논문 2010-47SD-2-7

드모르간 및 재대입 변환의 경로지연고장 테스트집합 유지

(Path Delay Test-Set Preservation of De Morgan and Re-Substitution Transformations)

이 준 환*, 이 현 석**

(Joonhwan Yi and Hyunseok Lee)

요 약

드모르간 및 재대입 논리변환은 unate gate network (UGN)을 보다 일반적인 balanced inversion parity (BIP) network으로 전환하는데 충분하다. 이러한 회로계층에 대해서도 자세히 논의하고 있다. 우리는 드모르간 및 재대입 논리변환이 경로지연고장 테스트집합을 유지한다는 것을 증명하였다. 본 논문의 결과를 이용하여 함수 z를 구현하는 모든 UGN에서 모든 경로지연고장을 검출하는 상위수준 테스트집합은 함수 z의 어떠한 BIP realization에서도 모든 경로지연고장을 검출한다는 것을 보일 수 있다.

Abstract

Two logic transformations, De Morgan and re-substitution, are sufficient to convert a unate gate network (UGN) to a more general balanced inversion parity (BIP) network. Circuit classes of interest are discussed in detail. We prove that De Morgan and re-substitution transformations are test-set preserving for path delay faults. Using the results of this paper, we can easily show that a high-level test set for a function z that detects all path delay faults in any UGN realizing z also detects all path delay faults in any BIP realization of z.

Keywords : path delay fault, test-set preserving, testability, logic transformation, De Morgan, re-substitution

Ⅰ. Introduction

Test-set preserving logic transformations[1] have been studied in several areas including logic synthesis[2~3] where a circuit is transformed for various reasons such as low area, high speed, or low power[9]. The automatic test pattern generation (ATPG) cost can be kept low if a test set for a circuit C can be reused even though C is re-synthesized several times. Note that ATPG of a modern system-on-a-chip whose size is more than ten million gates easily takes more than few days.

In addition, test-set preserving logic transformations are useful in high-level testing, which is the main motivation of this paper. High-level fault models pursue realization-independent coverage of gate-level faults for a well-defined logic circuit class[6, 10~12]. Usually their coverage for a relatively simple circuit class is proven first. Then, the result is extended to more complex circuit classes by using the fact that any circuit in the simple class can be converted to a circuit in the complex class via a sequence of test-set preserving logic transformations.

* 정회원, 광운대학교 컴퓨터공학과
(Computer Engineering, Kwangwoon University)
** 정회원, 광운대학교 전자통신공학과
(Electronics and Communications, Kwangwoon University)
※ 이 논문은 서울시 산학연 협력사업(10560) 지원에 의해 연구되었습니다.
※ "본 연구는 교육과학기술부/한국연구재단 기초연구사업(과제번호 2009-0088455) 지원으로 수행되었음."
접수일자: 2009년8월9일, 수정완료일: 2010년1월25일

In [6], Kim and Hayes show that their realization independent block (RIB) tests detect all single and multiple stuck-at line (SSL and MSL) faults in any unate gate network (UGN). Then, they extend their results to a more general circuit class named balanced inversion parity (BIP) networks by proving that any UGN can be transformed to a BIP network by a sequence of two logic transformations, De Morgan and re-substitution, which are test-set preserving for SSL and MSL faults[1]. Circuit classes of interest are discussed in detail later. Sparmann et *al.*[10] proposed so-called the universal delay test set for UGNs. The universal delay test set of a function z detects all path delay faults in any UGN realization of z. If De Morgan and re-substitution transformations are test-set preserving for path delay faults, the results of [10] can be easily extended to more general BIP networks.

Various test-set preserving logic transformations for SSL and MSL faults[1] or path delay faults[2~3, 5] have been studied previously. However, to our knowledge, the path delay testability of De Morgan and re-substitution transformations has not been addressed in other publications. In this paper, we show that both De Morgan and re-substitution transformations are test-set preserving for path delay faults.

Preliminary terminologies are explained in Section II and circuit classes of interest are discussed in Section III. Both De Morgan and re-substitution transformations are proven to be test-set preserving for path delay faults in Section IV. Then, we conclude this paper in Section V.

II. Preliminaries

A *path* is an alternating sequence of nets and gates from a primary input or a flipflop output to a primary output or a flipflop input. An input i of a gate G is an *on-path* input if i is on the path under consideration. Other inputs of G are called *off-path* inputs.

Each path p in a circuit is associated with two path delay faults, *slow-to-rise* $\uparrow p$ and *slow-to-fall* $\downarrow p$. To detect a slow-to-rise (slow-to-fall) fault, we need to propagate a signal transition on the path input to the path output, and see if a rising (falling) signal transition is propagated to the output in time. This implies that to detect a delay fault, we require a *delay test* $t = \langle v_1, v_2 \rangle$ composed of two input patterns v_1 and v_2. Figure 1 illustrates delay test generation for faults in the gate-level realization C of the 2-input XOR function. Figure 1a shows the test $t_R = \langle 00, 10 \rangle$, which first applies 00 to ab and then applies 10 to ab. In other words, b is held at 0 while a rising signal is applied to a. This delay test propagates a signal transition through the path under test adz without spurious pulses or *glitches* independent of the delays in C. Such a delay test t_R is called a *robust test*[4, 7] because t_R detects the target path delay fault independent of other path delays in C. It is obvious that if the measured output at z after the specified time is 1, path adz is free from slow-to-rise faults. On the other hand, test $t_{NR} = \langle a'b, ab \rangle$ propagates a signal transition through path $acdz$, and some glitches may occur at the path output depending on the delay configuration in C, as shown in Figure 1b. In this case, even if the measured output after the specified time is 0, it is not guaranteed that path $acdz$ is free from slow-to-fall faults. Note that path $acdz$ is faulty if the measurement time is t_1, but fault-free if the measurement time is t_2. That is, the test result of t_{NR} is not definite and such test is called *non-robust*[4, 7]. Robust tests guarantee detection of the target fault, while non-robust tests do not.

III. Circuit Classes

A *restricted gate network* (RGN)[8] is constructed from AND and OR gates only, with inverters used, if necessary, at the branches of primary inputs. Figure 1 illustrates the general form of an RGN realization N_z of a function z. The AND/OR subnetwork N_z* of

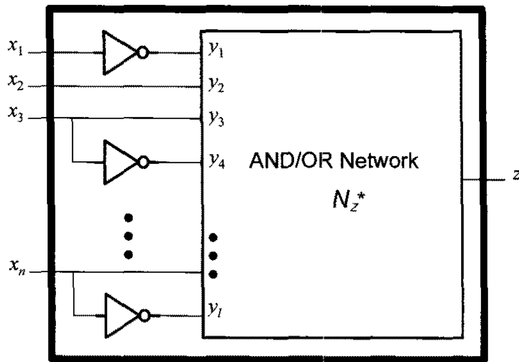


그림 1. Restricted gate network
Fig. 1. Restricted gate network.

N_z contains only AND or OR gates and all inverters appear only before N_z*. Since inverters are used only if necessary, the unateness of each input x_i with respect to an output z_j can be determined by the presence or absence of inverters. For example, output z in Figure 1 is negative unate in x_1, positive unate in x_2, binate in x_3, and binate in x_n. We discuss the relationship between RGNs and more general networks in the following.

The implementation restrictions are more relaxed in *unate gate networks* (UGNs)[8] which realize the AND/OR subnetwork N* of an RGN N such that the inversion parity for every path between two points in N* is the same. UGNs are more general than RGNs because they do not restrict inversions to primary input branches. In UGNs, inverters at primary input branches may move into N* and inverters in N* may move to primary input branches as long as all paths between any two points in N* retain the same inversion parity. That is, UGNs have the same inversion parity for all paths between two points after the inverters at primary input branches.

An even more general circuit class is the balanced inversion parity (BIP) class[6]. In a BIP network, whenever possible all paths between any input variable to an output have the same inversion parity. That is, in a BIP realization of a function z, all paths from a unate variable of z have the same inversion parity, while at least one path from a binate variable must have a different inversion parity from other paths. This implies that all paths between any two

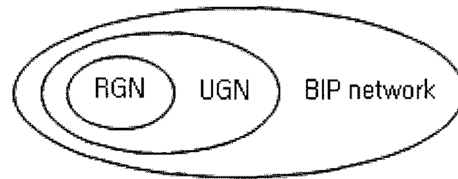


그림 2. RGNs, UGNs, BIP networks의 관계
Fig. 2. Relationship among RGNs, UGNs, BIP networks.

points driven by a unate input must have the same inversion parity but paths between two points driven by a binate input may have different inversion parities. BIP networks are more general than UGNs because they allow different inversion parities for paths between two points.

RGNs, UGNs, and BIP networks are defined not only by the circuit structures but also by the circuit functions. Because inverters are only added if necessary, an RGN also meets the BIP constraint, which allows only odd (even) inversion parity for all paths between a positive (negative) unate input to an output. A UGN changes the AND/OR subnetwork N* of an RGN such that paths between any two points including inputs and outputs of N* have the same inversion parity. Since RGNs have inverters at primary input branches only if necessary, paths between any primary input and any primary output in UGNs have the same inversion parity whenever possible. Therefore, a UGN is also an BIP network. Of course, an RGN is also a UGN because every path in the AND/OR subnetwork of an RGN has the same even inversion parity. The Venn diagram in Figure 2 shows this inclusion relationship among RGNs, UGNs, and BIP networks.

Figure 3 shows examples of restricted gate, unate gate, and BIP realizations of the function $z = bd + abc + a'cd$. Figure 3a appears to be the only RGN realization of z. Figures 3a and 3b show UGN realizations of z because all paths between any two points after the inverters at primary input branches have the same inversion parity. For example, paths between c and z have the same even inversion parity in Figures 3a and 3b. Observe that Figure 3c is not a

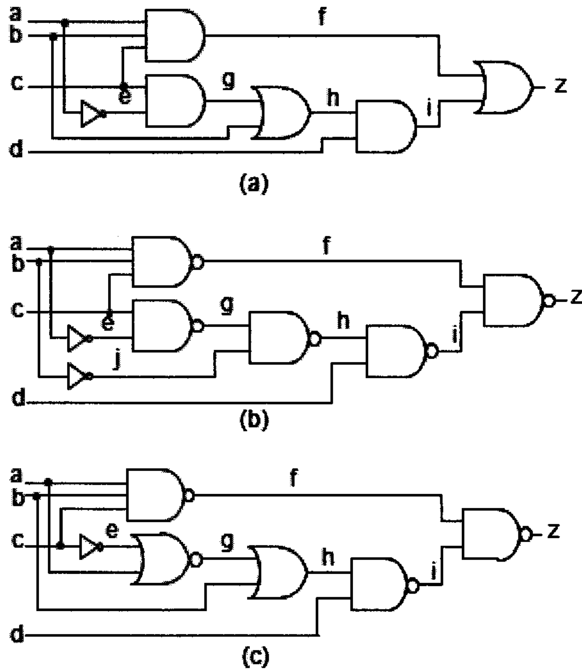


그림 3. 4-입력 함수 z = bd + abc + a′cd의 세가지 구현회로: (a) restricted gate network (RGN) (b) unate gate network (UGN) (c) balanced inversion parity (BIP) network

Fig. 3. Three realizations of the 4-input function z = bd + abc + a′cd: (a) RGN, (b) UGN, and (c) BIP network.

UGN because the two possible paths between a and z have different inversion parities. Finally, all three realizations in Figure 3 are BIP realizations of z. BIP realizations cover a broad range of implementations including minimal ones, because any inversion that violates the BIP condition can be easily removed without adding gates[6].

Ⅳ. Test-Set Preservation of De Morgan and Re-Substitution Transformations

A *test-set preserving* logic transformation[1] L for a fault type F transforms a circuit N_1 to N_2 denoted $L(N_1) = N_2$ such that a test set for N_1 detects all faults of type F in N_2.

Definition 1[1] $M(N_1) = N_2$ is a *De Morgan transformation* if $N_1 = x_1x_2...x_n$ and $N_2 = (x_1' + x_2' + ... + x_n')'$, or vice versa.

Figure 4 illustrates the De Morgan transformation

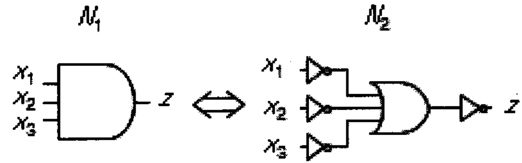


그림 4. 드모르간 변환

Fig. 4. De Morgan transformation.

표 1. 게이트수준 회로에서 경로지연고장 테스트의 구분

Table 1. Path delay test classification in gate-level circuits (cv and ncv stand for controlling and non-controlling values, respectively. X denotes don't care values including glitches).

Off-path input	On-path input	
	$cv \to X \to ncv$	$ncv \to X \to cv$
$X \to ncv$	robust	non-robust
stable ncv	robust	robust

applied to three-input circuits.

Path delay test generation assumes the worst-case delay configuration of the circuit C_0 under test, which makes a test non-robust whenever possible. In other words, a test t for a fault f is a robust test if t detects f with any delay configuration of C_0; t is a non-robust test if it detects f in some, but not all, cases. So, under this assumption, the classification of robust and non-robust tests is determined only by the on- and off-path input values as shown in Table 1.

Consider the example in Figure 5. Test t_r = <$a'b'$, ab'> for fault ↓$acfh$ is robust because it detects ↓$acfh$ with any delay configuration including the worst case. Note that every off-path inputs are stable at

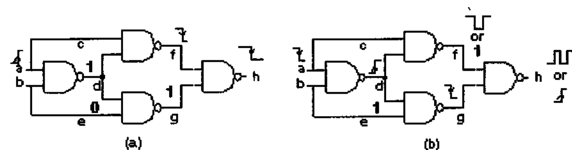


그림 5. 2-input XOR 함수의 구현 예: (a) ↓$acfh$의 강건 테스트 예 <$a'b'$, ab'> (b) ↑$adgh$의 비강건 테스트 예 <ab, $a'b$>

Fig. 5. A realization of the 2-input XOR function: (a) a robust test <$a'b'$, ab'> for ↓$acfh$ and (b) a non-robust test <ab, $a'b$> for ↑$adgh$.

noncontrolling values while t_r is applied. On the other hand, test t_n = <ab, a'b> for fault ↑*adgh* is non-robust because it may not detect ↑*adgh*. Assume that the rising transition arrives at d earlier than the falling transition at c so that line f has a glitch. If the glitch on f arrives earlier than the falling transition on g, t_n causes glitches on output h and thus t_n may not detect ↑*adgh*. That is, t_n may not detect the target fault with some (worst-case) delay configurations.

Consider a logic transformation $L(N_1)$ = N_2. Assume that every path p_1 in N_1 has a corresponding path p_2 in N_2, and each on- and off-path input of p_1 has a corresponding one for p_2 in N_2. Then, one possible way to prove that L is test-set preserving for path delay faults is to show that every two-pattern pair causes the same controlling and non-controlling value changes on corresponding on- and off-path inputs of p_1 and p_2. De Morgan and re-substitution transform circuits in such a way. We will use this fact to prove that De Morgan and re-substitution are test-set preserving for path delay faults.

Theorem 1 The De Morgan transformation $M(N_1)$ = N_2 is test-set preserving for path delay faults.

Proof. N_1 and N_2 have the same input and output signals, and the same, unique path from each input i to the output z. Therefore, there is a one-to-one correspondence between paths in N_1 and N_2. Path p_1 from i to z in N_1 corresponds to path p_2 from i to z in N_1. Let G_1 and G_2 denote the gates in N_1 and N_2, respectively, except inverters. That is, if G_1 is an AND (OR) gate, then G_2 is an OR (AND) gate. The input x_1 of G_1 on p_1 also corresponds to the input x_2 of G_2 on p_2. Also, there is a one-to-one correspondence between on- and off-path inputs of p_1 and p_2. Assume that G_1 is an AND gate. To apply the controlling value 0 to G_1, an input j of N_1 needs to be 0. Likewise, to apply the controlling value 1 to G_2, input j in N_2 needs to be 0 because there is an inverter between j and G_2. Similarly, to apply the non-controlling value of G_1 and G_2, j needs to be 1 in both N_1 and N_2, which also holds when G_1 is an OR gate. Therefore, a two-pattern test t = <v_1, v_2> causes the same controlling/non-controlling value transitions at inputs of G_1 and G_2 without glitches because there is a unique path from each primary input to G_1 and G_2. This implies that t is a robust (non-robust) test for f_2 if t is a robust (non-robust) test for f_1. □

When $M(N_1)$ = N_2, with the worst-case delay configuration of N_1 and N_2, an input change t never causes a spurious signal change at the output of N_1 if t does not cause a spurious signal change at the output of N_2. This is because there is only one multiple-input gate in both N_1 and N_2, and t causes the same controlling/non-controlling value changes at the inputs of the gates.

Definition 2[1] $R(N_1)$ = N_2 is a re-substitution transformation if N_1 = $[E]_1,[E]_2,...,[E]_k$ is composed of k copies of a logic circuit E that does not fan out and N_2 is a single copy of E whose output fans out to k places, as illustrated in Figure 6.

Theorem 2 The re-substitution transformation $R(N_1)$ = N_2 is test-set preserving for path delay faults.

Proof. Consider an output z_j in N_1 and N_2. As shown in Figure 6, z_j is driven by $[E]_j$ in N_1 and by $[E]$ in N_2. Since $[E]_j$ and $[E]$ are identical, each path p_1 ending at z_j in N_1 has a corresponding path p_2 ending at z_j in N_2. Also, on- and off-path inputs of p_1 correspond exactly to on- and off-path inputs of p_2 respectively. Therefore, with the worst-case delay assumption, all on- and off-path inputs of corresponding paths in N_1 and N_2 have the same

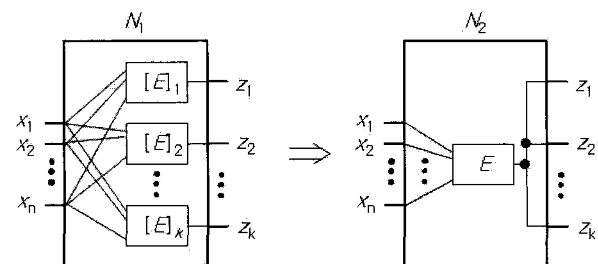


그림 6. 재대입 변환
Fig. 6. Re-substitution transformation.

controlling/non-controlling signal changes for the same input change. This implies that a test for a robust (or non-robust) fault f_1 in N_1 is also a test for the corresponding fault f_2 of f_1 in N_2. □

In circuit N_2 of Figure 6, all outputs always experience the same signal changes, including glitches, for any input change on the x_i's although their propagation delays may vary. On the other hand, in circuit N_1 of Figure 6, all outputs will have the same initial and final values for an input change but they generally will exhibit different signal changes and delays because glitches may be present. We next show that all outputs of N_1 have the same signal changes with or without glitches if N_1 is embedded in a circuit and the worst-case delay configuration is assumed. This fact will be used to prove that re-substitution still preserves path delay test set when it is applied to subcircuits of a circuit.

According to Table 1, on- and off-path inputs need to have opposing controlling/noncontrolling signal transitions in order to make a test non-robust. This condition is always satisfied when an off-path input has glitches that always have both rising and falling transitions. That is, if an off-path input has glitches when a test t is applied, t becomes a non-robust test.

Assume that N_1 is embedded in a circuit C. Let A denote the circuit surrounding N_1 in C. Consider a fault f_p associated with a path p in C where p

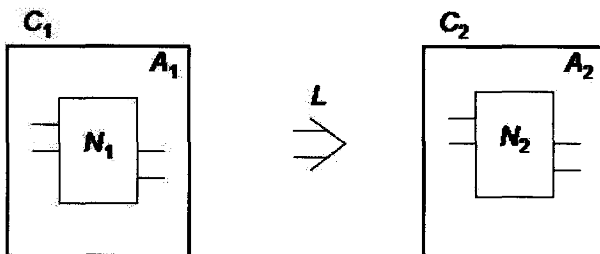


그림 7. 주어진 회로 C_1과 변환된 회로 C_2. 경로지연고자 테스트집합 유지 논리변환 L에 의해 C_1의 하위회로 N_1이 N_2로 변환되었다.

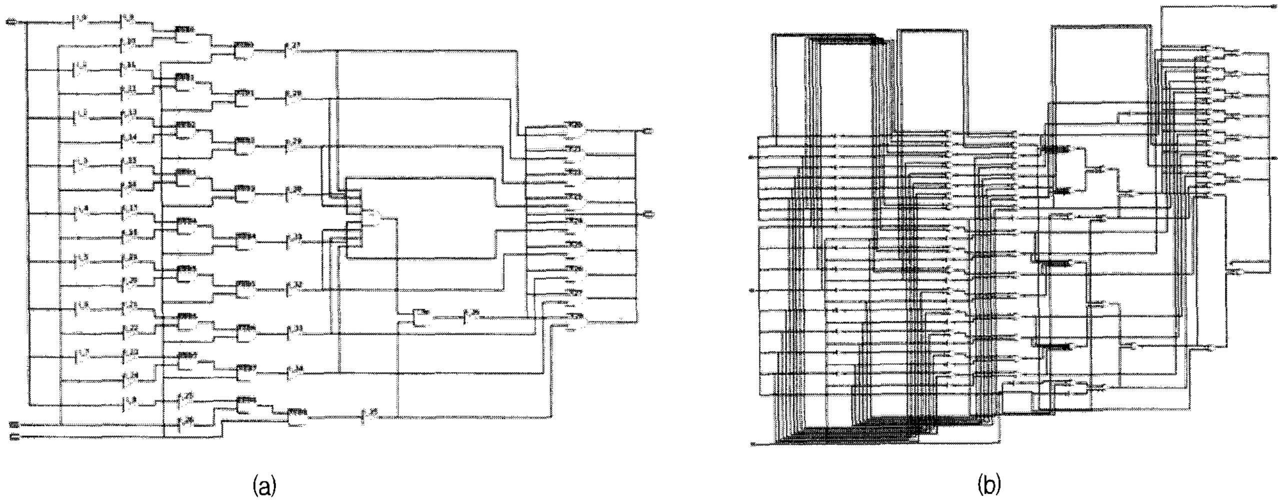
Fig. 7. The original circuit C_1 and its transformed circuit C_2. The subcircuit N_1 of C_1 is transformed to N_2 by a test-set preserving logic transformation L for path delay faults.

contains a sub-path p_j ending at output z_j in N_1. In N_1, subcircuits $[E]_1$, $[E]_2$,..., $[E]_k$ are identical. So, if z_j has no glitch when a test t for f_p is applied to C with any (worst-case) delay configuration of C, all other outputs $Z_j = \{z_1, z_2, ..., z_{j-1}, z_{j+1},..., z_k\}$ of N_1 cannot have glitches. On the other hand, when t causes a glitch on z_j with the worst-case delay configurations, the test engineer may assign glitches to some other outputs in Z_j to maximize the probability that t becomes non-robust. If z_i has glitches and they are propagated to some off-path inputs of p in A, t becomes a non-robust test for f_p. Obviously, when all outputs of Z_j have glitches, the probability of t becoming non-robust is highest. Consequently, for delay test generation that assumes the worst-case delay configuration of C, all outputs z_1,..., z_k of N_1 must have glitches when a path output z_j has glitches under a particular test. Conversely, all outputs z_1,..., z_k of N_1 must have no glitches when z_j has no glitches.

Logic transformations are applied repeatedly to subcircuits of a circuit in order to transform more general circuits. When a test-set preserving logic transformation L is applied to a subcircuit N of a circuit C, it is not guaranteed a priori that L also preserves the test set for C. Hence it is necessary to show that the application of L also preserves the test set for C so that the repeated application of L preserves the test set for C.

Consider the two circuits C_1 and C_2 shown in Figure 7 where C_2 is obtained from C_1 by transforming the subcircuit N_1 in C_1 and C_2 using a test-set preserving logic transformation L for path delay faults, i.e. $L(N_1) = N_2$. Note that if L is the De Morgan or the resubstitution transformation, all outputs of N_1 and N_2 have the same normal signal transition, or all outputs have the same signal transition with glitches, for an input change used in delay testing.

Theorem 3 Assume that a subcircuit N_1 of a circuit C_1 is transformed to N_2 by the De Morgan or the resubstitution transformation. A test set T_1 for C_1



(a) (b)

그림 8. ISCAS85 benchmark c432의 일부분: (a) 원본과 (b) 그 RGN 버젼
Fig. 8. A part of ISCAS85 benchmark c432: (a) original and (b) its RGN version.

detects all path delay faults in C_2 where C_2 is the circuit obtained from C_1 by replacing N_1 with N_2.

Proof. Let A_1 denote the part of C_1 excluding N_1. Similarly, let A_2 denote the part of C_2 excluding N_2. Since A_2 is identical to A_1, a test t in T_1 will yield the same signal changes $t*$ at the inputs of N_1 and N_2. Note that t may cause glitches on inputs of N_1 and N_2. In such a case, $t*$ can be viewed as a sequence of two-pattern pairs of N_1 and N_2. With the worst-case delay configuration, $t*$ will produce the same signal transition at the all outputs of N_1 and N_2 with or without glitches. In other words, all corresponding lines in A_1 and A_2 have the same normal signal transitions or the same signal transitions with glitches when t is applied. This implies that T_1 also detects all robust and non-robust path delay faults in C_2. □

Recall that any UGN can be transformed to a BIP network by a sequence of De Morgan and re-substitution[6].

Corollary 1 A test set of a function z that detects all path delay faults in any UGN realizing z also detects all path delay faults in any BIP networks of z.

Proof. This follows Theorm 3 and the fact that any UGN can be transformed to a BIP network by a sequence of De Morgan and re-substitution[6]. □

As can be seen in Section III, UGN has relatively simple structure so that it is easy to analyze. Corollary 1 contributes to reduce the analysis efforts of high-level delay test sets. For example, the universal test set[9] now detects all path delay faults in any UGN as well as BIP networks.

V. Experimental Results

An ISCAS85 benchmark c432 is used to show the correctness of the theorems although the theorems are formally proven. Circuit c432 is transformed to an RGN c432_RGN by using multiple De Morgan and re-substitution transformations on a part of the circuit. For example, a part C_1 of c432 is depicted in Fig. 8a whose corresponding part C_1_RGN of c432_RGN is shown in Fig. 8b. As can be seen, C_1_RGN has inverters only at primary inputs while C_1 in Fig. 8a has inverters after gates. Note that C_1_RGN can be transformed to C_1 by applying 18 De Morgan and 18 re-substitution transformations that are test-set preserving for path delay faults. If C_1 is transformed to C_1_RGN, substitution [1] instead of re-substitution transformation needs to be applied reversely.

Synopsys TetraMax was used to generate a path delay test set T_{RGN} for c432_RGN. Then, T_{RGN} is

applied to c432 and the fault coverage is compared to see if the fault coverage is preserved. As expected, T_{RGN} detects same number of path delay faults for both circuits.

VI. Conclusions

Two logic transformations, De Morgan and re-substitution, are proven to be test-set preserving for path delay faults. In addition, it is shown that the application of De Morgan or re-substitution transformation on a subcircuit of a circuit also preserves the testability of path delay faults. It has been known that a sequence of De Morgan and re-substitution transformations on a UGN results in a BIP network[6]. So, a test set of a function z that detects all path delay faults in any UGN realizing z also detects all path delay faults in any BIP realization of z. For example, the universal delay test set[10] detects all path delay faults in any UGN. Now, we know that the universal delay test set detects all path delay faults in any BIP network too by following the theorems given in this paper.

참 고 문 헌 (References)

[1] M. Batek and J. P. Hayes, "Test set preserving logic transformations," *Proc. of Design Automation Conference*, pp. 454-458, 1992.

[2] S. Charkravarty, "A study of theoretical issues in the synthesis of delay fault testability circuits," *IEEE Trasn. on Computers*, vol. 45, no. 8, pp. 985-991, Aug. 1996.

[3] A. H. El-Maleh and J. Rajski, "Delay fault testability preservation of the concurrent decomposition and factorization transformations," *IEEE Trasn. on CAD*, vol. 14, no. 5, pp. 582-590, 1995.

[4] M. 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 Applications*, vol. 11, pp. 55-67, Aug. 1997.

[5] H. Henster, R. Drechaler, and B. Becker, "On the application of local circuit transformations with special emphasis on path delay fault testability," *Proc. of IEEE VLSI Test Symposium*, pp. 387-392, 1995.

[6] H. Kim and J. Hayes, "Realization-independent ATPG for designs with unimplemented blocks," *IEEE Tras. on CAD*, vol. 20, no. 2, pp. 290-306, 2001.

[7] E. S. Park and M. R. Mercer, "Robust and non-robust tests for path delay faults in a combinational circuit," *Proc. International Test Conf.*, pp. 1027-1034, 1987.

[8] S. M. Reddy, "Complete test sets for logic functions," *IEEE Trans. on Computers*, vol. C-22, no. 11, pp. 1016-1020, Nov. 1973.

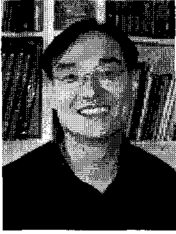
[9] Y. Son and J. Chong, "A New Logic Transformation Method for Both Low Power and High Testability," *Journal of Institute of Electronics Engineers of Korea (IEEK) - SD*, vol. 40, no. 9, pp. 62-71, Sep. 2003.

[10] U. Sparmann, H Muller, and S. M. Reddy, "Universal delay test sets for logic networks," IEEE Trans. on VLSI Systems, vol. 7, no. 2, pp. 156-166, 1999.

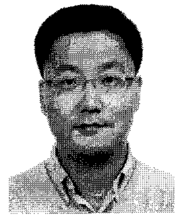
[11] J. Yi and J. P. Hayes, "The coupling fault model for function and delay faults," *Journal of Electronic Testing Theory and Applications*, vol. 21, no. 6, pp. 631-649, Dec. 2005.

[12] J. Yi and J. P. Hayes, "High-level delay test generation for modular circuits," *IEEE Trans. on CAD*, vol. 25, no. 3, pp. 576-590, Mar. 2006.

저 자 소 개



이 준 환(정회원)
1991년 연세대학교 전자공학과 학사
1998년 Univ. of Michigan, Electrical Engineering and Computer Science (EECS) 석사
2002년 Univ. of Michigan, EECS, 박사
1991년~1995년 삼성전자 시스템LSI 연구원
2003년~2008년 삼성전자 통신연구소 수석연구원
2008년~현재 광운대학교 컴퓨터공학과 조교수
<주관심분야 : SoC/MPSoC 구조설계, Computer Vision, 반도체설계, 저전력설계>



이 현 석(정회원)
1992년 KAIST 전기및전자공학과 공학사.
1995년 POSTECH 전자전기과 공학석사.
2007년 Univ. of Michigan, EECS, 박사.
1992년~2008년 삼성전자 통신연구소 수석연구원
2008년~현재 광운대학교 전자통신공학과 조교수
<주관심분야 : DSP architecture, Software defined radio(SDR), Embedded system, VLSI>