

## ブロードサイドテストにおけるN回検出用テストパターンに対するX判定

谷口謙二郎<sup>†</sup> 宮瀬紘平<sup>‡</sup> 梶原誠司<sup>†</sup> 温暁青<sup>†</sup>

<sup>†</sup>九州工業大学 〒820-8502 福岡県飯塚市川津 680-4

<sup>‡</sup>科学技術振興機構 研究成果活用プラザ福岡 〒814-0001 福岡県福岡市早良区百道浜 3 丁目 8-34

E-mail: <sup>†</sup>taniguchi@aries30.cse.kyutech.ac.jp, {kajihara,wen}@cse.kyutech.ac.jp <sup>‡</sup>miyase@fukuoka.jst-plaza.jp

あらまし DSM 技術の進歩により、遅延故障テストの重要性が増している。遅延故障の検出をより確実にするには、各故障を複数回検出するN回検出テストパターンが有効である。しかしN回検出用テストパターンで行う遷移遅延テストは、テストパターン数が増加するため、テストコストの増加が問題となる。その解決策として、テストパターン中にドントケアを判定し、テスト圧縮に利用することが、有効である。本論文では、ブロードサイドテストを前提とした遷移遅延故障用N回検出テストパターンのドントケア判定を行う手法を提案する。ベンチマーク回路に対する実験結果では、提案手法がテスト集合中に効果的にドントケアを判定できることを示す。

キーワード ドントケア判定, 遷移遅延故障, ブロードサイドテスト, N回検出

## Test relaxation for N-detection test patterns in broad-side delay testing

Kenjiro Taniguchi<sup>†</sup> Kohei Miyase<sup>‡</sup> Seiji Kajihara<sup>†</sup> Xiaoqing Wen<sup>†</sup>

<sup>†</sup>Department of Computer Science and Electronics, Kyushu Institute of Technology

680-4 Kawazu, Iizuka, Fukuoka, 820-8502 Japan

<sup>‡</sup>Innovation Plaza, Fukuoka, Japan Science and Technology Agency

3-8-34 Momoti-hama, Sawara-ku, Fukuoka, Fukuoka, 814-001 Japan

E-mail: <sup>†</sup>taniguchi@aries30.cse.kyutech.ac.jp, {kajihara,wen}@cse.kyutech.ac.jp <sup>‡</sup>miyase@fukuoka.jst-plaza.jp

**Abstract** Developing DSM technologies, delay testing is getting more and more important. Generating N-detection test patterns for transitional delay faults can enhance the defect coverage. But the delay testing with an N-detection test set has a problem about test cost. A solution to the test cost problem is to identify Xs in the N-detection test set and to use the Xs for test compaction/compression. In this paper, we propose a method of test relaxation in an N-detection test set for transition delay faults using the broadside testing. Experimental results for benchmark circuits show that the proposed method can identify don't care bits in a given test set effectively.

**Keyword** Don't care identification, Transitional fault, Broad-side testing, N-detection test

### 1. 背景

DSM 技術の進歩に伴い、タイミングに関連した欠陥が VLSI 中で起こる可能性が増加している [1][2]。VLSI 回路に対するテストでは、実速度で動作を保証する実速度 (at-speed) テストが重要になってきている。実速度の性能を検証する機能テストは、高い故障検出率を

得ることが困難なため、スキャンテストが実速度テストにおいても用いられるようになってきている [3]。実速度テストにより、遷移遅延故障 [4] やパス遅延故障 [5] のようなタイミングに関連した故障を検出できるようになる。

スキャン回路に対する実速度テストには、2 つの方法が広く知られている。1 つはスキュードロード

(skewed-load または launch-off-shift) 方式[6], もう 1 つはブロードサイド (broadside または launch-off-capture) 方式[7]と呼ばれる。スキュードロードテストではスキャンシフトにより 2 パターン目を印加する。一方、ブロードサイド方式では、1 パターン目の応答をキャプチャし 2 パターン目とする。クロックタイミングの制御の容易さを考慮した場合、ブロードサイド方式が容易であり、広く用いられている。

また故障動作の多様化により、単に対象とする故障を検出するテストパターンを生成するだけでは、実際に生じる故障の検出には不十分となっている。縮退故障においても、各縮退故障を異なるテストパターンで複数回検出する  $N$  回検出テストの有効性が注目されている[8][9]。それは縮退故障の検出条件が、他の論理故障の検出の必要条件になっていることが多く、 $N$  個の異なるテストパターンでテストを行えば、その他の論理故障の検出する条件を満たすテストパターンが含まれる確率が高くなるからである。このことは遷移遅延故障においても同様で、遷移遅延故障を対象に  $N$  回検出テストを行えば、抵抗性ブリッジのような複数の信号線の値が作用して生じる遅延故障やクロストーク故障などを検出できる可能性が高くなる。

$N$  回検出テストではテストパターン数がほぼ  $N$  倍になるため、遅延故障に対する  $N$  回検出のテストパターン数は膨大になる。テストパターン数の増加はテストコストの増大につながり、 $N$  回検出テストを行う際の解決すべき重要な問題となる。

スキャンテストを行う場合、テストコストの他にも、テスト時の消費電力削減も重要な課題の一つである[10]。縮退故障を対象にした場合、テストパターン数増加や消費電力増加を解決するために、テストパターンのドントケア判定手法[11]が提案されている。テストコスト削減のためのテスト圧縮では、ドントケアが重要な役割を果たす[12][13][14][15][16][17]。また[11]を基に、ブロードサイド方式を前提とした遷移遅延故障に対するテストパターンのドントケア判定法も提案され[18]、判定したドントケアを利用したテスト圧縮により、ドントケア判定の有効性を示している。しかし、これは遷移遅延故障用の 1 回検出テストパターンにしか対応しておらず、 $N$  回検出テストパターンに対してはドントケア判定することはできない。そのため、 $N$  回検出テストパターンに含まれるドントケアを求めることは、 $N$  回検出テストにおける問題を解決するのに有効なアプローチとなる。

本論文では、遷移遅延故障用の  $N$  回検出テスト集合中のドントケア判定手法を提案する。提案手法は、与えられたテスト集合の遷移遅延故障検出率を低下させることなくテスト集合中にできるだけ多くのドントケ

アを見つけることを目的とする。ドントケアを判定する際に、必須故障[19]とそれ以外の故障をそれぞれ処理するが、そのような手法は[11][18]で提案されている。しかし、[11][18]では、それぞれ、縮退故障の遷移遅延故障の 1 回検出テストを対象としており、 $N$  回検出テストは考慮していない。提案手法では、1 回検出用テスト集合における必須故障の概念を  $N$  回検出用テスト集合に対して拡張する。 $N$  回検出テスト集合に対する必須故障を新たに定義し、その概念を用いることにより  $N$  回検出テスト集合のドントケアを最大化する。本論文では、テストパターン中のドントケアを利用した  $N$  回検出テスト集合に対するテスト圧縮についても述べる。実験結果では、 $N$  回検出テスト集合中に判定できたドントケアの割合とテスト圧縮の効果を示す。

## 2. 準備

### 2.1. $N$ 回検出テスト

$N$  回検出テストとは、各故障を検出するテストパターンが少なくとも  $N$  個含まれるテスト集合を使ったテストである。また、そのようなテスト集合を  $N$  回検出テスト集合という。 $N$  個の異なるテストパターンを用いれば、その対象とする故障だけでなく、その故障の検出条件を必要条件に含む他の故障モデルの故障を検出する確率も増すことにつながる。

$N$  回検出テスト生成では、テストパターン数はほぼ  $N$  倍になるものの、 $N$  回テスト生成の処理は単純であり実装も容易であるので、複雑な故障の検出への有効性も実証されている。また  $N$  回検出の対象故障として遷移遅延故障を取扱えば、抵抗性ブリッジに起因する遅延故障などタイミングに関する欠陥の検出の確率を増すことができる。

### 2.2. 必須故障とドントケア判定

テストパターン中のドントケア判定では、故障を必須故障とそれ以外の非必須故障に分類を行い、それに基づきテストパターン中にドントケアの判定を行っている[11]。必須故障は以下のように定義される。

[定義 1: 必須故障]

テスト集合  $T$  において、故障  $f$  がテストパターン  $t \in T$  によって検出できるが、 $T$  の  $t$  以外のテストパターンでは検出できないとき、 $f$  を  $t \in T$  の必須故障という。

必須故障は、2 重検出法に基づく故障シミュレーションを用いて容易に判別できる[19]。必須故障をひとつも含まないテストパターンは、冗長テストパターンと呼ばれ、このようなテストパターンは、テスト集合中から削除されても故障検出率に影響を与えることは

ない。一方、テストパターン  $t$  を変換して必須故障  $f$  が検出されなくなってしまうとそのテスト集合の故障検出率が低下してしまう。そのため[11]では、まずこの必須故障の検出に必要な各テストパターンの論理値割当てを求め、故障シミュレーションを行いその割当てで検出できる故障を求める。その後、検出できない非必須故障に対してのみ、必要な論理値割当てを求める。このように必須故障と非必須故障を区別して処理を行うことが、ドントケア数の最大化において重要となる。

### 2.3. 遷移故障用テストパターンの生成

遷移遅延故障をテストする場合、2パターンでテストすることが必要となる。そこで、非検査対象回路を2時刻に展開した回路モデルとして取り扱う。[18]では、実速度テストを容易にするため、外部入力に2パターン共通の論理値割当てにとした。そのため[18]の手法では、図1は(a)非検査対象回路と、(b)その回路を2時刻展開した回路構造を表す。

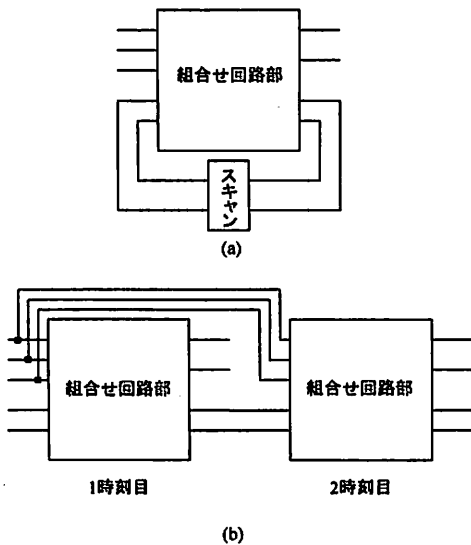


図1: (a) 順序回路 (b) 2時刻展開回路

信号線  $a$  の遷移遅延故障の検出には以下の以下の2つの条件を満たすことが求められる。

- (1) 2時刻目の回路の信号線  $a$  で発生する縮退故障が検出されること
- (2) 信号線  $a$  の論理値は1時刻目と2時刻目で異なること

たとえば、テストパターンが1時刻目に信号線  $a$  の値を論理値0と、2時刻目の回路の信号線  $a$  で発生する0縮退故障を検出されるなら、そのテストパターンは

信号線  $a$  の立ち上がり遷移遅延故障を検出する。図1(b)に示すように、外部入力は入力を両時刻で共有すると、1時刻目と2時刻目で信号値変化を生じさせることができず、構造的に検出が不可能となる故障が存在する。それらはあらかじめ対象故障から除かれている。

## 3. N回検出テストパターンのドントケア判定

### 3.1. N回必須故障

N回検出用テスト集合の場合、必須故障の定義は2.2節で述べた定義とは異なる。そこでN回検出用テスト集合に対するN回必須故障を以下のように定義する。

[定義2: N回必須故障]

N回検出用テスト集合  $T$  において、故障  $f$  がテストパターン  $\{t_1, t_2 \dots t_N\} \subseteq T$  によって検出できるが、 $T - \{t_1, t_2 \dots t_N\}$  のどのテストパターンでも検出できないとき、 $f$  を  $T$  のN回必須故障という。

N回必須故障は、故障シミュレーションにおいて各故障を  $N+1$  回の検出されるまでシミュレートすることで容易に求めることができる。N回検出テストパターンのドントケア判定においてテストパターンの一部のビットがドントケアになったとしても、N回必須故障は、その故障が検出される同じテストパターンで必ず検出されなければならない。一方で、 $N+1$  回以上検出される故障は、他のパターンでN回の検出が保証されるなら、必ず検出する必要はない。

X identification ( $T, F$ )

test set  $T$ , fault list  $F$ ;

```

{
  fault_simulation(T); /* step1 */
  for each test pattern  $t_i$  in  $T$  ( $i=1$  to  $N$ ) {
     $E = \text{collect\_N\_essential\_fault}(t_i)$ ; /* step2 */
     $t_i' = \text{find\_value}(E)$ ; /* step3 */
  }

  reset_fault_list(); /* step4 */
  fault_simulation(T); /* step5 */
  for each test pattern  $t_i$  in  $T$  ( $i=1$  to  $N$ ) {
     $U = \text{collect\_N\_undetected\_fault}(t_i)$ ; /* step6 */
     $t_i' += \text{find\_value}(U)$ ; /* step7 */
    fault_simulation( $t_i'$ ); /* step8 */
  }
  return  $T'$  composed of  $t_i'$ ;
}

```

図2: ドントケア判定処理手順

### 3.2. ドントケア判定手順

ドントケア判定手法のアルゴリズムの概略を図2に示す。まず Step1 としてテスト集合に対して故障シミュレーションを行い、N 回必須故障を求める。次に各テストパターン  $t_i$  に対し、Step2 で  $t_i$  が検出すべき N 回必須故障を故障リスト全体から抽出し、Step3 で N 回必須故障を検出するために必要なテストパターン  $t_i$  の入力値を求める。この入力値で構成されたテストパターンを  $t_i'$  とする。ここで必要とされなかった入力値は、ドントケアの候補となる。Step5 では、N 回必須故障のみの検出を保証したドントケアを含むテストパターン  $t_i$  に対する故障シミュレーションを行い、 $t_i$  によって検出される N 回必須故障以外の故障を調べる。Step6-7 では、Step5 の故障シミュレーションで検出回数が N 回に満たない故障を N 回検出するためのテストパターン中の入力値を求める。Step5 の故障シミュレーションによって、Step6-7 で扱う故障数が削減されているため入力値割当が新たに必要となる故障は少なくでき、ドントケア数を少なくする可能性が低くなる。

### 3.3. 処理例

ドントケア判定の処理を N=2 の簡単な例を用いて説明する。図 3(a) のようなドントケアを含まない 2 回検出用テスト集合が与えられたとする。そのテスト集合に対して故障シミュレーションを行い、それぞれの故障の検出回数を求めると表 1 の(a)の欄ようになる。2 回しか検出されない故障は  $f_4$  と  $f_5$  なので、これらが N 回必須故障となる。

そして、ドントケア判定を行い N 回必須故障の検出に必要な割当てを求め、その割当てだけを論理値として残す。そのテスト集合は、図 3 (b) のようになる。そのテスト集合で故障シミュレーションを行い、検出される故障を求めると、表 1(b) のようになる。図 3 (b) のテストパターンで選ばれた論理値は、N 回必須故障を検出するために求めたものであるが、 $f_1$  と  $f_2$  のように、必須故障に対する割当てで、それ以外の故障も見つかることがある。そして、2 回以上検出していない故障に対する必要な割当てを求め、論理値として残す。元の図 3(a) のテスト集合で 2 回以上検出された故障がすべて 2 回以上検出されたなら終了する。この例では求めるテスト集合は図 3(c) のようになる。

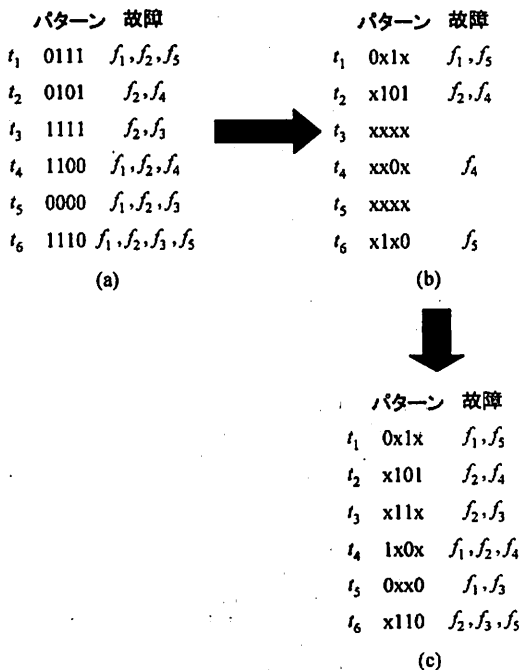


図 3: X 判定処理例

表 1: 故障検出回数

故障	検出回数		
	(a)	(b)	(c)
$f_1$	4	1	3
$f_2$	6	1	4
$f_3$	3	0	3
$f_4$	2	2	2
$f_5$	2	2	2

### 4. テストコンパクション

テスト集合中の判定されたドントケアは様々な用途に用いることができる。本論文では、ドントケアを用いたコンパクションによるテスト圧縮について述べる。

ドントケアを含むテスト集合が与えられると、いくつかのテストパターンは故障検出率を低下させることなくマージすることができる。これを静的圧縮という。静的圧縮を行うと、2 つかそれ以上のテストパターンがマージされ、1 つのテストパターンとして取扱うことができ、それによりテストデータ量が削減できる。

パターン	故障		パターン	故障	
$t_1$	0x1x	$f_1, f_3$	$t_1'$	0110	$f_1, f_2, f_3, f_5$
$t_2$	x101	$f_2, f_4$	$t_2'$	x101	$f_2, f_4$
$t_3$	x11x	$f_2, f_3$	$t_3'$	1x0x	$f_1, f_2, f_4$
$t_4$	1x0x	$f_1, f_2, f_4$	$t_4'$	x110	$f_2, f_3, f_5$
$t_5$	0xx0	$f_1, f_3$			
$t_6$	x110	$f_2, f_3, f_5$			

図4：テスト圧縮例（2回検出テストパターン）

図4にテスト圧縮の処理の例を示す。この例では、2回検出を行うテストパターンに対するテスト圧縮を行う。この例では、2回しか検出されていない故障は $f_4$ と $f_5$ である。そのためこの2つ故障はN回必須故障であり、互いに同じN回必須故障を含むテストパターンはマージすることは、故障検出率の低下につながるので行うことはできない。これは、通常の(N=1)のテストパターンにおける静的圧縮ではありえないケースであり、N回検出テストの静的圧縮に固有の制約である。テストパターン $t_1(0X1X)$ はテストパターン $t_3(X11X)$ とマージすることが可能である。2つのパターンをマージしてテストパターン(011X)が導出される。そしてそのテストパターン(011X)はさらにテストパターン $t_5(0XX0)$ とマージすることができ、パターン(0110)が導出される。このマージにより故障 $f_3$ の検出回数は3回から2回に削減され、故障 $f_3$ もN回必須故障となる。このようにN回検出テストパターンに対してテスト圧縮を行う場合は、N回必須故障がパターンのマージの度に関わる。またマージされたパターン(0110)は、さらにテストパターン $t_6(X110)$ ともパターンはマージすることができるが、そのマージを行うと故障 $f_3$ と $f_5$ の検出回数が2回から1回に削減されてしまい、2回検出を満たさなくなるので、このマージは行うことができない。パターン $t_2(X101)$ と $t_4(1X0X)$ についてもパターン同士はマージ可能であるが、このマージを行うと故障 $f_4$ の検出回数が2回検出を満たさなくなるので、マージすることができない。

このようにN回検出のテスト圧縮は、パターン同士のマージ可能性と共に、検出故障同士のマージ可能性の判定も行わなければならないので、1回検出の場合に比べて、圧縮の効果が小さくなることが予測される。この例では、最終的にテスト集合( $t_1', t_2', t_3', t_4'$ )が求まる。圧縮後のそれぞれの故障検出回数は、 $f_2$ の検出回数が4回でその他の故障は全て検出回数が2回である。よってこのテスト集合の故障検出率は、元のテ

スト集合( $t_1, t_2, t_3, t_4, t_5, t_6$ )と同じである。しかしテストパターン数は6から4に削減されているので、テストデータ量を削減することができている。

### 5. 実験結果

提案手法をPC(OS: FreeBSD 4.11 Release, CPU: Athlon XP 1800+ 1.53GHz, memory: 768MB)上にCプログラミング言語を用いて実装し、ISCAS'89ベンチマーク回路に対して行った。実験で使用したテスト集合は、以下の方法で生成している。まず1つの対象故障を選択し、その故障に対してテストパターンの生成を行い、生成したテストパターンで故障シミュレーションを実行する。これにより、そのテストパターンで検出される故障はテスト生成を行う必要はなくなり、テスト生成を行う回数を少なくする。ある故障が $k+1$ 回の対象故障となるのは、すべての故障が $k$ 回対象故障として処理された後である。またこの $k$ 回目の対象故障とされた時、すでに検出がN回されているか、すでにテスト生成不可と判定されている場合は、テスト生成の処理はされない。この処理をテスト生成が可能であり、かつN回検出を満たしていない対象故障がなくなるまで実行することにより、テスト集合を生成している。

表1にN=1,2,5のテスト集合に対してドントケア判定を行った実験結果を示す。表1の最初の列は回路名を表す。次の3列は、1回、2回、5回検出それぞれのテストパターン数を表している。そして、次の3列がX判定の結果を表し、テストパターン中で判定されたドントケアの平均の割合をそれぞれ表している。

表2：X判定の実験結果

回路名	#vec.			X判定		
	1回	2回	5回	1回	2回	5回
s5378	402	729	1725	85.08	60.47	59.87
s9234	992	1789	4074	86.62	36.21	35.6
s13207	1030	1837	4222	96.51	53.8	53.63
s15850	772	1426	3388	94.36	40.27	40.2
s35932	193	401	811	91.4	92.13	74.39
s38417	2360	4282	10263	96.35	65.95	63.46
s38584	2411	4242	9753	96.91	34.2	33.84
平均				92.46	54.72	51.57

結果を見ると与えられたテスト集合に対して、提案手法は2回検出以上のテスト集合に対しては1回検出の場合ほどドントケアを判定できていないことがわか

る。しかし検出回数が5回になっても約50%のドントケアを判定できており、ドントケアを利用したアプリケーションを行う場合に十分な割合と考えられる。

表3：テスト圧縮の実験結果

回路名	1回		2回		5回	
	#vec.	%	#vec.	%	#vec.	%
s5378	272	67.7	564	77.4	1403	81.3
s9234	619	62.4	1697	94.9	3885	95.4
s13207	555	53.9	1544	84.1	3766	89.2
s15850	478	61.9	1266	88.8	3098	91.4
s35932	72	37.3	155	38.7	422	52.0
s38417	1352	57.3	3565	83.3	9021	87.9
s38584	1499	62.2	3813	89.9	9245	94.8
平均		57.5		79.5		84.6

表3にテスト集合の圧縮結果を示す。#vec.は圧縮後のそれぞれのテストベクトル数を、%は元のテストベクトル数に対する割合を示す。結果を見ると、2回以上検出テストパターンに対しても、テスト集合をおよそ80%まで圧縮することができたことを示す。

## 6. 結論

本論文では、遷移遅延故障用のN回検出テスト集合に対してドントケア判定を行う手法を提案した。ISCAS'89 ベンチマーク回路に対する実験結果では、提案手法がテスト集合中に平均して2回検出以上でもテスト集合中に50%以上のドントケアを判定できたことを示した。これは1回検出のテストパターンに対する結果より、判定されるドントケアの割合は少ない。これはN回対象とすることにより、テスト生成の対象とする故障を選択する順番による影響がなくなるためと考えられる。

また本実験では、判定されたドントケアをテストデータ量削減のために用いた。実験結果はドントケア判定されたテスト集合に対して、2回以上検出のテストパターンに対してもテスト圧縮の効果があることを示せた。

今後の課題としては、N回テストパターンを生成する際のテスト生成の効率化などが挙げられる。

## 文 献

[1] S. Natarajan, M.A. Breuer, S.K. Gupta, "Process Variations and Their Impact on Circuit Operation,"

in Proc. IEEE Int. Symp. On Defect and Fault Tolerant in VLSI Systems, pp.73-81, 1998.

- [2] R. Wilson, "Delay-Fault Testing Mandatory, Author Claims," EE Design, Dec. 2002.
- [3] J. Savir and S. Patil, "Scan-Based Transition Test", IEEE TCAD, pp. 1232-1241, 1993.
- [4] J. A. Waicukauski, E. Lindloom, B. K. Rosen and V. S. Iyengar, "Transition Fault simulation," IEEE Design and Test of Computers, pp.32-38, 1987.
- [5] G. L. Smith, "Model for Delay Faults Based Upon Paths," Proc. ITC, pp.342-349, 1985.
- [6] J.Savir, "On broad-side delay testing," Proc. VLSI Test Symposium, pp.284-290, 1994.
- [7] J. Savir, "Skewed-Load Transition Test: Part I, Calculus," International Test Conference, pp. 705-713, 1992.
- [8] S.M.Reddy, I.Pomeranz, S.Kajihara, "Compact Test Sets for High Defect Coverage," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol. 16, No. 8, pp.923-930, Aug. 1997.
- [9] B.Benware, et al.. "Impact of Multiple-Defect Test Patterns on Product Quality," Proc. Of International Test Conference, pp.1031-1040, 2003.
- [10] W. LI, Sudhakar M. Reddy, I. Pomeranz, "On Test Generation for Transitional Faults with Minimized Peak Power Dissipation," DAC., pp504-509, June. 2004.
- [11] S. Kajihara, K. Miyase, "On Identifying Don't Care Inputs of Test Patterns for Combinational Circuits," ICCAD-2001, pp.364-369, Nov. 2001.
- [12] A. El-Maleh and A. Al-Suwaiyan, "An Efficient Test Relaxation Technique for Combinational & Full-Scan Sequential Circuits," VLSI Test Symposium, pp. 53-59, April 2002.
- [13] A. El-Maleh and K. Al-Utaibi, "An efficient test relaxation technique for synchronous sequential circuits," VLSI Test Symp., pp.179-185, April 2003.
- [14] Y. Higami, S. Kajihara, S. Kobayashi, Y. Takamatsu, I. Pomeranz "A Method to Find Don't Care Values in Test Sequences for Sequential Circuits," International Conf. on Computer Design, pp. 397-399, Oct. 2003.
- [15] B. Koenemann, et. al., "A Smart BIST Variant Guaranteed Encoding," 10th Asian Test Symposium, pp. 325-330, Nov. 2001.
- [16] H.-G. Liang, S. Hellebrand, H.-J. Wunderlich, "Two-dimensional Test Data Compression for Scan-based Deterministic BIST," 2001 International Test Conf., pp. 894-902, Nov. 2001.
- [17] R. Sankaralingam, R. R. Oruganti, and N. A. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation," 18th VLSI Test Symposium, pp. 35-40, 2000.
- [18] 谷口謙二郎, 官瀬紘平, 梶原誠司, 温曉青, "フルスキャン順序回路用2パターンテストに対するドントケア判定," DA シンポジウム, pp.103-108, 2006年7月.
- [19] S. Kajihara, I. Pomeranz, K.Kinoshita, and S. M. Reddy, "Cost-effective generation of minimal test sets for stuck-at faults in combinational logic circuits," IEEE Trans. Computer-Aided Design, vol. 14, pp. 1496-1504, Dec. 1995.