the proposed method, to generate two-pattern tests for PDFs in a combinational circuit, checker circuitry is constructed which is composed of logic gates corresponding to the mandatory assignments for detecting the faults. Like the other approaches of [5]–[8], this checker circuitry allows us to use any existing combinational stuck-at test generation tool. The size of the modified circuit used in our method depends on the number of PDFs targeted during test generation. This feature enables us to handle a large circuit in which the total number of paths is usually very large. Furthermore, our method can handle a large class of PDFs, i.e., FS PDFs, compared with [7] and [8]. Through experimental results, we show that our method can accelerate path delay test generation and can improve fault efficiency

1-4244-1382-6/07/$25.00 ©2007 IEEE

TABLE I

OFF-INPUT CONDITIONS FOR DETECTION

On-input transitions	Off-input conditions	
	NR	FS
$cv \rightarrow ncv$	$x \rightarrow ncv$	
$ncv \rightarrow cv$	$x \rightarrow ncv$	$x \rightarrow ncv$ or $ncv \rightarrow cv$

compared with an ordinary method. In this paper, we also discuss some applications of our method to deal with the over-testing and test power problems.

II. PRELIMINARIES

This paper targets non-robust (NR) and functionally sensitizable (FS) path delay faults (PDFs) in a combinational circuit. Some definitions and notations are described below.

In a combinational circuit C, a path p is defined as a sequence of gates. The starting (resp. ending) point of p corresponds to a primary input (resp. primary output) of C. There exist two types of PDFs on p depending on the direction of the transition at the starting point of p, i.e., rising and falling PDFs. A controlling value, which is denoted as *cv*, of a gate is a value at an input of the gate that determines the value of the gate output regardless the values of the other inputs. A non-controlling value, which is denoted as *ncv*, of a gate is the complement value of its controlling value. A gate input is referred to as an on-input of p if it is on p. A gate input is referred to as an off-input of p if it is an input to a gate on p but it is not an on-input. Whether a PDF is NR or FS is determined according to whether the mandatory values can be assigned to all the off-inputs. Table I lists the off-input conditions for NR PDFs and FS PDFs, where x is an unknown value. In notation $v_1 \rightarrow v_2$, v_1 (resp. v_2) represents the first vector (resp. second vector) of a vector pair on a gate input.

III. PROPOSED TEST GENERATION FRAMEWORK

This section discusses a test generation framework for NR and FS PDFs using checker circuitry. First, we present a test generation model based on checker circuitry. Then, we propose a test generation procedure using the model, and prove the correctness of the proposed method.

A. Test Generation Model

Given a combinational circuit C and a set of PDFs F in C, a test generation model C^F_{TGM} for F (PDTGM) is constructed as follows.

(1) Duplicate C as C_1 and C_2.

(2) Insert additional signal lines into all the starting points of the paths corresponding to F in C_1 and C_2.

(3) Insert additional signal lines into all the off-inputs of the gates on the paths in C_1 and C_2.

(4) Create a checker circuit C^F_{C} according to the transitions at the starting points and the off-input conditions.

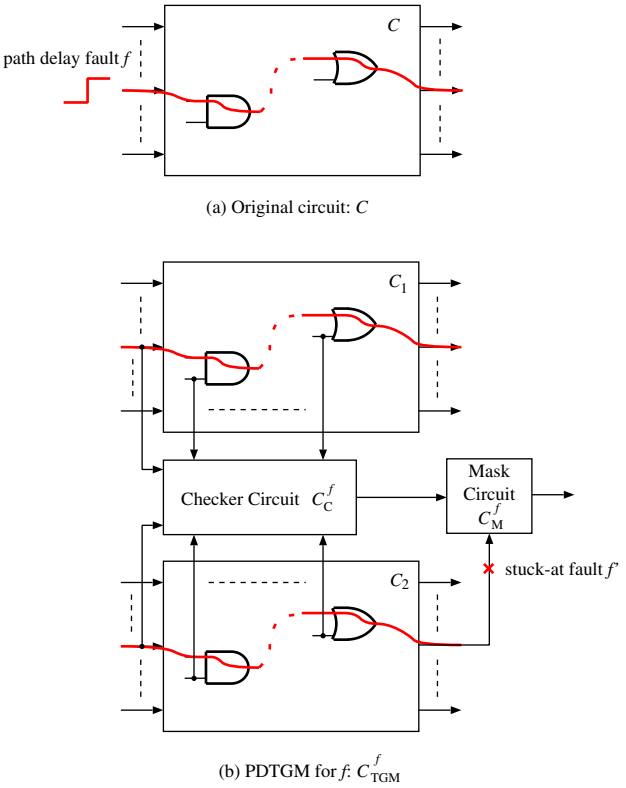


Fig. 1. Overview of a test generation model for PDFs (PDTGM)

(5) Create a mask circuit C^F_{M}.

(6) Connect all the additional signal lines to C^F_{C}, and connect the outputs of C^F_{C} and the ending points of the paths in C_2 to C^F_{M}.

Figure 1 shows the overview of a PDTGM for one PDF f. In this figure, C_1 and C_2 are used to generate the first vector and the second vector of two-pattern tests, respectively. The checker circuit in this figure outputs '1' only if the condition for the primary on-input and all the off-input conditions are satisfied. PDF f in the original circuit corresponds to the stuck-at fault (SAF) f' on the bottom input of the mask circuit. The mask circuit blocks the fault effect of f' except when the checker circuit outputs '1.' If f is a rising (resp. falling) PDF and the number of inverters on the path with f is even, f' is a stuck-at 0 (resp. 1) fault. In contrast, if f is a rising (resp. falling) PDF and the number of inverters on the path with f is odd, f' is a stuck-at 1 (resp. 0) fault. In (4) (resp. (5)) of the above procedure, C^F_{C} (resp. C^F_{M}) includes $|F|$ sub checker circuits (resp. mask circuits) and has $|F|$ outputs, where $|F|$ denotes the number of PDFs in F. In other words, in Figure 1, $|F|$ pairs of a sub checker circuit and a sub mask circuit must exist in parallel. If NR PDFs (resp. FS PDFs) are targeted, the conditions of "NR" (resp. "FS") shown in Table I are used

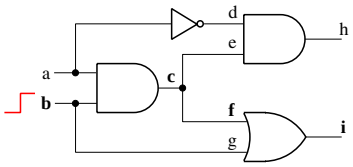


Fig. 2. Example circuit

to create checker circuitry. A PDTGM for NR PDFs (resp. FS PDFs) is hereinafter referred to as an NR-TGM (resp. FS-TGM).

For a PDF f in a combinational circuit C, the size of the NR-TGM (resp. FS-TGM) obtained by the above procedure is roughly approximated as $2g_o + g_p + 1$ (resp. $2g_o + 2g_p + 1$) [gates], where g_o is the number of gates in C, and g_p is the number of gates on the path corresponding to f. Here, it is assumed that all the circuits of our test generation model are composed of 2-input gates.

B. Test Generation Procedure

Given a combinational circuit C and a set of PDFs F in C, test generation is performed as follows.

(1) Transform C into a PDTGM C^F_{TGM}.

(2) For each $f' \in F'$, where F' is the set of SAFs corresponding to F, perform the following steps.

  (a) Generate a test pattern for f' by performing stuck-at test generation on C^F_{TGM}.

  (b) Perform fault simulation by applying the test pattern to C^F_{TGM}, then the SAFs detected by it are dropped from F'.

  (c) Repeat step (2) until F' is empty.

(3) Transform the obtained test patterns for F' into the two-pattern tests for F.

Note that, in (1) of the above procedure, C^F_{TGM} is an NR-TGM (resp. FS-TGM) if NR PDFs (resp. FS PDFs) are targeted.

To clarify our test generation method, we give an example here. Let us consider performing test generation for FS PDFs in the example circuit shown in Figure 2. Figure 3 is a FS-TGM for the rising PDF on path (b, c, f, i) in the example circuit. By performing stuck-at test generation on the FS-TGM, for example, a two pattern test (a, b) = (01, 01) is obtained.

The correctness of our method is guaranteed by the following theorem.

Theorem 1: Let C and f be a combinational circuit and a PDF in the circuit, respectively. Let $C^f_{\text{NR-TGM}}$ (resp. $C^f_{\text{FS-TGM}}$) and f' be an NR-TGM (resp. FS-TGM) for f and the SAF

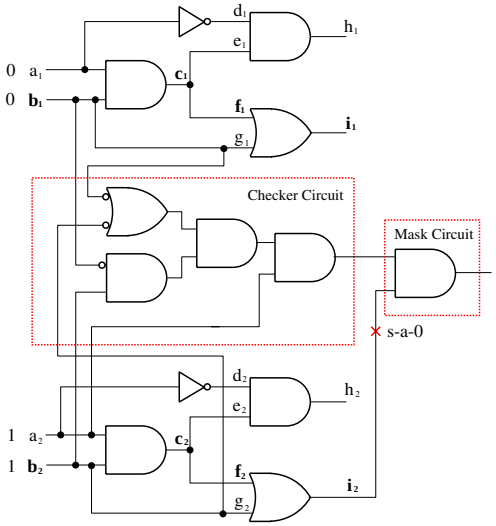


Fig. 3. FS-TGM for the rising PDF on path (b, c, f, i)

corresponding to f, respectively. Then, f is NR (resp. FS) if and only if f' is testable.

Proof: First, we show that if f' is testable, f is NR (resp. FS). Since f' is testable, there exists a test pattern t'. Let t be the two-pattern test for C mapped from t'. Clearly, if t is applied to C, it is guaranteed that the appropriate transition is launched at the starting point of the path with f and all the off-inputs satisfy the conditions for the detectability of an NR PDF (resp. FS PDF) by the checker circuit. This implies that f is NR (resp. FS).

Next, we demonstrate that if f is NR (resp. FS), f' is testable. Since f is NR (resp. FS), there exists a two-pattern test $\tau = (v_1, v_2)$. Let τ' be a test pattern for $C^f_{\text{NR-TGM}}$ (resp. $C^f_{\text{FS-TGM}}$) in which the pattern for the first copy C_1 of C is mapped from v_1, and that for the second copy C_2 of C is mapped from v_2. By applying τ' to $C^f_{\text{NR-TGM}}$ (resp. $C^f_{\text{FS-TGM}}$), the checker circuit produces the value that makes the mask circuit inactive. This is because the structures of C_1 and C_2 are identical to the structure of C, and τ sets all the off-inputs of C to the values that satisfy the conditions for the detectability of an NR-PDF (resp. FS-PDF). Moreover, f' is activated by v_2. Since the mask circuit is inactive, the fault effect of f' is propagated to a primary output. Hence, f' is testable and the proof is complete. ■

Theorem 1 implies that f is not FS, i.e., functional unsensitizable, if f' is untestable in an FS-TGM. Any functional unsensitizable PDF never impacts the timing behavior of a circuit [9]. Thus, our method can also identify false paths in a combinational circuit.

IV. Experimental Results

To show the feasibility of our method and its effectiveness, we conducted the following experiments on a Linux

TABLE II

CIRCUIT SIZES AND TIME FOR CONSTRUCTION

Circuit	Class	ORG-size [gates]	TGM-size [gates]	TRN-time [s]
c5315	NR	2,309	23,048	3.09
	FS		129,177	4,44
c6288	NR	2,416	92,309	19.87
	FS		646,732	23.04
c7552	NR	3,512	24,099	4.45
	FS		136,122	5.08
s5378	NR	3,137	23,409	3.19
	FS		83,069	3.51
s6669	NR	3,558	24,000	7.98
	FS		250,836	8.37
s9234.1	NR	6,019	36,051	4.70
	FS		124,361	5.18
s13207.1	NR	9,227	29,870	3.21
	FS		75,208	3.51
s15850.1	NR	10,840	38,702	6.62
	FS		150,921	7.11
s35932	NR	19,526	54,088	5.73
	FS		161,505	5.89
s38417	NR	25,451	44,806	6.57
	FS		172,168	7.06
s38584.1	NR	22,107	33,799	5.20
	FS		116,001	5.58

TABLE III

TEST GENERATION RESULTS FOR NR AND FS PDFS

Circuit	Class	#flt	#det	#unt	#abt	#vec	TGT [s]
c5315	NR	10,000	6,080	3,920	0	1,070	38.49
	FS		7,442	2,558	0	1,065	37.06
c6288	NR	10,000	0	10,000	0	0	45.92
	FS		0	10,000	0	0	137.21
c7552	NR	10,000	4,292	5,684	24	1,282	290.61
	FS		6,949	3,043	8	1,868	256.80
s5378	NR	10,000	7,420	2,580	0	1,257	1.11
	FS		8,702	1,298	0	932	1.58
s6669	NR	10,000	5,070	4,930	0	1,018	2.59
	FS		5,090	4,910	0	819	12.54
s9234.1	NR	10,000	3,464	6,536	0	713	1.09
	FS		4,778	5,222	0	668	2.66
s13207.1	NR	10,000	8,171	1,829	0	1,200	0.84
	FS		8,920	1,080	0	801	1.02
s15850.1	NR	10,000	4,068	5,932	0	667	2.45
	FS		5,729	4,271	0	690	8.39
s35932	NR	10,000	0	10,000	0	0	0.09
	FS		131	9,869	0	20	0.61
s38417	NR	10,000	3,854	6,146	0	1,407	2.78
	FS		6,082	3,918	0	1,460	5.43
s38584.1	NR	10,000	5,251	4,749	0	1,262	1.72
	FS		7,088	2,912	0	1,529	4.77

workstation (CPU: Opteron 150 2.4GHz, Memory: 8GB). In the experiments, we used large ISCAS 85' benchmark circuits (c5315, c6288 and c7552) and the combinational parts of large ISCAS '89 benchmark circuits (s5378, s6669, s9234.1, s13207.1, s15850.1, s35932, s38417 and s38584.1). For each circuit, we extracted $10,000$ long paths by using a commercial static timing analyzer, and we used a commercial ATPG tool which can handle both SAFs and PDFs.

First, to show the feasibility of the proposed method, we classified the PDFs on the $10,000$ long paths into NR or FS PDFs by using our method. In each circuit, we constructed the PDTGMs for NR and FS PDFs by using an in-house Perl program. Table II shows the sizes of each circuit ("ORG-size") and its PDTGM ("TGM-size") for NR or FS PDFs, which are estimated by the ATPG tool, and time required for constructing a PDTGM ("TRN-time"). Note that the ATPG tool removed some unnecessary gates during test generation. Rows "NR" and "FS" represent the results of test generation for NR and FS PDFs, respectively. In all the circuits, the sizes of the PDTGMs became large, especially in "FS," compared with those of the original circuits. However, this is not critical for the efficiency of test generation as demonstrated later. Even if increasing the size of a PDTGM is critical for some aspects such as the memory size used during test generation, it can be alleviated as follows. In the checker circuit of a PDTGM, only its functionality is important. Therefore we can perform logic minimization for the checker circuit, and it will produce a much simpler circuit. The checker circuit, moreover, can be reduced by targeting a reasonable number of faults at a time and by performing test generation several times. In Table III, columns "#flt," "#det," "#unt" and "#abt" give the number of targeted faults, detected faults, untestable faults identified during test generation for NR or FS PDFs and aborted faults, respectively. Columns "#vec" and "TGT" denote the number of two-pattern tests and test generation time including fault simulation time, respectively. Theoretically, all the NR PDFs are also FS. In Table III, the number of FS PDFs was always larger than that of NR PDFs. Some non-robustly untestable PDFs are FS in theory. This can also be observed in Table III. For example, in the case of c5315, 1,362 of 3,920 non-robustly untestable PDFs were FS. The remaining 2,558 PDFs were functionally unsensitizable which do not affect the circuit performance.

TABLE IV

NORMAL APPROACH VS. OUR APPROACH

Circuit	Method	#flt	#det	#unt	#abt	#vec	TGT [s]
c5315	Normal	10,000	6,080	3,920	0	1,167	215.71
	Proposed		6,080	3,920	0	1,070	38.49
c6288	Normal	10,000	0	10,000	0	0	16.04
	Proposed		0	10,000	0	0	45.92
c7552	Normal	10,000	4,301	5,653	46	1,373	1,464.81
	Proposed		4,292	5,684	24	1,282	290.61
s5378	Normal	10,000	7,420	2,580	0	1,369	15.58
	Proposed		7,420	2,580	0	1,257	1.11
s6669	Normal	10,000	5,070	4,930	0	1,395	27.35
	Proposed		5,070	4,930	0	1,018	2.59
s9234.1	Normal	10,000	3,464	6,536	0	749	15.09
	Proposed		3,464	6,536	0	713	1.09
s13207.1	Normal	10,000	8,171	1,829	0	1,196	13.00
	Proposed		8,171	1,829	0	1,200	0.84
s15850.1	Normal	10,000	4,068	5,932	0	704	20.39
	Proposed		4,068	5,932	0	667	2.45
s35932	Normal	10,000	0	10,000	0	0	4.19
	Proposed		0	10,000	0	0	0.09
s38417	Normal	10,000	3,854	6,146	0	1,500	44.54
	Proposed		3,854	6,146	0	1,407	2.78
s38584.1	Normal	10,000	5,251	4,749	0	1,265	31.22
	Proposed		5,251	4,749	0	1,262	1.72

TABLE V
MULTI-PASS TEST GENERATION

Fault group	TGT [s]	TRN-size [gates]
1–1,000	0.05	9,326
1,001–2,000	0.07	11,167
2,001–3,000	0.07	10,538
3,001–4,000	0.09	12,077
4,001–5,000	0.07	11,230
5,001–6,000	0.07	11,608
6,001–7,000	0.07	10,942
7,001–8,000	0.07	11,537
8,001–9,000	0.06	11,551
9,001–10,000	0.05	10,711
Total	0.67	—

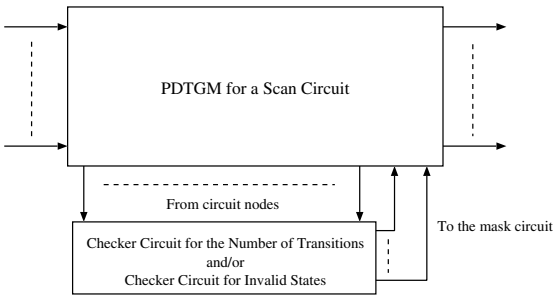


Fig. 5. Test generation model considering over-testing and/or test power for a scan circuit

Here, we show the experimental results to verify the effectiveness of our method. The best way to verify the effectiveness of our method is that the results obtained by using the following two approaches are compared. One approach uses the most efficient ATPG tool for PDFs, and the other approach uses the proposed method with the most efficient ATPG tool for SAFs. However, it is difficult to perform such a experiment. Instead, we performed the following experiment. Since the commercial ATPG tool that we used can handle not only SAFs but also PDFs in a combinational circuit, we compared a normal method ("Normal") where the tool targets PDFs to our method ("Proposed") where the tool targets SAFs. Note that since the ATPG tool cannot handle FS PDFs, NR PDFs were only targeted in this experiment. Table IV shows the test generation results of "Normal" and "Proposed." Except the case of c6288, we achieved short test generation time compared with the normal method. Furthermore, in the case of c7552, our method reduced the number of aborted faults. In terms of the number of two-pattern tests, our method also obtained good results. To examine the case of c6288 further, an additional experiment was performed. In the additional experiment, 10,000 PDFs were partitioned into 10 groups. Each group has 1,000 PDFs. Our method was applied to each group, i.e., test generation was performed 10 times. The total time required for constructing 10 PDTGMs was 20.65 seconds. As shown in Table V, the total test generation time was drastically improved from 45.92 seconds to 0.67 seconds. In this way, the efficiency of our method can be enhanced. In future work, we should consider a good way to partition given PDFs into groups in order to enhance the efficiency of our method.

From the above experimental results, we can see that our test generation framework using checker circuitry is effective and can easily be implemented by making use of an existing ATPG tool for SAFs.

V. SOME APPLICATIONS

This section briefly discusses some applications of the proposed test generation framework using checker circuitry. In the future, the following topics should be discussed in detail.

A. Scan Circuits

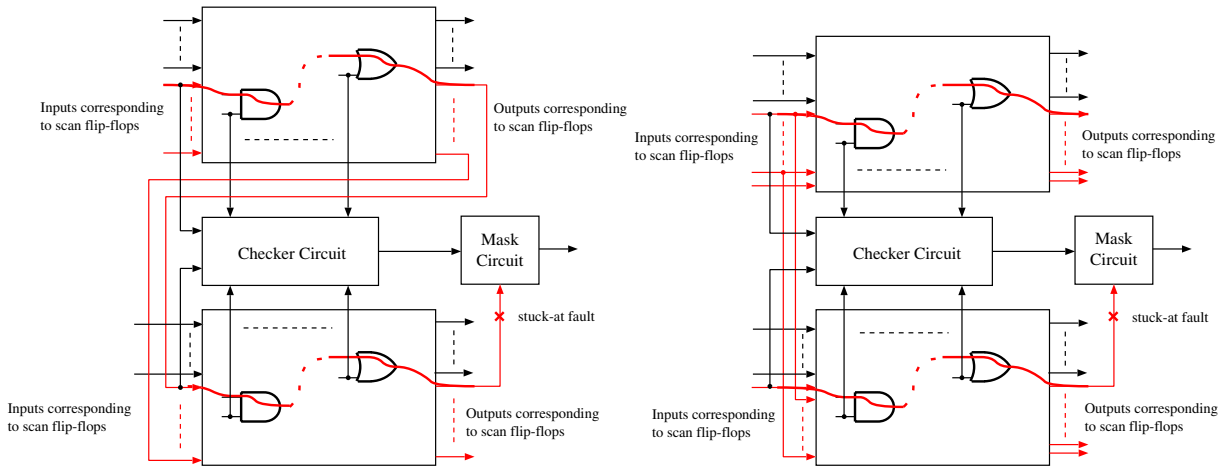
In section III, a combinational circuit is targeted. Let us consider a scan circuit here. Scan methodology for PDFs can be classified into two categories: the enhanced scan method [10] and the standard scan one. The standard scan method can be further divided into the broadside method [11] and the skewed-load method [12]. For a circuit designed by the enhanced scan method, our method described in section III can be applied without modification. In the case of the standard scan method, PDTGMs can be represented as Figure 4. Figure 4(a) explains how to generate broadside tests for PDFs. The combinational part of a scan circuit is duplicated and connected as Figure 4(a), that is, the outputs corresponding to the scan flip-flops (FFs) in the first copy are cascaded to the inputs corresponding to the scan FFs in the second copy. In the case of skewed-load testing shown in Figure 4(b), the respective inputs corresponding to the scan FFs in the first copy are connected to the inputs corresponding to those adjacent scan FFs on the scan chain. By using the above test generation model, we can test PDFs in a scan circuit.

B. Over-testing and Test Power

Over-testing [14] and excessive test power [15] in scan methodology are crucial issues, and it is important to enhance the test quality and to reduce the yield loss caused by those issues. Let us consider here how to deal with such problems under the proposed framework.

For a scan circuit, detecting PDFs by unsettable values in the scan FFs during normal operation is one of the main reasons that over-testing is induced. To avoid this phenomenon, the information of invalid states in a given circuit should be taken into account during test generation. In our test generation method, by checking the information of invalid states, which are extracted by some method such as [13], as well as the conditions of the off-inputs, over-testing can be alleviated.

Excessive test power may not only damage the circuit under test, but may also cause instantaneous voltage drops which result in making the test invalid. In testing of static faults such as SAFs, this undesirable situation can be avoided by slowing down the clock frequency during testing. We cannot, however,



(a) PDTGM for broadside testing

(b) PDTGM for skewed-load testing

Fig. 4. PDTGMs for a scan circuit

slow down the frequency in delay testing. In this case, power-aware two-pattern tests are required, i.e., we should test delay faults under power constraints in a scan environment. Such constraints may be estimated by applying functional patterns to the circuit. As shown in Figure 4, the first and second vectors of two-pattern tests are considered separately. This makes it possible to identify which circuit nodes have transitions. As a result, we can compare the number of transitions in a given circuit to the information of power constraints during test generation. This can be realized by some kind of checker circuitry in our framework.

It can be seen from the above discussion that our test generation framework can consider various properties in a unified test generation model of Figure 5.

VI. Conclusions

In this paper, we presented a test generation framework for PDFs using checker circuitry. This framework can utilize existing mature ATPG techniques for SAFs. Since our method can handle FS PDFs, false paths in a combinational circuit can be identified. We showed that our method is effective in terms of test generation time, fault efficiency and the number of two-pattern tests through the ISCAS '85 and '89 benchmark circuits. The feasibility of applying our framework to the over-testing and test power problems was also discussed.

Acknowledgments

This work was supported in part by the Research Promoting Expenses for Associates of JAIST, the Foundation for the Promotion of Industrial Science, and the Japan Society for the Promotion of Science under Grants-in-Aid for Young Scientists (B) (No. 17700062) and for Scientific Research (C) (No. 19560340).

References

[1] G. L. Smith, "Model for delay faults based upon paths," *Int. Test Conf.*, pp. 342–349, 1985.

[2] C. J. Lin and S. M. Reddy, "On delay fault testing in logic circuits," *IEEE Trans. on CAD*, Vol. CAD-6, No. 5, pp. 694–703, Sep. 1987.

[3] S. Patil and S. M. Reddy, "A test generation system for path delay faults," *Proc. Int. Conf. on Comp. Design*, pp. 40–43, 1989.

[4] K. Fuchs, F. Fink and M. H. Schulz, "DYNAMITE: an efficient automatic test pattern generation system for path delay faults," *IEEE Trans. on CAD*, Vol. 10, No. 9, pp. 1323–1335, Oct. 1991.

[5] A. Saldanha, R. K. Brayton and A. L. Sangiovanni-Vincentelli, "Equivalence of robust delay-fault and single stuck-fault test generation," *Proc. Design Automation Conference*, pp. 173–176, 1992.

[6] M. A. Gharaybeh, M. L. Bushnell and V. D. Agrawal, "Classification and Test Generation for path-delay faults using single stuck-at fault tests," *Journal of Electronic Testing: Theory and Application*, Vol. 11, No. 1, pp. 55–67, Aug. 1997.

[7] S. Ohtake, K. Ohtani and H. Fujiwara, "A method of test generation for path delay faults using stuck-at fault test generation algorithms," *Proc. Design, Automation and Test in Europe*, pp. 310–315, 2003.

[8] D. Xiang, K. Li, H. Fujiwara and J. Sun, "Generating compact robust and non-robust tests for complete coverage of path delay faults based on stuck-at tests," *Proc. 24th IEEE International Conference on Computer Design*, pp. 446–451, 2006.

[9] A. Krstić and K.-T. Cheng, *Delay fault testing for VLSI circuits*, Kluwer Academic Publishers, 1998.

[10] B. I. Dervisoglu and G. E. Stong, "Design for testability: using scanpath techniques for path-delay test and measurement," *Proc. International Test Conference*, pp. 365–374, 1991.

[11] J. Savir and S. Patil, "Broad-side delay test," *IEEE Trans. on CAD*, Vol. 13, No. 8, pp. 1057–1064, Aug. 1994.

[12] J. Savir and S. Patil, "Scan-based transition test," *IEEE Trans. on CAD*, Vol. 12, No. 8, pp. 1232–1241, Aug. 1993.

[13] H. C. Liang, C. L. Lee and J. E. Chen, "Invalid state identification for sequential circuit test generation," *Proc. Asian Test Symposium*, pp. 10–15, 1996.

[14] *International technology roadmap for semiconductors*, http://public.itrs.net/, 2005.

[15] P. Girard, "Survey of low-power testing of VLSI circuits," *IEEE Design and Test of Computers*, Vol. 19, No. 3, pp. 82–92, May 2002.