

多重スキャンツリー設計によるテスト圧縮手法

宮瀬 紘平[†] 梶原 誠司[†] 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, [†] kajihara@cse.kyutech.ac.jp, [‡] reddy@engineering.uiowa.edu

あらまし 本論文では、テスト圧縮を目的とした多重スキャン設計手法を提案する。提案手法は、通常のスキャンチェーンの代わりに、与えられたテストパターンに基づいて、スキャンフリップフロップを木状に構成する。テストデータ量とテスト印加時間は、最大の高さのスキャンツリーによって決まるので、そのスキャンツリーの高さを削減するようにスキャンフリップフロップを配置する。さらに、提案手法はテスト圧縮の効果を高めるためのテストベクトル変換を行う。ISCAS-89 ベンチマーク回路に対する実験結果では、提案手法は通常の多重スキャン設計によるテストデータ量を77%削減した。スキャンツリーを用いるとスキャン出力数が増加するが、スキャン出力数を制限した場合でも、通常の多重スキャン設計のテストデータ量を62%削減できた。

キーワード スキャンツリー、多重スキャン設計、テストベクトル変換、ドントケア

Multiple Scan Tree Design for Test Compression

Kohei MIYASE[†] Seiji KAJIHARA[†] and Sudhakar M. REDDY[‡]

[†] Kyushu Institute of Technology 680-4 Kawazu, Iizuka, Fukuoka, 820-8502 Japan

[‡] University of Iowa Iowa City, IA 52242, U.S.A.

E-mail: [†] miyase@aries30.cse.kyutech.ac.jp, [†] kajihara@cse.kyutech.ac.jp, [‡] reddy@engineering.uiowa.edu

Abstract In this paper, we propose a method of test compression for multiple scan designs. Instead of the conventional serial scan chains, the proposed method constructs scan trees in which scan flip-flops are placed and routed in a tree structure. Inputs of the scan trees drive several scan trees of different lengths (height). Since test data volume and test application time are dominated by the scan tree with the maximum height among the constructed scan trees, the proposed method distributes the scan flip-flops to the scan trees so as to minimize the maximum height of the scan trees. In addition, the proposed method modifies the given test vectors to maximize the reduction in test data volume and test application time. Experimental results for ISCAS-89 benchmark circuits show that the proposed method could reduce, on the average, test data volume by 77% compared with the conventional multiple scan design. The scan tree construction enlarges the number of scan outputs required. However test data volume could be reduced by 62% even if the number of scan outputs is limited.

Keyword scan tree, multiple scan design, test vector modification, don't-care

1. はじめに

近年、LSI のテストデータ量・テスト印加時間の増加は深刻な問題になっている。今日まで、この問題を解決する手法は数多く提案されてきた[1-12][15-18]。その中でも多重スキャン設計は実用的な解決策である。多重スキャン設計は、テスト印加時間を削減する一方、テストデータ量は削減しない。そこで、[6,7]では、多重スキャン設計のテストデータ量を削減するためのテストデータ符号化法が提案されている。符号化技術を用いる手法のねらいは、テストからチップに転送するピン数の削減にある。符号化されたテストデータは、チップ上の復号化回路によって元のテストデータに変

換される。これらの手法は、データの削減量の増加に伴って、復号化回路のオーバーヘッドが増加する。[8-12]では、単一のスキャン入力で複数のスキャンチェーンを直接駆動し、復号化回路を必要としない手法が提案されている。同じ入力から複数のスキャンチェーンにデータを転送するため、スキャンフリップフロップに設定される論理値には強い制約が生じ、その制約のため、検出可能な故障が検出できなくなる可能性がある。[8]では、多重スキャンモードに加え単一スキャンモードを追加することにより全ての検出可能故障の検出を保障している。その他のアプローチとしては、スキャンフリップフロップを木状に配置する“スキャ

ンツリーと呼ばれる手法が提案されている[12]。木構造の根はスキャン入力に対応し、葉はスキャン出力に対応する。スキャン入力は異なる長さのスキャンチェーンを駆動することになるが、与えられたテスト集合中の全てのテストベクトルをスキャンフリップフロップ上に設定できる。

本論文は、多重スキャン入力に対するスキャンツリーを用いたテストデータ量・テスト印加時間削減手法を提案する。提案手法では、スキャン入力数と同数のスキャンツリーを構成する。それぞれのスキャン入力が各スキャンツリーを駆動するので、テストデータ量とテスト印加時間はスキャンツリーの高さの最大値によって決定する。提案手法は、その高さの最大値が最小となるように各スキャンツリーにスキャンフリップフロップを割り当てる。また、提案手法の効果を向上させるためにテストベクトル変換を行う。テストベクトル変換は、与えられたテスト集合に対するドントケア判定[14]と、ドントケアへの値の再割り当てに基づく。ドントケアへの値の再割り当ては、インコンパチビリティグラフに対する点彩色問題 [12][18]に帰着する。スキャンツリーは、テストデータ量とテスト印加時間を削減するが、スキャン出力数を増加させるという欠点もある。本論文では、スキャン出力数の制限法も示す。ISCAS-89 ベンチマーク回路に対する実験では、提案手法の有効性を示す。ただし、本論文では、スキャン入力データ量のみの削減を考慮しておりスキャン出力データ量の削減は考慮していない。スキャン出力データ量は、MISR やスペースコンパクターを使用することによって削減することを前提とする。

本論文は以下のように構成される。2章では、提案手法で使用される基礎技術について述べる。3章で多重スキャン設計に対するスキャンツリーの構成法について述べ、4章でベンチマーク回路に対する実験結果を示す。最後に5章でまとめを行う。

2. 準備

2.1 スキャンツリー

図1にスキャンフリップフロップ(以下単にフリップフロップと呼ぶ)を直列に連結した単一スキャンチェーンを示す。単一スキャンチェーンのテストデータ量は、フリップフロップ数とテストベクトル数の積となる。図1の場合、フリップフロップ数が9でテストベクトル数が4なのでテストデータ量は36(=9*4)ビットになる。図1の例では、どのテストベクトルに対しても同じ論理値をとる複数のフリップフロップが存在する。例えば、フリップフロップ ff_3, ff_4, ff_6 は論理値1101をとる。そのようなフリップフロップをコンパチブルフリップフロップと呼ぶ[16]。図1では、 $\{ff_3, ff_4,$

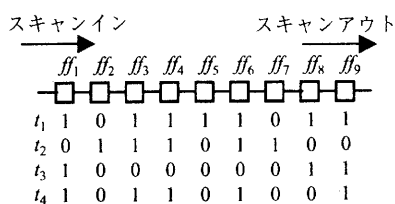


図1: 単一スキャンチェーン

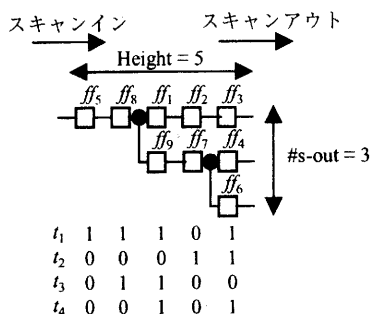


図2: スキャンツリー

$ff_6\}$, $\{ff_2, ff_7\}$, $\{ff_1, ff_9\}$, $\{ff_5\}$, $\{ff_8\}$ の5つのコンパチブルフリップフロップのグループが存在する。コンパチブルフリップフロップがとる論理値は同じスキャン入力から同時に印加できるので、図2のように木状にフリップフロップを配置することができる。これをスキャンツリーと呼ぶ。その木構造の根はスキャン入力に対応し、葉はスキャン出力に対応している。スキャンツリーの基本概念は、フルスキャン回路に対するテストデータ量・テスト印加時間の削減手法の一つである[9][10][13]。図2のスキャンツリーは、スキャンチェーン長を9から5に削減し、その結果テストデータ量を20(=5*4)ビットに削減する。同様に一つのテストベクトルに対するテスト印加時間を9クロックサイクルから5クロックサイクルに削減する。

2.2 スキャンツリーの構成法

スキャンツリーの構成には、まず、各フリップフロップがどれか一つのコンパチブルグループに属するようにフリップフロップをグループ分けする。テストベクトル中の論理値が全て0か1に特定されたテスト集合が与えられたとき、コンパチブルフリップフロップのグループは唯一に決定し、スキャンツリーの構成は容易に決まる。一方、テストベクトルが未設定信号値を含んだテスト集合が与えられた場合、構成されるスキャンツリーは複数存在する。例えば、図1の ff_9 の各テストベクトルでの論理値が110xの場合、 ff_9 は ff_1 とコンパチブルだけでなく ff_8 ともコンパチブルになる。ただし、 ff_1 と ff_8 はコンパチブルではない。したがって、図2のスキャンツリーの ff_1 と ff_8 は置き換えてもよい。

複数の異なるスキャンツリーが構成可能な場合、高

さが最小のスカンツリーが選ばれるべきである。[12]の手法は、単一スカン入力に対して最適なスカンツリーを求めるものである。その手法は論理値が0か1に特定されたテストベクトル中のドントケアを判定し、そのドントケアにスカンツリーの高さを削減するような論理値を割り当てる。ドントケアに論理値を割り当てる問題は、ドントケアを含んだテスト集合により構築したインコンパチビリティグラフに対する点彩色問題に帰着する。

3. 多重スカン入力に対するスカンツリー

3.1 テストデータ量と印加時間の評価

ここでは多重スカンツリーのテストデータ量とテスト印加時間の算出方法を述べる。初期テスト集合 T_{set} に対する単一スカン入力のスカンツリーを st とし、スカンツリーの高さを $H(st)$ と表す。スカンツリー st のテストデータ量は次の式で表される：

$$T_{vol}(st, T_{set}) = H(st) * |T_{set}|$$

同様にテスト印加時間は次の式で表される：

$$T_{time}(st, T_{set}) = (H(st)+1) * |T_{set}|$$

複数のスカン入力を仮定すると $H(st)$ を削減することができる。例えば、2つのスカン入力を使用する場合、スカンツリーを2つ構成できる。図2のスカンツリーに対して、一つ目のスカンツリーを $\{ff_3\}$, $\{ff_8\}$, $\{ff_4, ff_6\}$ で構成し、二つ目のスカンツリーを $\{ff_2, ff_7\}$, $\{ff_1, ff_9\}$ により構成すると図3のようになる。

多重スカンツリー $ST = \{st_1, st_2, \dots, st_{|ST|}\}$ を考えたとき、テストデータ量とテスト印加時間は次のように表される：

$$T_{vol}(ST, T_{set}) = \max(H(st_1), H(st_2), \dots, H(st_{|ST|})) * |ST| * |T_{set}|$$

$$T_{time}(ST, T_{set}) = (\max(H(st_1), H(st_2), \dots, H(st_{|ST|})) + 1) * |ST| * |T_{set}|$$

図3の例では、テストデータ量は $24 (= 3 * 2 * 4)$ ビットになり、テスト印加時間は $16 (= (3+1) * 4)$ クロックサイクルになる。

$H(st) > \max(H(st_1), H(st_2), \dots, H(st_{|ST|}))$ の関係からわかるように、多重スカンツリーのテスト印加時間 $T_{time}(ST, T_{set})$ は、単一スカンツリーの $T_{time}(st, T_{set})$ に

比べて小さい。一方、多重スカンツリーのテストデータ量は、単一スカンツリーに比べて増加することがある。 $\max(H(st_1), H(st_2), \dots, H(st_{|ST|}))$ は $H(st) / |ST|$ 以上なので、 $T_{vol}(ST, T_{set})$ は決して $T_{vol}(st, T_{set})$ より小さくならない。実際、図2の単一スカンツリーのテストデータ量 $T_{vol}(st, T_{set})$ は20ビットであるのに対し、図3の多重スカンツリーの $T_{vol}(ST, T_{set})$ は24ビットである。しかし、 $T_{vol}(ST, T_{set})$ は、通常の単一スカンチェーンのテストデータ量の36ビットよりは少なくなっている。ただし、多重スカンツリーのテストデータ量の算出は、スカンツリーの高さにばらつきがある場合には高さの低いスカンツリーで不要となるデータも含んでいる。したがって悲観的な計算方法となっている。

3.2 高さが均一なスカンツリー

初期テスト集合 T_{set} とスカン入力数が与えられると仮定する。スカン入力数がスカンツリー数 $|ST|$ となるので、 $T_{vol}(ST, T_{set})$ と $T_{time}(ST, T_{set})$ は $\max(H(st_1), H(st_2), \dots, H(st_{|ST|}))$ によって決まる。また、 n をコンパチブルフリップフロップのグループ数とする。 $\max(H(st_1), H(st_2), \dots, H(st_{|ST|}))$ の下界は $\lceil n / |ST| \rceil$ で与えられるので、 n を最小化することが $T_{vol}(ST, T_{set})$ と $T_{time}(ST, T_{set})$ の削減に重要である。その他には、スカンツリーの高さを均一にすることも重要である。すなわち、どのスカンツリー st_i と st_j に対しても $|H(st_i) - H(st_j)| \leq 1$ とすべきである。 $H(st_i)$ が $H(st_j)$ より大きいとき、ダミーデータが必要となりテストのメモリを浪費する[19]。フリップフロップが15個、 $|ST|=3$ の場合の例を図4と図5に示す。図4のスカンツリーは、コンパチブルフリップフロップのグループ数が7で最大の高さが st_1 の4である。各テストベクトルのテストデータ量は12ビットで、そのうち5ビットは st_2 と st_3 に対するダミーデータである。図5のスカンツリーの場合、グループ数が7で最大の高さが st_3 の3である。各テストベクトルのテストデータ量は9ビットで、そのうち2ビットは st_1 と st_2 に対するダミーデータである。このように、グループ数とスカンツリーの高さを小さくすることでテストデータ量とテスト

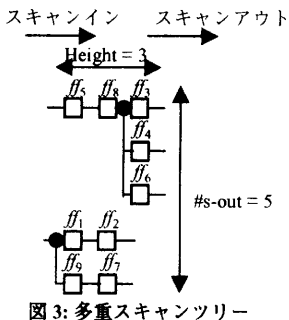


図3: 多重スカンツリー

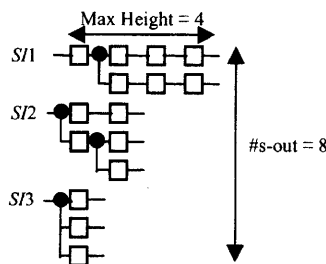


図4: 均一でないスカンツリー

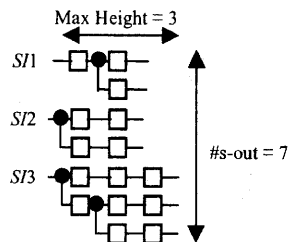


図5: 均一なスカンツリー

印加時間を削減できる。

3.3 スキャン出力数の制限

スキャンツリー構成により、スキャン出力数は増加する。出力数が制限されなければ、スキャンツリーはテストデータ量とテスト印加時間の削減の効果を最大化できるが、出力データを圧縮する MISR のオーバーヘッドは大きくなる。そこで本論文では、スキャンツリーの出力数を制限することを考える。MISR のオーバーヘッドを削減すると、テストデータ量とテスト印加時間は増加するので、その関係はトレードオフである。

スキャンツリーの出力数は、最大のコンパチブルフリップフロップのグループの大きさによって決まる。よって各グループのフリップフロップ数を制限することで出力数を制限する。その結果スキャンツリーの高さは増加する。ここでは、出力数を予め決めておき、その後スキャンツリーを求める。グループ内のフリップフロップ数が制限された出力数を越えた場合、超過したフリップフロップを新しいグループとしてスキャンツリーに加える。図6に例を示す。出力数を2に制限した場合、フリップフロップの ff_9 と ff_{10} を取り除き、スキャンツリーの葉の後ろに加える。

3.4 処理手順

以下では、全ての入力値が0か1に特定されたテスト集合 T_{set} 、スキャン入力数 nsi 、スキャン出力数の上限 nso が与えられたときの、スキャンツリーを構築する処理手順を示す。

- (1) 与えられたテスト集合中にできるだけ多くのドントケアを判定する。
- (2) コンパチブルフリップフロップのグループ数が最小となるように、フリップフロップをグループ分けする。
- (3) uso 以上のフリップフロップを含むグループ G を、 $|G_i| \leq uso$ となるように、最小数のサブグループ G_i

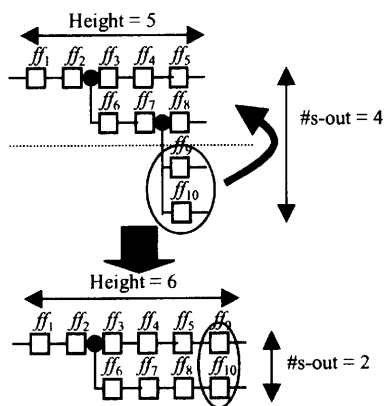


図6: スキャン出力数の制限

に分割する。 ngr を、分割後に得られたグループ数とする。

- (4) ngr 個のグループをフリップフロップ数により昇順に並び替える。
- (5) $\lceil ngr/nsi \rceil$ 個のグループ毎に、スキャンツリーを構築する。

なお、ステップ2は、ドントケアを含んだテスト集合によりインコンパチビリティグラフを構成し、そのグラフに対する点彩色問題を解くことにより行う[12]。

4. 実験結果

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、ISCAS'89 のベンチマーク回路に対して実験を行った。本実験で用いたテスト集合は、テスト圧縮技術を含む[2]の ATPG によって生成されたものである。本章では、提案手法と従来の多重スキャンチェーンによる手法を比較する。

まず表1に出力数を制限しない場合の提案手法による結果を示す。表中の最初の4つの欄はそれぞれ、回路名、テストベクトル数、外部入力数、フリップフロップ数を表す。“ $\#s-in$ ”の欄は、スキャンツリー(スキャン入力)の数を表す。ここでは、スキャンツリーの数が8, 16, 32 の場合について実験を行った。次の“ $m scl$ ”の欄は、与えられた入力数に対して多重スキャンチェーンを使用した場合の最大スキャンチェーン長を表している。“ $max H$ ”の欄では、提案手法によって求めた多重スキャンツリーの高さの最大値を表す。“ T_{vol} ”の欄はテストデータ量をビット数で表しており、“ $conv.$ ”の欄は多重スキャンチェーンに対するビット数、“ $mstree$ ”の欄は多重スキャンツリーに対するビット数である。多重スキャンチェーンに対するビット数は、テストベクトル数と最大スキャンチェーン長とスキャン入力数の積である。多重スキャンツリーに対するビット数は、3章で述べた式で求める。次の“ $Ratio$ ”の欄は、以下の式で計算した削減率である：

$$Ratio = (|T_{org}| - |T_{comp}|) / |T_{org}|$$

ここで、 $|T_{org}|$ は多重スキャンチェーンに対するテストデータのビット数、 $|T_{comp}|$ は多重スキャンツリーを用いた場合のビット数である。提案手法は、スキャン入力に印加するデータのみを削減するが、“ $Total T_{vol}$ ”の欄は外部入力とスキャン入力のテストデータ量の総和、つまりテストに保存すべきテストデータ量を表しており、次の式で求められる：

$$Total T_{vol} = (\#PI + (\#s-in * maxH)) * \#tv$$

ここで、 $\#PI$ は外部入力数、 $\#s-in$ はスキャンツリー数(スキャン入力数)、 $maxH$ は多重スキャンツリーの高さの最大値、 $\#tv$ はテストベクトル数をそれぞれ表す。“ $\#s-out$ ”の欄は多重スキャンツリーの出力数の合計を

表1:出力数を制限しない場合の実験結果

	#rv	#PI	#ff	#s-in	mscl	max H	T_{vol}		Ratio	Total T_{vol}	#s-out	time
							conv.	mstree				
s13207	235	31	669	8	84	13	157920	24440	0.85	31725	82	19.85
				16	42	7	157920	26320	0.83	33605	126	19.85
				32	21	4	157920	30080	0.81	37365	213	19.85
s15850	97	14	597	8	75	25	58200	19400	0.67	20758	35	30.12
				16	38	13	58976	20176	0.66	21534	57	30.12
				32	19	7	58976	21728	0.63	23086	102	30.12
s35932	12	35	1728	8	216	3	20736	288	0.99	708	618	32.44
				16	108	2	20736	384	0.98	804	1037	32.44
				32	54	1	20736	384	0.98	804	1728	32.44
s38417	87	28	1636	8	205	69	142680	48024	0.66	50460	67	100.38
				16	103	35	143376	48720	0.66	51156	86	100.38
				32	52	18	144768	50112	0.65	52548	128	100.38
s38584	114	12	1452	8	182	45	165984	41040	0.75	42408	39	113.42
				16	91	23	165984	41952	0.75	43320	70	113.42
				32	46	12	167808	43776	0.74	45144	133	113.42
Average								0.77				

表し、最後の欄はCPU時間を秒で表している。ただし、CPU時間にテスト生成の時間は含まれていない。

提案手法は、多重スキャンチェーンを用いた手法と比較して、平均約77%のテストデータ量を削減できた。特にs35932の回路に対する実験では、98%のテストデータ量を削減した。表1より、テストデータ量が削減されるほど、多重スキャンツリーの出力数が増加していることがわかる。テスト印加時間も多重スキャンツリーの高さに依存しているため、各テストベクトルを印加するのに必要な時間を77%削減する。

表2では、提案手法と他の圧縮手法をテストデータのビット数で比較した。比較を公正に行うために、提案手法と[8][7]の手法のスキャン入力数は16とした。また、表中のテストデータ量は、外部入力とスキャン入力に対するテストデータ量の合計である。3つの回路に対して提案手法は、最小のテストデータ量を得ることができた。また、[6][7]の手法がテストデータ量削減のために復号化回路を必要とするのに対して、提案手法が復号化回路を必要としない点は提案手法の利点でもある。

次に、多重スキャンツリー出力の合計を16, 32, 64, 128, 256に制限して実験を行った。図3に実験結果を示す。多重スキャンツリーの高さの最大値は、出力数を制限しない場合より高くなる。それでも、提案手法は平均約62%のテストデータ量を削減することができた。

表2:従来手法との比較

	FTCS'99[8]	DAC'01[6]	VTS'02[7]	Proposed
s13207	28808	25344	56635	33605
s15850	29328	22784	23474	21534
s35932	2288	7218	10788	804
s38417	54336	89856	65163	51156
s38584	33136	38796	63612	43320

5. まとめ

本論文では、多重スキャンツリーを用いたテストデータ量とテスト印加時間の削減手法を提案した。テストデータ量とテスト印加時間は多重スキャンツリーの高さに依存するので、その高さの最大値が最小になるようにフリップフロップを各スキャンツリーに分配し、さらに、削減効果を向上させるため、テストベクトル変換を行った。ISCAS-89のベンチマーク回路に対する実験結果では、通常多重スキャンチェーンを用いる手法に比べ平均77%のテストデータ量とテスト印加時間を削減できることを示した。スキャンツリーの出力数を制限した場合でもテストデータ量を平均62%削減できることを示した。

文 献

- [1] I. Hamzaoglu and J. H. Patel, "Test Set Compaction Algorithms for Combinational Circuits," Int'l Test Conf, pp. 283-289, Oct. 1998
- [2] 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. CAD, pp.1496-1504, Dec. 1995.
- [3] B. Koenemann, et. al., "A Smart BIST Variant Guaranteed Encoding," 10th Asian Test Symposium, pp. 325-330, Nov. 2001.
- [4] C. Barnhart, V. Brunkhorst, F. Distler, O. Fransworth, B. Keller and B. Koenemann, "OPMISR: The Foundation for Compressed ATPG Vectors," Int'l Test Conf., pp. 784-757, 2001.
- [5] J. Rajski, J. Tyszer, M. Kassab, N. Mukherjee, R. Thompson, H. Tsai, A. Hertwig, N. Tamarapalli, G. Murgalski, G. Eide, and J. Qian, "Embedded deterministic test for low cost manufacturing test,"

表3:出力数を制限した場合の実験結果

	#tv	#s-in	#s-out	mscl	limited max H	T_{vol}		Ratio	Total T_{vol}	time	
						conv.	mstree				
s35932	12	8	16	216	109	20736	10464	0.50	10884	32.39	
			32		55	20736	5280	0.75	5700	32.38	
			64		28	20736	2688	0.87	3108	32.48	
			16	32	108	55	20736	10560	0.49	10980	32.39
				64		28	20736	5376	0.74	5796	32.38
				128		14	20736	2688	0.87	3108	32.48
			32	64	54	28	20736	10752	0.48	11172	32.39
				128		14	20736	5376	0.74	5796	32.38
				256		7	20736	2688	0.87	3108	32.48
s38417	87	8	16	205	117	142680	81432	0.43	83868	80.52	
			32		84	142680	58464	0.59	60900	80.50	
			64		75	142680	52200	0.63	54636	100.31	
			16	32	103	59	143376	82128	0.43	84564	80.52
				64		42	143376	58464	0.59	60900	80.50
				128		38	143376	52896	0.63	55332	100.31
			32	64	52	30	144768	83520	0.42	85956	80.52
				128		21	144768	58464	0.60	60900	80.50
				256		19	144768	52896	0.63	55332	100.31
s38584	114	8	16	182	101	165984	92112	0.45	93480	113.60	
			32		62	165984	56544	0.66	57912	46.31	
			64		48	165984	43776	0.74	45144	113.39	
			16	32	91	51	165984	93024	0.44	94392	113.60
				64		31	165984	56544	0.66	57912	46.31
				128		24	165984	43776	0.74	45144	113.39
			32	64	46	26	167808	94848	0.43	96216	113.60
				128		16	167808	58368	0.65	59736	46.31
				256		12	167808	43776	0.74	45144	113.39
Average							0.62				

Proc. ITC, pp. 301-310, October 2002.

- [6] I. Bayraktaroglu and A. Orailoglu, "Test Volume and Application Time Reduction through Scan Chain Concealment," Design Automation Conference, pp.151-155, June 2001.
- [7] S. M. Reddy, K. Miyase, S. Kajihara and I. Pomeranz, "On Test Data Volume Reduction for Multiple Scan Chain Designs," 20th IEEE VLSI Test Symposium, pp. 103-108, April 2002.
- [8] I. Hamzaoglu and J. H. Patel, "Reducing Test Application Time for Full Scan Embedded Cores," Int'l Symposium on Fault-Tolerant Computing, pp. 260-267, July 1999.
- [9] S. -C. Chang, K. -J. Lee, Z. -Z. Wu and W. -B. Jone, "Reducing test application time by scan flip-flops sharing," IEE Proc. -Comput. Digit. Tech, Vol. 147, No.1, Jan 2000.
- [10] S. Sybille, H. -G. Liang and H. -J. Wunderlich, "A Mixed Mode Bist Scheme Based on Reseeding of Folding Counters," Int'l Test Conf., pp. 778-784, 2000.
- [11] K. -J. Lee, J. -J. Chen, C. -H. Huang, "Using a single input to support multiple scan chains," ICCAD-98, pp.74-78, Nov. 1998.
- [12] K. Miyase and S. Kajihara, "Oprimal Scan Tree Construction with Test Vector Modification for Test Compression", ATS 2003, Nov. 2003 (to appear).
- [13] J. C. Rau, W. B. Jone, S. C. Chang and Y. L. Wu, "Tree-structured LFSR synthesis scheme for

pseudo-exhaustive testing of VLSI circuits," IEE Proc. -Comput. Digit. Tech, Vol. 147, No.5, Sep 2000.

- [14] S. Kajihara, K. Miyase, "On Identifying Don't Care Inputs of Test Patterns for Combinational Circuits," ICCAD-2001, pp. 364-369, Nov. 2001.
- [15] A. R. Pandey and J. H. Patel, "Reconfiguration Technique for Reducing Test Time and Test Data Volume in Illinois Scan Architecture Based Designs," 20th IEEE VLSI Test Symposium, pp. 9-15, April 2002.
- [16] C. -A. Chen, S. K. Gupta, "Efficient BIST TPG Design and Test Compaction via Input Reduction," IEEE Trans. on CAD, Vol. 17, No. 8, August 1998, pp.692-705.
- [17] P. Goel and B. C. Rosales, "Test Generation and Dynamic Compaction of Tests," Digest of Papers 1979 Test Conf., pp. 189-192, Oct. 1979.
- [18] K. Miyase, S. Kajihara and S. M. Reddy, "A Method of Static Test Compaction Based on Don't Care Identification," IEEE International Workshop on Electronic Design, Test & Applications, pp. 392-395, Jan. 2002.
- [19] A. L. Crouch, *DESIGN for Test*. Prentice Hall PTR, 1999.