社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2004－SLDM－117 (29)
2004／12／2

入力ベクトルからの信号値を正当化する最小キューブ抽出

宮瀬 紘平† 永山 忍† 梶原 誠司† 温 暁青† レディ スダーカ‡

†九州工業大学 〒820-8502 福岡県飯塚市川津 680-4
‡アイオワ大学 Iowa City, IA 52242, U.S.A.

E-mail: †miyase@aries30.cse.kyutech.ac.jp, †nagayama@aries02.cse.kyutech.ac.jp,
†{kajihara, wen}@cse.kyutech.ac.jp, ‡reddy@engineering.uiowa.edu

あらまし ある入力ベクトルによって組合せ回路内部の信号値が正当化されるとき，その内部信号値を正当化する外部入力値はキューブで表すことができる．本論文では，ある入力ベクトルから内部信号値を正当化する最小キューブ抽出手法を提案する．正当化する信号値が一つの場合を基に，複数の信号線を同時に正当化する最小キューブ抽出手法へ拡張する．複数信号値を正当化する場合は，正当化可能なすべてのキューブを BDD で表現し，キューブ抽出問題を BDD 上での最短路問題に帰着させる．提案手法は，テストベクトルの特定ビット数が最小であるテストキューブの探索に使用でき，そのようなテストキューブは例えばテストデータコンプレッションの効果を高めるのに有効である．

キーワード 正当化操作，BDD，最短路問題，テストデータコンプレッション

On the Extraction of a Minimum Cube to Justify Signal Line Values

Kohei MIYASE† Shinobu NAGAYAMA† Seiji KAJIHARA† Xiaoqing Wen†
and Sudhakar M. REDDY‡

†Kyushu Institute of Technology 680-4 Kawazu, 820-8502 Japan
‡University of Iowa Iowa city, IA 52242, U.S.A.

E-mail: †miyase@aries30.cse.kyutech.ac.jp, †nagayama@aries02.cse.kyutech.ac.jp,
†{kajihara, wen}@cse.kyutech.ac.jp, ‡reddy@engineering.uiowa.edu

Abstract When there are internal logic values in a combinational circuit that must be justified by a given input vector, necessary input values to justify them are expressed as a cube. In this paper we propose methods to extract a cube with the minimum number of specified values from an input vector. One is the case of justifying a single line value and the other is the case when the number of signal values to be justified is more than one. In the latter case, we utilize a BDD to extract the cube, and we reduce the problem to a shortest path problem. We also present how to reduce the size of BDDs used. The method can be used for finding test cubes with minimal numbers of specified bits in test vectors. Such test cubes can be used to achieve higher test data compression.

Keyword Justification, BDD, Shortest path problem, Test data compression

1. はじめに

論理回路のテストにおけるテストデータ量の削減やテスト品質の向上に，テストベクトル中の未設定信号値は重要な役割を果たしている．これまで提案されているテストコンプレッション手法の多くは，テストベクトル中の未設定信号値に依存しており，未設定信号値を用いることなしで高いコンプレッション効果を得ることは困難である[1][2][3][4].

組合せ回路に対して，ある入力ベクトルにより全ての内部信号値が決まっているとき，各信号線の信号値の決定には入力ベクトル中の一部の値だけが関与しており，そのような入力値はキューブで表すことができる．一般に，ある信号値設定要求を満たすキューブは多数存在する．その中からできるだけ少ないリテラル数のキューブを抽出する手法は，テストベクトル中のドントケア(X)を判定する際に考えられてきた[5][6]. キューブを抽出する際に，[5]の手法では限定含意操作と限定正当化操作を使用し，[6]の手法ではテスト容易性尺度を利用している．しかし，[5][6]の手法で抽出されたキューブは，リテラル数が最小であることを保証されていない．

与えられた入力ベクトルから最小リテラル数のキューブを抽出する問題は二つの場合に分割することができる．一つは回路中の一つの信号値を正当化する場

合であり，他の一つは複数の信号値を同時に正当化する場合である．本論文では，これらの問題をそれぞれ単一信号値正当化と複数信号値正当化と呼ぶことにする．単一信号値正当化では，まずその信号値の正当化が可能な全てのキューブを算出する．一つのキューブは，積項で表すことができるので，全てのキューブは積和形論理式で表現できる．その積和形論理式のうち最小リテラルで構成される積項が求めるキューブとなる．複数信号値正当化では，正当化の対象となる各信号値に対する単一信号値正当化の処理に基づいて，複数信号値を正当化する入力組合せを AND-OR-AND 形の論理式で表現する．一般にその論理式を満足する最小キューブを求めるには多大な計算量が必要になる．本論文では，AND-OR-AND 形で表される論理式を BDD[8]によって表現し，リテラル数最小のキューブを BDD 上での最短路問題に帰着させて求める手法を提案する．少ないメモリで BDD を構成するには，BDD のサイズを削減することが重要である．また，BDD のサイズを削減すれば，最短路問題を解く際の処理時間も削減できる．本論文では，変数の順序付けと入力変数を削減することで BDD のサイズを削減する．

最小キューブを抽出する手法は，テストベクトル中の入力値を X に変換する場合に有効である[5][6]．本論文では，ISCAS ベンチマーク回路に対する実験結果により，提案手法を[5]の手法に組み込んだ場合の判定される X の割合を示し，提案手法の有効性を示す．

本論文は以下のように構成される．2 節では，本論文で用いる用語の定義と問題設定を行う．3 節で単一信号値正当化について述べ，4 節で複数信号値正当化について述べる．どちらの場合も得られるキューブのリテラル数は理論的に最小である．また，4 節では，BDD のサイズの削減についても述べる．5 節では，提案手法をドントケア判定手法に適用し，ISCAS ベンチマーク回路に対する実験により提案手法の有効性を示す．最後に 6 節でまとめを行う．

2. 準備

2.1. 諸定義

定義 1: 二分決定グラフ（BDD: Binary Decision Diagram）は，論理関数を表現する非循環有向グラフ（DAG: Directed Acyclic Graph）であり，与えられた論理関数にシャノン展開を繰り返し適用することにより，その BDD を得ることが出来る．BDD は，論理値 0，1 を表現する二つの終端節点と，論理変数 x を表現する非終端節点から構成され，各非終端節点は，変数の値に対応する二つの枝（0-枝，1-枝）を持つ．

定義 2: BDD において，根から終端節点までの経路を BDD のパスという．BDD のパスは枝と非終端節点の繰り返しから成り，本論文ではパス上の 1-枝の数をパス長とする．

定義 3: BDD の全てのパスにおいて，変数順序が同じとき，その BDD を OBDD（Ordered BDD）という．特に，同型な部分グラフおよび 0-枝と 1-枝が同じ節点に繋がっている非終端節点が OBDD 内に存在しないとき，その OBDD を，ROBDD（Reduced OBDD）とよぶ．

本論文では，ROBDD のみを扱うため，以下 BDD は ROBDD を意味する．

2.2. 問題設定

本論文では，次のような問題を取り扱う：組合せ回路 C，入力ベクトル v，および C 中の信号線の集合 $L = \{l_1, l_2, \ldots, l_n\}$ が与えられたとき，最小数リテラルのキューブ u を出力する．u は L 中の全信号線上の信号値を正当化し，u は v を被覆する．ここでは，v 中の全ての入力値は 0 か 1 の信号値に設定されているとするが，v 中の入力値が部分的に設定されている場合でも提案手法は適用可能である．

図 1 に例を示す．組合せ回路 C，入力ベクトル $v = <a, b, c, d, e> = <1, 0, 0, 1, 0>$，信号線集合 $L = \{h, k, l\}$ が与えられたと仮定する．この問題の解はキューブ $u = (\bar{b}\ \bar{c})$ となる．$b = c = 0$ は信号線 h, k, l の信号値を正当化する．一方で一つだけのリテラルで構成されるキューブで L の信号値全てを正当化できるものはない．

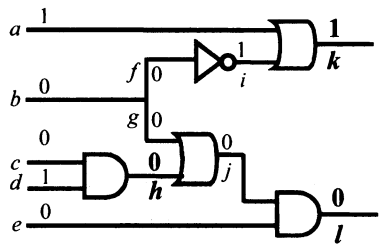

図1: 組合せ回路 C, 入力ベクトル v, 信号線集合 L

3. 単一信号値正当化

まず初めに $|L|=1$，即ち一つの信号線 l 上の信号値を正当化する最小キューブ u を入力ベクトル v から求める手法を提案する．この問題を単一信号値正当化と呼ぶことにする．単一信号値正当化問題を解く手法は，次のように三つのステップで構成される．

Step 1: l 上の信号値の正当化に関わる信号値を持つ信号線を l から入力側に向かって全てマークする．
Step 2: l を含む全てのマーク済み信号線上の信号値を

正当化するキューブを入力側から出力側に向かって順次計算する．

Step 3: l に対するキューブの中から，最小数のリテラルを持つキューブを抽出する．

キューブは積項で表すことができるので，l 上の信号値を正当化する全てのキューブの組合せは積和形論理式で表すことができる．ゲート出力を正当化するキューブは，ゲート入力に対するキューブより演繹される．それゆえ，外部入力から l の方向に各信号線を正当化するキューブを順に計算することにより，最終的に l を正当化するキューブを求めることができる．図2に例を示す．入力ベクトル $v = <a, b, c, d, e> = <1, 0, 0, 1, 0>$ と信号線 l に対する単一信号値正当化を考える．Step 1では，まず l, j, e, g, h, b と c をマークする．マークした信号線を図中の太線で示す．ただし，d はマークしない．なぜなら d は h の正当化に必要ないからである．次に，マークした信号線を正当化するキューブを算出する．外部入力を正当化するキューブは外部入力自身なので，Step 2では l に対する全キューブが外部入力から l の方向に計算される．図3中の括弧の中に，キューブに含まれる入力変数を示す．ただしキューブの算出において，変数の極性は無視できる．なぜなら各変数に対して高々一つのリテラルしか現れないからである．それゆえ，キューブに含まれる入力変数を表す積和形論理式は単調増大関数となる．その積和形論理式のうち最小リテラルで構成される積項が求めるキューブとなる．図3の場合，l を正当化する必要十分な入力変数は $b \cdot c \vee e$ である．つまり，関数 $F = b \cdot c \vee e = 1$ を満たす変数が，リテラルとしてキューブに含まれる．図3の例では，入力ベクトルにおける e の信号値が0であることと合わせて，単一信号値正当化問題の解としてキューブ $\bar{e}$ が得られる．

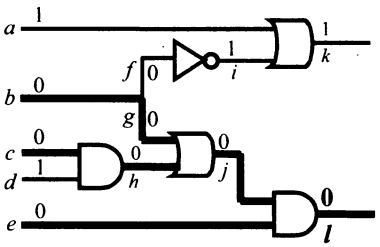

図2: lの正当化に対する信号線のマーク

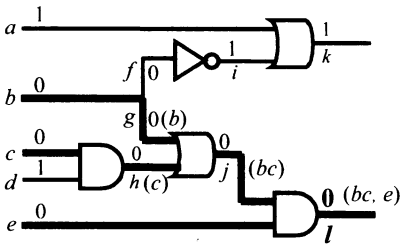

図3: キューブの計算

4. 複数信号値正当化

本節では $|L|>1$，即ち複数の信号値を正当化する場合について述べる．この問題を複数信号値正当化と呼ぶことにする．

4.1. 例

図4に例を示す．入力ベクトル $v = <a, b, c, d, e> = <1, 0, 0, 1, 0>$ と信号線 h, k, l に対する複数信号値正当化を考える．各信号線に対して単一信号値正当化を適用すると，h, k, l を正当化するキューブに含まれる入力変数は，それぞれ (c), $(a \vee b)$, $(b \cdot c \vee e)$ の論理式で表すことができる．複数信号値正当化を行う場合は，単一信号値正当化によって得られた各積和形論理式を同時に充足することが求められるので，AND-OR-AND 形に結合した論理式で表現できる．図4の例では，$(c) \cdot (a \vee b) \cdot (b \cdot c \vee e)$ となる．単一信号値正当化と同様に，関数 $F = (c) \cdot (a \vee b) \cdot (b \cdot c \vee e) = 1$ を満たす変数が，リテラルとしてキューブに含まれる．AND-OR-AND の論理式を積和形論理式に展開すれば，その積和形論理式のうち最小リテラルで構成される積項が求めるキューブとなる．しかし，一般にそのような展開は多大な計算量を必要とする．そこで，本論文では，複数の信号値を正当化するキューブを BDD[8]で表現する．そして，そのようなキューブを求める問題を BDD 上で最短路問題に帰着させ，キューブを求める．

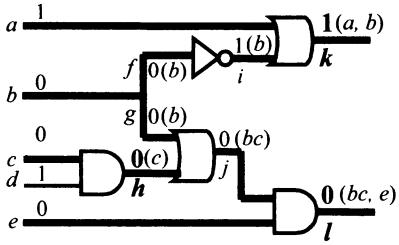

図4: 複数信号値正当化

4.2. BDD を用いる解法

単一信号値正当化と同様に，得られた AND-OR-AND 形の論理式は，信号線集合 L の信号値を正当化する入力変数の組合せを表している．得られた論理式を BDD で表現すると，全てのキューブが BDD の根から1終端節点へのパスで簡潔に表される．そのパス上の1-枝は，その論理式を充足するキューブに含まれる変数である．したがってパス上の1-枝を数えることによって，

キューブのリテラル数は容易に計算できる．図5は，関数 $F = (c) \cdot (a \vee b) \cdot (b \cdot c \vee e) = 1$ に対するBDDを示す．信号線集合 L の信号値を正当化する入力変数に対応する1-枝を実線で表す．信号線 a, b, c の信号値はそれぞれ 1, 0, 0 なので，キューブ $u = (a\ \overline{b}\ \overline{c})$ は $L = \{h, k, l\}$ 上の信号値を正当化する．しかし，このキューブのリテラル数は最小ではない．信号値設定要求を満たすキューブは一般に多数存在する．次節では，最小数のリテラルを持つキューブの抽出法について述べる．

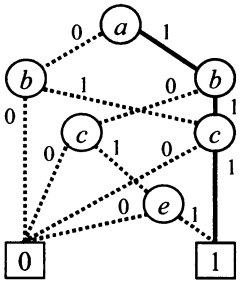

図5: (c)·(a ∨ b)·(b·c ∨ e) = 1 に対するBDD

4.3. 最短路問題

全ての複数の信号値を正当化するキューブは，BDDの1終端接点から根までのパスとして表されており，リテラル数はそのパス上の1-枝の数と対応する．それゆえ，複数の信号値を同時に正当化するキューブを求める問題は次のようにBDD上の最短路問題に帰着できる：

複数信号値正当化に対応する最短路問題：単調増大関数を表すBDDが与えられたとき，BDDの根と1終端節点との間のパスで最小数の1-枝を持つパスを探索する．このように，BDDを重みつきグラフとして考え，BDDの枝のラベルを重みと見なせば，この問題はBDD上の最小の重みを持つパスを探索する問題と等価であることが分かる．この問題はBDDが非循環有効グラフであることから $O(E)$ で解くことができる．ただし E はBDDの枝の数である[9].

図6の実線が，上記で述べた最短路に対応する．つまり，複数の信号値を同時に正当化する最小数のリテラルを持つキューブは $u = (\overline{b}\ \overline{c})$ である．

BDDの構造は変数順序によって決まる．一般に，最短路問題の厳密な解を得るには変数順序を最適化する必要がある．しかし，複数信号値正当化で取り扱うBDDに関しては，次の定理が示すように変数順序を考慮する必要はない．

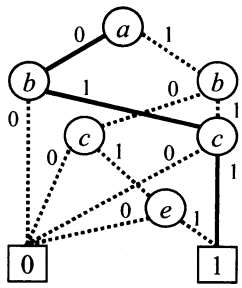

図6: (c)·(a ∨ b)·(b·c ∨ e) = 1 に対するBDD

定理 1：単調増大関数に対するBDDでは，1-枝の数が最小であるパスは，変数順序に依存しない．

定理1の証明はページ数の制限により割愛する．

4.4. BDDのサイズ削減

少ないメモリでBDDを構成するには，BDDのサイズを削減することが重要である．また，BDDのサイズの削減が，最短路問題を解く際の処理時間を削減することは明らかである．本論文では，変数の順序付けと入力変数の削減でBDDのサイズを削減する．

定理1より，変数順序を変えても最短路問題の解の精度が失われることはない．BDDのサイズは変数順序に強く依存するので，BDDのサイズ削減を考える上で変数順序の最適化は大変重要である．変数順序最適化アルゴリズム(Shifting algorithm[10])はよく利用されるアルゴリズムであるが，計算時間が大きくなる．本論文では，変数の出現頻度による単純な順序付けを行う．ここで，再度関数 $F = (c) \cdot (a \vee b) \cdot (b \cdot c \vee e) = 1$ を考える．変数の出現頻度を $freq()$ とすると，それぞれ $freq(c) = 2$, $freq(a) = 1$, $freq(b) = 2$, $freq(e) = 1$ となる．これもとに変数を頻度の高い順に並べ変数順序(c, b, a, e)が得られる．図7はこの変数順序をもとに構築したBDDである．なお，図の例ではBDDのノード数は最小であるが，このアルゴリズムは，必ずしもBDDのノード数が最小になることを保証しない．

入力変数の削減もBDDのサイズ削減に貢献する．

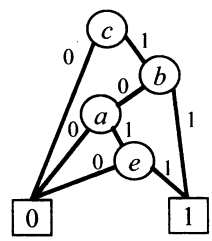

図7: 変数の順序付け

BDDを構成する前に，複数の信号値を同時に正当化するキューブが必ず含んでいる入力変数を見つけ，関数から全て取り除く．そのような入力変数は，AND-OR-AND 形論理式において入力変数を一つしか持たない節となるため，その判別は容易である．関数 $F = (c) \cdot (a \vee b) \cdot (b \cdot c \vee e) = 1$ の例では，変数 c をこの論理式から取り除く．$F = 1$ を満たすには $c = 1$ が必要であり，c はどのキューブにもリテラルとして含まれることが明らかだからである．c を取り除いたBDDを図8に示す．図8のBDDに対する最短路問題の解は b になり，最終的にキューブ $u = (\bar{b}\ \bar{c})$ が得られる．

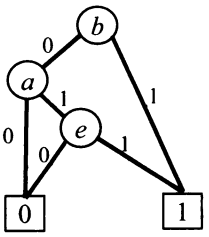

図8: $(a \vee b)(b \vee e) = 1$ に対するBDD

4.5. 処理手順

組合せ回路 C，入力ベクトル v そして C 中の信号線の集合 L が与えられたとき，複数信号値正当化が最小リテラル数のキューブ u を抽出する処理手順を以下にまとめる：

Step 1: L 中の信号線 l_i $(1 \le i \le |L|)$ に対して，

1.1 l_i 上の信号値の正当化に関わる信号線を全てマークする．

1.2 l_i を含む全てのマーク済み信号線の信号値を正当化するキューブを計算し，積和形論理式で表す．

Step 2: L 中の各 l_i $(1 \le i \le |L|)$ に対する積和形論理式をAND-OR-AND 形に結合する．

Step 3: 得られた論理式の除去可能な変数を式から取り除く．

Step 4: Step 3 で得られた AND-OR-AND 中の変数の出現頻度を計算し，降順に並び替える．

Step 5: Step 4 で得られた変数順序をもとに BDD を構築する．

Step 6: BDD 上で最短路問題を解く．

Step 7: 最短路問題の解と Step 3 で取り除いた変数から，複数の信号値を同時に正当化するキューブを出力する．

5. 応用と実験結果

リテラル数の少ないキューブ抽出は，故障検出率を低下させない条件下でテストベクトル中にドントケア(X)を見つける処理に利用されている[5][6]．そのようなキューブを抽出する際に，[5]の手法では[5]の論文中で定義されている限定含意操作と限定正当化操作を使用し，[6]の手法ではテスト容易性尺度を利用している．しかし，[5][6]の手法は X を最大数見つけることを保障していない．本節では，手法[5]に最小キューブ抽出手法を組み込み，その効果をベンチマーク回路に対する実験により示す．

提案手法を Dual Athlon MP 2000+，512MB メモリの計算機上で C 言語により実装し，[5]の手法に組み込んで判定できる X の数を調べる実験を ISCAS'85，ISCAS'89 のベンチマーク回路に対して行った．X を判定するテストパターンは，テスト圧縮技術を含む[11]の ATPG によって生成されたものである．

表1に，判定された X の割合を示す．最初の三つの欄はそれぞれ，回路名，外部入力数，テストベクトル数を表す．次の"%X-ave"以下の三つの欄は異なる3種類の手法を用いて判定した X の割合を示す．最初の手法は，手法[5]によるものである．次の"single"は，単一信号値正当化を逐次的に手法[5]に適用した場合の X の割合を示している．よって，単一信号値正当化は，複数の信号値を正当化する場合，最小リテラル数のキューブ抽出を保証しない．"multiple"以下に示す割合は，複数信号値正当化による結果を示す．最後の3つの欄は，各手法に対する CPU 時間を示す．

表1より，単一信号値正当化を逐次的に適用した場合，および，複数信号値正当化を適用した場合において，手法[5]が判定する X の数を増加させることができた．また，複数信号値正当化は単一信号値正当化を逐次的に適用するよりも効果的である．上記で述べたように，単一信号値正当化は，複数の信号値を正当化する場合に最小リテラル数のキューブ抽出を保証しないからである．また，そのため，c432, c880, s1494 の回路に対しては，単一信号値正当化を用いた場合より手法[5]の判定する X の割合が多いと考えられる．s38417 の回路では，構築する BDD のノード数が増加し，必要メモリが使用計算機の実装メモリ容量を超えたため複数信号値正当化を適用できなかった．手法[5]に提案手法を組み込んだ場合，CPU 時間は増加するが，その増加量は許容範囲であると考えられる．

6. まとめ

本論文では，組合せ回路内部の信号値を正当化する最小リテラルを持つキューブを入力ベクトルから抽出する手法を提案した．単一信号値正当化は，一つの信号値を正当化するキューブを抽出する．複数信号値正当化は，複数の信号値を正当化するリテラル数最小の

表1:圧縮テスト集合に対する実験結果

circuits	#PIs	#tests	%X-ave			time(sec)		
			[5]	single	multiple	[5]	single	multiple
c432	36	28	47.1	45.6	49.5	0.00	0.03	0.10
c499	41	52	0.6	0.6	0.6	0.02	0.02	0.06
c880	60	21	34.4	33.2	34.4	0.02	0.03	0.06
c1355	41	84	0.0	0.0	0.0	0.04	0.07	0.15
c1908	33	106	17.0	20.6	20.8	0.13	0.23	0.42
c2670	233	45	70.7	71.2	71.3	0.26	0.47	0.76
c3540	50	93	53.3	53.3	54.2	0.84	1.33	2.44
c5315	178	46	61.4	61.4	61.5	0.65	3.36	7.97
c6288	32	14	0	0	0	0.47	2.73	5.96
c7552	207	75	54.5	54.7	54.8	1.90	6.63	16.12
s1238	32	125	56.5	56.7	56.9	0.06	0.13	0.22
s1423	91	24	42.0	42.6	42.9	0.04	0.07	0.14
s1494	14	100	27.4	26.6	27.4	0.06	0.12	0.22
s5378	214	100	73.3	73.4	73.7	1.36	3.54	7.50
s9234	247	111	69.2	69.3	69.5	2.81	11.34	27.27
s13207	700	235	92.0	92.4	92.4	8.27	30.83	70.58
s15850	611	97	77.3	77.6	77.4	4.81	27.79	68.42
s35932	1763	12	36.2	36.2	36.2	3.06	116.74	291.59
s38417	1664	87	74.8	76.9	NA	12.20	134.92	NA
s38584	1464	114	81.1	81.0	81.3	15.75	190.52	528.73
Average			48.4	48.7				

キューブを BDD の最短路問題を解くことにより抽出する．実験結果では，提案手法がテストベクトル中に判定できるXの数を増加することに役立つことを示した．

謝辞 本研究の一部は，笹川科学研究助成，科学研究費補助金基盤研究(c)(課題番号 16500036)，および，日本学術振興会二国間交流事業アメリカとの共同研究によるものである．

文　献

[1] B. Koenemann, et. al., "A Smart BIST Variant Guaranteed Encoding," 10th Asian Test Symposium, pp. 325-330, Nov. 2001.

[2] H. Ichihara, K. Kinoshita, I. Pomeranz and S. M. Reddy, "Test Transformation to Improve Compaction by Statistical Encoding," International Conf. on VLSI Design, pp. 294-299, Jan. 2000.

[3] A. Jas, J. Ghosh-Dastidar, N. A. Touba, "Scan vector Compression/Decompression Using Statistical Coding," VLSI Test Symposium, pp. 114-120, April 1999.

[4] A. Chandra and K. Chakrabarty, "Test Data Compression for System-on-a-Chip Using Golomb Codes," VLSI Test Symposium, pp. 113-120, April 2000.

[5] S. Kajihara and K. Miyase, "On identifying don't care inputs of test patterns for combinational circuits," Int'l Conf. on Computer Aided Design 2001, pp. 364-369, Nov. 2001.

[6] A. El-Maleh and A. Al-Suwaiyan, "An Efficient Test Relaxation Technique for Combinational & Full-Scan Sequential Circuits," 20th IEEE VLSI Test Symposium, pp. 53-59, April 2002.

[7] T. Sasao, *SWITCHING THEORY FOR LOGIC SYNTHESIS*. Kluwer Academic Publishers, 1999.

[8] R. Bryant, "Graph-based algorithms for Boolean function manipulation," IEEE Trans. on Comp., Vol. 35, No. 8, pp. 677-691, 1986.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms Second Edition*. The MIT Press, 2001.

[10] R. Rudell, "Dynamic variable ordering for ordered binary decision diagrams," Int'l Conf. on Computer Aided Design, pp. 42-47, Nov. 1993.

[11] 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 of Integrated Circuits and Systems, Vol. 14, No. 12, pp.1496-1504, Dec. 1995.