多重スキャンツリー設計によるテストデータ量・テスト印加時間の削減

宮瀬紘平[†] 梶原誠司[††] レディ スダーカ[†††]

本論文では，多重スキャン設計に対してテストデータ量・テスト印加時間を削減する多重スキャンツリーを設計し，多重スキャンツリーに対するテスト集合を求めるまでの手順を提案する．まず，スキャンツリーのテストデータ量・テスト印加時間の評価法を示し，スキャンツリーの評価を容易化する．次に，単一スキャン入力に対するスキャンツリー設計から複数入力に対する多重スキャンツリー設計への拡張を行う．さらにスキャン出力数を制限した多重スキャンツリーを設計し，設計した多重スキャンツリーに対するテスト集合を求めるまでの手順を示す．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・テスト印加時間をそれぞれ 76%削減することを示す．スキャンツリーの構築はスキャンアウトに必要な出力数を増加させるが，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・テスト印加時間をそれぞれ 62%削減することができることを示す．

Reduction of Test Data Volume and Test Application Time with Multiple Scan Tree Design

Kohei Miyase,[†] Seiji Kajihara[††] and Sudharkar M. Reddy[†††]

In this paper, we propose a method to design a multiple scan tree so as to reduce test data volume and test application time for multiple scan designs and a procedure to obtain a test set for the multiple scan tree. We first describe how to evaluate test data volume and test application time for scan tree designs in order to facilitate their evaluation. Then we extend a scan tree for single scan input to the one for multiple scan inputs. Next we design a multiple scan tree with the limitation of the number of scan outputs, and show a procedure to obtain a test set for the multiple scan tree. Experimental results for ISCAS-89 benchmark circuits show the proposed method reduces 76% of test data volume and test application time compared to conventional multiple scan design. The scan tree construction enlarges the number of scan outputs required. However, even if we limit the number of scan outputs, the method still reduces 62% of test data volume and test application time.

1. はじめに

VLSI または SoC のテストにおいて，テストデータ量・テスト印加時間の増加は深刻な問題である．特にスキャン設計を行った回路に対するテストでは，スキャンチェーンの長さに比例してテスト印加時間が増加するため，テストデータ量・テスト印加時間の削減が必要不可欠である．これまでにテストデータ量・テスト印加時間を削減する手法は，数多く提案されている[1)〜15),20)〜22)]．

スキャン設計に対するテスト印加時間削減の一般的な手法は，多重スキャン設計である．多重スキャン設計は，スキャンチェーンを複数に分割し，各スキャンチェーンを同時にスキャンシフトさせてテスト印加時間を削減する．しかし，多重スキャン設計は，テストデータ量を削減しない．そこで，文献 5)〜15) では，多重スキャン設計のテストデータ量を削減する手法が提案されている．多重スキャン設計に対するテストデータ量の削減手法は，符号化技術に基づく手法とスキャンフリップフロップの配置に関するものに分類される．

文献 5)〜10) の手法は，符号化技術などでテストデータを圧縮し，テスタに格納するデータ量を削減す

† 科学技術振興機構
Japan Science and Technology Agency
†† 九州工業大学
Kyushu Institute of Technology
††† アイオワ大学
University of Iowa

る．実際にテストを行うときは，圧縮データをチップ上の復号化回路によって元のテストデータに戻す．つまり，チップ上のハードウェアが元のテストデータの一部になっている．このような手法は，テストデータを，テスタが格納する圧縮データと，チップ上の復号化回路に分割するため，TRP（Test Resource Partitioning）と呼ばれている．SoC のテストでは，テストデータ量・テスト印加時間と同様にテストに必要な入出力数を削減することも重要である．文献 5)〜10) の手法は，テスタからチップに転送するピン数の削減することでテストデータ量の削減を実現している．このような TRP では，テストデータ量の削減率が大きい場合に，復号化回路によるオーバヘッドが大きくなることがある．

文献 11)〜15) では，スキャンフリップフロップの配置配線方法を工夫し，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，文献 5)〜10) で使用される復号化回路を必要としない手法が提案されている．文献 11)，14)，20) で提案されている手法はブロードキャストスキャン（Broadcast Scan）と呼ばれ，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．ブロードキャストスキャンは，複数のスキャンフリップフロップに同じ論理値を同時に設定するので，設定される論理値には強い制約が生じ，本来は検出可能な故障が検出できなくなる可能性がある．文献 11) では，ブロードキャスト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すべての検出可能故障の検出を保証している．ブロードキャストスキャンをより一般化したアプローチとして，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[12),13),15),18)]” と呼ばれるテストデータ量・テスト印加時間削減手法が提案されている．文献 16)，17) では，テスト時の消費電力削減手法としてスキャンツリーが紹介されている．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り，ブロードキャストスキャンと同様に設定される論理値には強い制約が生じる．文献 15) では，与えられたテスト集合を基にテストベクトル変換[19)] を用いて単一スキャン入力に対するスキャンツリーを構築する手法が提案されている．しかし，文献 15) では，スキャン入力が 1 つの場合にしか対応しておらず，多重スキャン設計のような複数のスキャン入力が許される回路は対象としていない．通常，スキャン入力数・スキャン出力数はスキャン設計を行う前に設定されているが，大規模化回路において，スキャン入力が 1 つである可能性は非常に低い．

また，スキャンツリーは，スキャン出力数を制限しない場合，テストデータ量・テスト印加時間の削減効果を最大化できるが，出力数が増加すると MISR（Multiple Input Signature Register）などによるオーバヘッドが大きくなってしまう．そのため，スキャンツリーを設計する場合，あらかじめスキャン出力数の上限値を決めたうえで，テストデータ量・テスト印加時間の削減効果を最大化することが求められる．

本論文では，複数のスキャンイン入力を持つ回路におけるテストデータ量・テスト印加時間削減を目的とした，多重スキャンツリー構築法を提案する．まず，多重スキャンツリーに対するテストデータ量とテスト印加時間について考察し，スキャンツリーの評価法を示す．次に，各スキャンツリーへのスキャンフリップフロップの割当て方法を提案する．単一スキャン入力のスキャンツリーの場合と異なり，多重スキャンツリーでは，テストデータ量とテスト印加時間がスキャンツリーの高さの最大値に大きな影響を受ける．そのため，多重スキャンツリーの高さの最大値が最小となるようにスキャンツリーへのフリップフロップ割当てをする．スキャンツリーを構築するときフリップフロップをグループ化するが，その手法は文献 15) の手法を用いる．文献 15) では，スキャンツリーの構築法のみに注目しているが，本論文ではさらに，多重スキャンツリーに対するテスト集合が求まるまでのデータフローについても述べる．なお，提案手法は，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 や空間圧縮回路を使用することによって削減することを前提とする．ISCAS-89 ベンチマーク回路に対する実験では，提案手法がスキャン出力数を制限しない場合に，テストデータ量とテスト印加時間をそれぞれ約 76%削減でき，出力数を制限した場合においても約 62%削減できることを示す．

本論文は以下のように構成される．2 章では，提案手法の基礎技術であるスキャンツリーとテストベクトル中のドントケア判定について述べる．3 章では，単一スキャンツリーと多重スキャンツリーのテストデータ量とテスト印加時間の評価方法，および多重スキャンツリーの構成法について述べる．4 章でベンチマーク回路に対する実験結果により多重スキャンツリーのテストデータ量・テスト印加時間の削減効果を示す．また，従来手法との比較により提案手法の有効性を示す．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．たとえば，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両立フリップフロップと呼ぶ[21]．図 1 では，$\{ff_3, ff_4, ff_6\}$，$\{ff_2, ff_7\}$，$\{ff_1, ff_9\}$，$\{ff_5\}$，$\{ff_8\}$ の 5 つの両立フリップフロップのグループが存在する．両立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と対応し，葉はスキャン出力と対応している．スキャンツリーの基本概念は，文献 12)，13)，16)〜18) で紹介されている．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 1 つのテストベクトルに対するテスト印加に必要なクロック数を 9 から 5 に削減する．

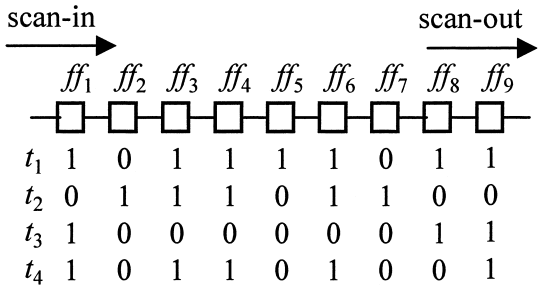

図 1 単一スキャンチェーン
Fig. 1 Single scan chain.

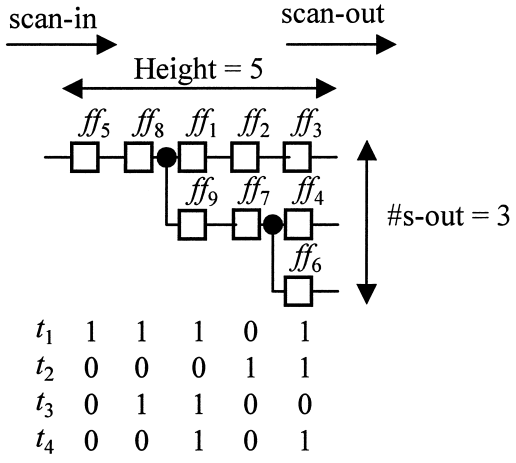

図 2 スキャンツリー
Fig. 2 Scan tree.

スキャンツリーは与えられたテスト集合から構成する．そのため，スキャンツリーの構築に使用しなかったテストベクトルを回路に印加する際，すべてのフリップフロップに任意の論理値を設定できない場合がある．スキャンツリーを構築した後に，すべてのフリップフロップに任意の論理値を設定するには，文献 11)，13) で提案されている単一スキャンモードを追加すればよい．**図 3** に単一スキャンモードを可能にしたスキャンツリーを示す．

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 1 つの両立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すべて 0 か 1 に特定されたテスト集合が与えられたとき，両立フリップフロップのグループは唯 1 つに決定し，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，構成されうるスキャンツリーは複数存在する．たとえば，図 1 の ff_9 の各テストベクトルでの論理値が 101x の場合，ff_9 は ff_1 と両立なだけでなく ff_8 とも両立になる．ただし，ff_1 と ff_8 は両立ではない．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，高さが最小のスキャンツリーが選ばれるべきである．文献 15) では，単一スキャン入力に対するスキャンツ

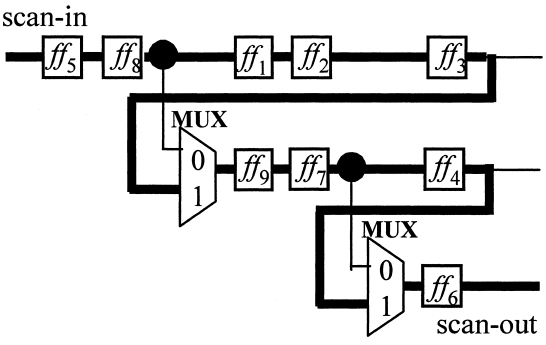

図 3 拡張スキャンツリー
Fig. 3 Extended scan tree.

リーのテストデータ量・テスト印加時間の削減効果を最大化する手法が提案されている．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キャ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非両立グラフに対する点彩色問題に帰着する．

2.3 テストベクトル中のドントケア

いったん生成されたテスト集合中には，通常ドントケアは存在しない．ある故障に対してテスト生成を行った直後のテストベクトルにはドントケアが存在するが，一般にそのようなドントケアには動的/静的圧縮[22] などの処理により 0 か 1 どちらかの論理値が設定される．しかし，すべての論理値が 0 か 1 に設定されたテストベクトル中には，逆の論理値を割り当てても故障検出率が低下しない入力値が存在する．そのような入力値はドントケアと見なすことができる．ただし，いったん生成されてしまったテストベクトル中のどのビットがドントケアであるかは不明である．ドントケア判定手法[19] は，いったん生成されたテストベクトル中のドントケアを，故障検出率を低下させることなく判定する．また，ドントケア判定手法は，故障シミュレーションと ATPG アルゴリズムの含意操作と正当化操作に基づいた処理を用いる．ドントケア判定手法で用いる含意操作と正当化操作は探索をともなわないため，高速にドントケアを判定することができる．

近年，テストベクトル中のドントケアが論理回路のテストにおいて重要な役割を果たしている．ドントケアには任意の論理値を割り当てることができるため，テストベクトルに新しい特性を持たせることが可能である．特に符号化技術などを用いたテストデータ量削減手法[3)~8),10),11),13),15),20)] では，テストベクトル中のドントケアを利用することが必要不可欠である．

3. 多重スキャン入力に対するスキャンツリー

3.1 多重スキャンツリー構築の概略

図 4 に多重スキャンツリー構築のためのスキャンフリップフロップの接続情報と多重スキャンツリーに対するテスト集合を求めるデータフローを示す．まず，単一縮退故障を対象として生成されたテスト集合に対してドントケアを判定する．多重スキャンツリー構築処理の入力データは 2 つある．1 つ目はドントケアを含むテスト集合であり，2 つ目はスキャン設計においてのスキャン入出力数の情報である．文献 15) では，単一スキャン入力と複数スキャン出力を考慮しているが，近年の大規模回路に対してのスキャン設計ではスキャン入力数が単数であることはありえない．本論文では，スキャン入力数・出力数ともに複数の場合に対応可能である．多重スキャンツリー構築処理の出力データは，多重スキャンツリーに対するテスト集合と多重スキャンツリーを構築するためのスキャンフリップフロップの接続情報である．

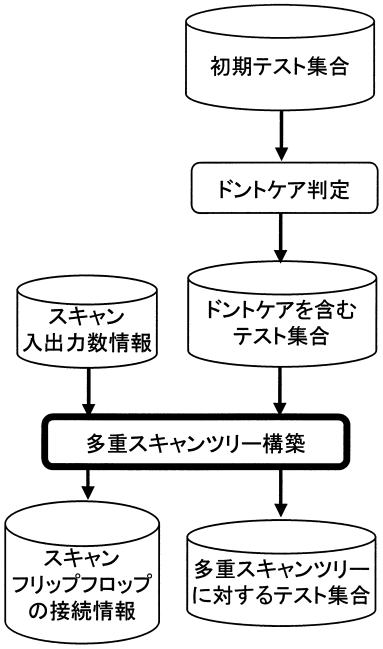

図 4 多重スキャンツリー構築のデータフロー
Fig. 4 Flow of multiple scan tree construction.

以下では，スキャンツリーに対するテストデータ量とテスト印加時間について考察し，スキャンツリーの評価法を示す．多重スキャンツリーは複数のスキャンツリーが構築されるため，各スキャンツリーの高さが異なる可能性がある．単一入力に対するスキャンツリー構築ではツリーの高さを考慮する必要はないが，提案手法では複数のスキャンツリーが構築されるため本章で対処法を述べる．次にスキャン出力数の制限について述べる．最後に，多重スキャンツリー構築の処理手順を述べる．

3.2 テストデータ量とテスト印加時間の評価

ここでは，多重スキャンツリーのテストデータ量とテスト印加時間，多重スキャンツリーによるテストデータ量の削減率，および，外部入力に対するデータ量を含めたデータ量の算出方法について述べる．ただし，テスト印加時間は，テスト印加に必要なクロック数で評価する．初期テスト集合 T_{set} に対する単一ス

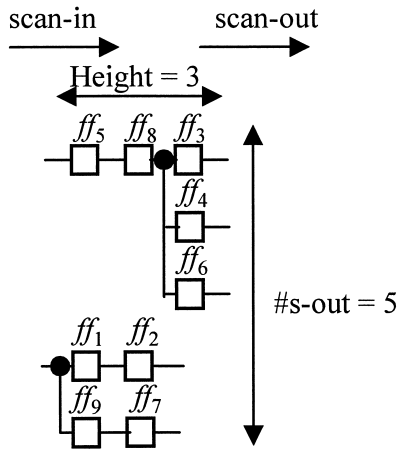

図 5 多重スキャンツリー
Fig. 5 Multiple scan tree.

キャン入力のスキャンツリーを st とし，スキャンツリーの高さを $H(st)$ と表す．スキャンツリー st のテストデータ量は次の式で表される：

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

ここで，上記の式はスキャン入力へ印加するデータ量のみを表しており，スキャン出力のデータ量は含まれていない．テスト印加に必要なクロック数は次の式で表される：

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

複数のスキャン入力を仮定すると $H(st)$ を削減することができる．たとえば，2 つのスキャン入力を使用する場合，スキャンツリーを 2 つ構成できる．図 2 のスキャンツリーに対して，1 つ目のスキャンツリーを $\{ff_5\}$，$\{ff_8\}$，$\{ff_3, ff_4, ff_6\}$ で構成し，2 つ目のスキャンツリーを $\{ff_2, ff_7\}$，$\{ff_1, ff_9\}$ により構成すると**図 5** のようになる．

多重スキャンツリー $ST = \{st_1, st_2, \ldots, st_{|ST|}\}$ を考えたとき，テストデータ量とテスト印加に必要なクロック数は次のように表される：

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

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

図 5 の例では，テストデータ量は 24（=3*2*4）ビットになり，テスト印加に必要なクロック数は 16（=(3+1)*4）になる．

$H(st) > max(H(st_1), H(st_2), \ldots, H(st_{|ST|}))$ の関係から分かるように，多重スキャンツリーのテスト印加に必要なクロック数 $T_{time}(ST, T_{set})$ は，単一スキャンツリーの $T_{time}(st, T_{set})$ に比べて小さい．一方，多重スキャンツリーのテストデータ量は，単一スキャンツリーに比べて増加することがある．多重スキャンツリーのテストデータ量の算出は，スキャ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キャンツリーで不要となるダミーデータも含んでいるからである．$max(H(st_1), H(st_2), \ldots, H(st_{|ST|}))$ 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 の単一スキャ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 5 の多重スキャンツリーの $T_{vol}(ST, T_{set})$ は 24 ビットである．しかし，$T_{vol}(ST, T_{set})$ は，通常の単一スキャ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．

本論文では，従来の一般的な多重スキャンチェーンに対するテストデータ量と多重スキャンツリーに対するテストデータ量を比較することにより提案手法の効果を示す．多重スキャンツリーによるテストデータ量の削減率 “*Ratio*” を，以下の式で表す：

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

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，$|T_{mstree}|$ 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，外部入力とスキャン入力のテストデータ量の総和，つまりテスタに保存すべきテストデータ量 “*Total* T_{vol}” は次の式で求められる：

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

ここで，$\#PI$ は外部入力数，$\#s\text{-}in$ 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．

3.3 高さが均一なスキャンツリー

初期テスト集合 T_{set} とスキャン入力数が与えられると仮定する．スキャン入力数がスキャンツリー数 $|ST|$ となるので，$T_{vol}(ST, T_{set})$ と $T_{time}(ST, T_{set})$ は $max(H(st_1), H(st_2), \ldots, H(st_{|ST|}))$ によって決まる．また，n を両立フリップフロップのグループ数とする．$max(H(st_1), H(st_2), \ldots, H(st_{|ST|}))$ 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．そのほかには，スキャ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，どのスキャンツリー 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タのメモリを浪費する[23]．フリップフロップが 15 個，$|ST| = 3$ の場合の例を**図 6** と**図 7** に示す．図 6 のスキャンツリーは，両立フリッ

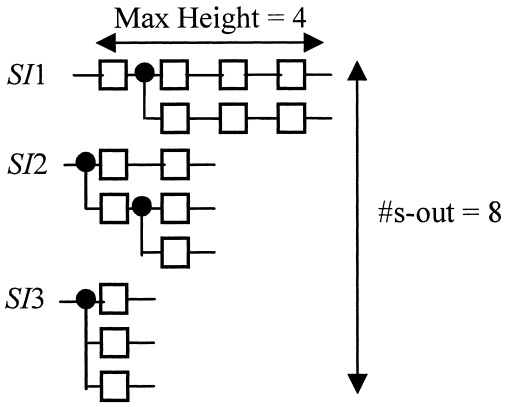

図 6 均一でないスキャンツリー
Fig. 6 Unbalanced scan tree.

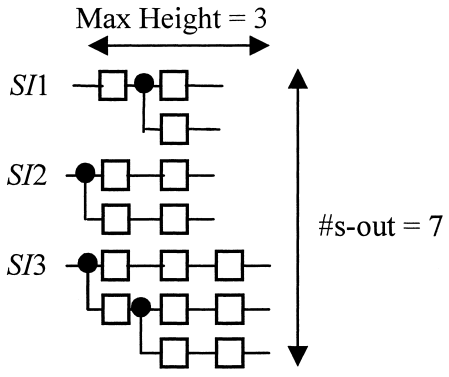

図 7 均一なスキャンツリー
Fig. 7 Balanced scan tree.

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 7 のスキャンツリーの場合，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，スキャンツリーの高さは均一にすべきである．

3.4 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 の面積は大きくなる．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となる．

スキャンツリーの出力数は，最大の両立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できる．ただし，スキャンツリーの高さは増加する．ここでは，出力数をあらかじめ決めておき，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超えた場合，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 8** 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．

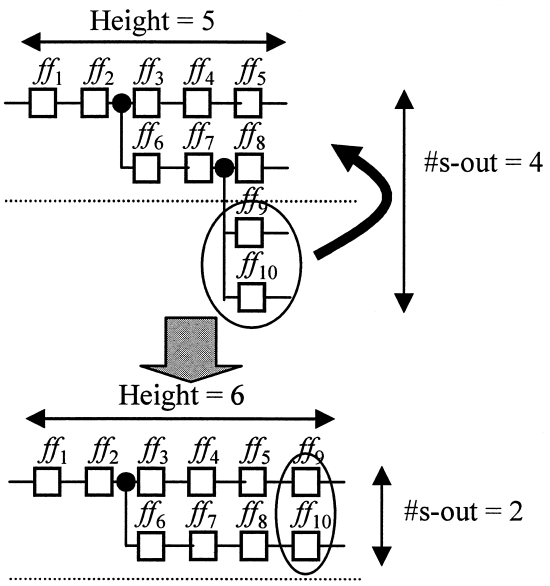

図 8 スキャン出力数の制限
Fig. 8 Limiting the number of scan outputs.

3.5 処 理 手 順

以下では，すべての入力値が 0 か 1 に特定されたテスト集合 T_{set}，スキャン入力数 nsi，各スキャンツリーのスキャン出力数の上限 nso が与えられたときの，多重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) 両立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) nso 以上のフリップフロップを含むグループ G を，$|G_i| \leq nso$ となるように，最小数のサブグループ G_i 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べ替える．
(5) $\lceil ngr/nsi \rceil$ 個のグループごとに，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非両立グラフを構成し，そのグラフに対する点彩色問題を解くことにより行う[15]．また，ステップ 2 のグループ分けの際に，テストベクトル中のドントケアには同じグループ内のフリップフロップがとる論理値 0 または 1 が割り当てられ，この時点で多重スキャンツリーに対するテスト集合が求まる．

以上の処理により，多重スキャンツリー全体のスキャン出力数を nso^*nsi に制限したテストデータ量・テスト印加時間を削減する多重スキャンツリーが構築され，多重スキャンツリーに対するテスト集合が求まる．

4. 実験結果

提案手法を Dual Athlon MP 2000+，512 MB 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，テストベクトル数削減技術を含む文献 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，テスト印加時間削減のために従来から使用されている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 8，16，32，64，128 の場合について実験を行った．次の “*mscl*” 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*”，“*Total* T_{vol}” の欄は，3 章で述べた式で得られる削減率と外部入力とスキャン入力のテストデータ量の総和である．“*#s-out*” の欄は多重スキャンツリーの出力数の合計を表し，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 76%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，93%のテストデータ量を削減した．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ので，各テストベクトルを印加するのに必要な時間を平均約 76%削減することができた．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

表 1 出力数を制限しない場合の実験結果
Table 1 Experimental results without the limitation of scan outputs.

	#tv	*#PI*	*#ff*	*#s-in*	*mscl*	*max H*	T_{vol} *conv.*	T_{vol} *mstree*	*Ratio*	*Total* T_{vol}	*#s-out*	*time*
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
				64	11	2	165440	30080	0.82	37365	371	19.85
				128	6	1	180480	30080	0.83	37365	669	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
				64	10	4	62080	24832	0.60	26190	192	30.12
				128	5	2	62080	24832	0.60	26190	333	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
				64	27	1	20736	768	0.96	1188	1728	32.44
				128	14	1	21504	1536	0.93	1956	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
				64	26	9	144768	50112	0.65	52548	217	100.38
				128	13	5	144768	55680	0.62	58116	383	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
				64	23	6	167808	43776	0.74	45144	257	113.42
				128	12	3	175104	43776	0.75	45144	504	113.42
Average									0.76			

表 2 従来手法との比較
Table 2 Comparison with previous works.

	FTCS'99[11]	DAC'01[9]	VTS'02[10]	TCOM'03[4]	ITC'04[5]	Proposed
s13207	28808	**25344**	56635	30880	83160	33605
s15850	29328	22784	23474	26000	52264	**21534**
s35932	2288	7218	10788	N/A	N/A	**804**
s38417	54336	89856	65163	93466	825552	**51156**
s38584	**33136**	38796	63612	77812	172224	43320

が分かる．また，スキャンツリー数を増やし続けると，多重スキャンツリーの高さの最大値は 1 になり，すべてのフリップフロップが並列に配置されることになる．多重スキャンツリーの高さの最大値が 1 になる場合，多重スキャンツリーの出力数は，フリップフロップ数と同数になり多すぎる．出力数が多いと 3.3 節で述べたように MISR などのオーバヘッドが大きくなるため出力数を考慮してスキャンツリーの数を決める必要がある．また，3.4 節で述べたように出力数を制限して多重スキャンツリーを構築することも重要になる．CPU 時間は，スキャン入力の本数には依存しない．対象回路によって CPU 時間のばらつきはあるが，いずれも実用的な時間内で多重スキャンツリーの構築が可能である．

表 2 では，提案手法と他のいくつかの異なる圧縮手法をテストデータのビット数で比較し，提案手法の有効性を示す．表 2 の 11) は，ブロードスキャンを用いており，5), 9), 10) の手法は多重スキャンチェーンに対して符号化技術を用いた手法である．4) は，多重スキャンチェーンとは関係なく符号化技術を用いた手法である．ただし，提案手法と 5), 10), 11) 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，4), 5), 9), 10) 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ある．表 2 の欄 “N/A” は文献にデータが示されていないことを示す．

表 3 に提案手法と文献 5) の手法で削減されたテストデータ量の比較を，スキャン入力数が 16，32，64，128 の場合について行った．文献 5) の手法は，スキャン入力数が増加するほど，テストデータ量を削減できる．しかし，入力数が増加すると復号化回路が複雑になるという欠点がある．また，スキャンチェーン数が少ない場合は，テストデータ量の削減効果が小さい．多重スキャンツリーは，スキャン入力数に依存することなく一定のテストデータ量の削減が可能である．

表 3 手法 5) との比較
Table 3 Comparison with the method of 5).

	#s-in	ITC'04[5]	Proposed
s13207	16	83160	33605
	32	41580	37365
	64	27729	37365
	128	15140	37365
s15850	16	52264	21534
	32	37529	23086
	64	24168	26190
	128	12220	26190
s38417	16	825552	51156
	32	550377	52548
	64	275196	52548
	128	172144	58116
s38584	16	172224	43320
	32	86116	45144
	64	57417	45144
	128	29964	45144

s38417 の回路に対する実験では，スキャン入力を 128 に設定した場合でも，多重スキャンツリーによるテストデータ量の削減量は大きい．したがって対象回路によってはスキャン入力数を増やしても多重スキャンツリーのほうが有効であるといえる．

次に，多重スキャンツリー全体での出力の合計を 16，32，64，128，256 に制限して実験を行った．各スキャンツリーの出力の制限は，出力の合計（$\#s\text{-}out$）をスキャンツリー数（$\#s\text{-}in$）で割った商で表される．**表 4** 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，提案手法は従来の多重スキャンチェーンによる手法に比べてテストデータ量を平均約 62%削減することができ，提案手法の有効性を示している．CPU 時間は，表 1 と同様に実用的な時間内で多重スキャンツリーを構築できることを示す．

5. おわりに

本論文では，多重スキャンツリー設計によるテストデータ量・テスト印加時間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ため，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た．さらに，スキャン出力数を制限して多重スキャンツリーを設計し，設計した多重スキャンツリーに対するテス

表 4 出力数を制限した場合の実験結果
Table 4 Experimental results of the proposed method.

	#tv	#s-in	#s-out	mscl	limited max H	T_{vol} conv.	T_{vol} mstree	Ratio	Total T_{vol}	time
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		

ト集合を求めるまでの処理手順を示した．ISCAS-89のベンチマーク回路に対する実験結果では，通常の多重スキャンチェーンを用いる手法に比べテストデータ量とテスト印加時間を平均 76%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．

謝辞 本研究の一部は，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合衆国との共同研究による．

参 考 文 献

1) Hamzaoglu, I. and Patel, J.H.: *Test Set Compaction Algorithms for Combinational Circuits*, pp.283–289, ITC (1998).
2) Kajihara, S., Pomeranz, I., Kinoshita, K. and Reddy, S.M.: Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits, *IEEE Trans. CAD*, Vol.14, No.12, pp.1496–1504 (1995).
3) Miyase, K., Kajihara, S. and Reddy, S.M.: A Method of Static Test Compaction Based on Don't Care Identification, *IPSJ Journal*, Vol.43, No.5, pp.1290–1293 (2002).
4) Chandra, A. and Chakrabarty, K.: Test Data Compression and Test Resource Partitioning for System-on-a-Chip Using Frequency-Directed Run-Length (FDR) Codes, *IEEE Trans. Comput.*, Vol.52, No.8, pp.1076–1088 (2003).
5) Wurtenberger, A., Tautermann, C.S. and Hellebrand, S.: *DATA COMPRESSION FOR MULTIPLE SCAN CHAINS USING DICTIONARIES WITH CORRECTIONS*, pp.926–935, ITC (2004).
6) Koenemann, B., et al.: A Smart BIST Variant Guaranteed Encoding, *10th Asian Test Symposium*, pp.325–330 (2001).
7) Barnhart, C., Brunkhorst, V., Distler, F., Fransworth, O., Keller, B. and Koenemann, B.: *OPMISR: The Foundation for Compressed ATPG Vectors*, pp.784–757, ITC (2001).
8) Rajski, J., Tyszer, J., Kassab, M., Mukherjee, N., Thompson, R., Tsai, H., Hertwig, A., Tamarapalli, N., Murgalski, G., Eide, G. and Qian, J.: *Emebedded deterministic test for low cost manufacturing test*, pp.301–310, ITC (2002).
9) Bayraktaroglu, I. and Orailoglu, A.: *Test Volume and Application Time Reduction through Scan Chain Concealment*, pp.151–155, DAC (2001).
10) Reddy, S.M., Miyase, K., Kajihara, S. and Pomeranz, I.: On Test Data Volume Reduction for Multiple Scan Chain Designs, *20th IEEE*

VLSI Test Symposium, pp.103–108 (2002).
11) Hamzaoglu, I. and Patel, J.H.: Reducing Test Application Time for Full Scan Embedded Cores, *Int'l Symposium on Fault-Tolerant Computing*, pp.260–267 (1999).
12) Chang, S.-C., Lee, K.-J., Wu, Z.-Z. and Jone, W.-B.: Reducing test application time by scan flip-flops sharing, *IEE Proc. Comput. Digit. Tech.*, Vol.147, No.1 (2000).
13) Hellebrand, S., Liang, H.-G. and Wunderlich, H.-J.: *A Mixed Mode Bist Scheme Based on Reseeding of Folding Counters*, pp.778–784, ITC (2000).
14) Lee, K.-J., Chen, J.-J. and Huang, C.-H.: Using a single input to support multiple scan chains, *ICCAD*, pp.74–78 (1998).
15) Miyase, K. and Kajihara, S.: Scan Tree Design: Test Compression with Test Vector Modification, *IPSJ Journal*, Vol.45, No.5, pp.1270–1278 (2004).
16) Xiang, D., Gu, S., Sun, J.-G. and Wu, Y.-L.: *A Cost-Effective Scan Architecture for Scan Testing with Non-Scan Test Power and Test Application Cost*, pp.744–747, DAC (2003).
17) Bhattacharya, B.B., Seth, S.C. and Zhang, S.: *Double-Tree Scan: A Novel Low-Power Scan-Path Architecture*, pp.470–479, ITC (2003).
18) Rau, J.C., Jone, W.B., Chang, S.C. and Wu, Y.L.: Tree-structured LFSR synthesis scheme for pseudo-exhaustive testing of VLSI circuits, *IEE Proc. Comput. Digit. Tech.*, Vol.147, No.5 (2000).
19) Miyase, K. and Kajihara, S.: XID: Don't Care Identification of Test Patterns for Combinational Circuits, *TCAD*, Vol.23, No.2, pp.321–326 (2004).
20) Pandey, A.R. and Patel, J.H.: Reconfiguration Technique for Reducing Test Time and Test Data Volume in Illinois Scan Architecture Based Designs, *20th IEEE VLSI Test Symposium*, pp.9–15 (2002).
21) Chen, C.-A. and Gupta, S.K.: Efficient BIST TPG Design and Test Compaction via Input Reduction, *IEEE Trans. CAD*, Vol.17, No.8, pp.692–705 (1998).
22) Goel, P. and Rosales, B.C.: Test Generation and Dynamic Compaction of Tests, Digest of Papers 1979 Test Conf., pp.189–192 (1979).
23) Crouch, A.L.: *DESIGN-FOR-TEST*, Prentice Hall PTR (1999).

(平成 17 年 10 月 25 日受付)
(平成 18 年 4 月 4 日採録)

宮瀬 紘平（正会員）

2000 年九州工業大学情報工学部電子情報工学科卒業，2002 年九州工業大学大学院情報工学研究科博士前期課程修了，2005 年同博士後期課程修了．博士（情報工学）．2005 年より科学技術振興機構研究成果活用プラザ福岡研究員．VLSI のテスト容易化設計，故障診断等の研究に従事．2004 年 IEEE 福岡支部学生研究奨励賞受賞．電子情報通信学会，IEEE 各会員．

梶原 誠司（正会員）

1987 年広島大学総合科学部総合科学科卒業，1992 年大阪大学大学院工学研究科博士後期課程修了．博士（工学）．同大学工学部応用物理学科助手を経て，1996 年九州工業大学情報工学部電子情報工学科助教授．2003 年より，同大学教授．この間，1997〜1999 年大阪大学大学院工学研究科助教授（併任）．VLSI のテスト生成，テスト容易化設計等の研究に従事．1996 年電子情報通信学会学術奨励賞，2002 年情報処理学会山下記念研究賞，2005 年電子情報通信学会論文賞受賞．電子情報通信学会，IEEE 各会員．

レディ スダーカ

オスマニア大学（インド）電気通信工学卒業．アイオワ大学電気コンピュータ工学科博士課程修了．工学博士．1968 年アイオワ大学電気コンピュータ工学科助教授，1971 年アイオワ大学電気コンピュータ工学科準教授，1977 年アイオワ大学電気コンピュータ工学科教授，1981 年よりアイオワ大学電気コンピュータ工学科長．論理回路のテスト容易化設計，テスト生成等の研究に従事．IEEE Transaction on Computer 等の多くの学術雑誌の編集委員を歴任，1990 University of Iowa Foundation Distinguished Professor，1995 Von Humboldt Prize for a Senior U.S. Scientist 受賞．IEEE フェロー．