ＬＳＩ検査高速化のための
テストアーキテクチャに関する研究

平成18年度

三重大学大学院工学研究科
博士前期課程 電気電子工学専攻

京谷忠雄

修士論文

LSI検査高速化のための テストアーキテクチャに関する研究

平成18年度修了
三重大学大学院工学研究科
博士前期課程　電気電子工学専攻

京谷　忠雄

目次

図一覧

表一覧

第1章

はじめに

1.1　研究の背景

近年，情報産業の大幅な成長や応用分野への拡大により，高集積回路 (LSI) は一般社会へ広く浸透している．それに伴い，社会システム全体がLSIに依存する度合が増してきており，LSIの故障によるシステムの停止や誤動作は単にシステムの利用者に一時的な不便を与えるだけにとどまらず，商工業活動停止による金銭的損害，人身へ与える被害など深刻な影響を及ぼす場合も多い．

技術的観点から見た場合，LSIの製造には微細加工技術が用いられており，LSI内部の配線は密集しているため，小さな埃が付着するだけでも故障が混入してしまう．そのため，LSIの高信頼化のための研究は非常に重要なものとなっている．各メーカーでは，高い品質と信頼性を確保するために，LSI製造時の最終工程において故障のあるLSIを特定する電気的検査（LSIテスト）を行っている．一般にLSIテストは，製造されたLSIから数個だけ取り出して行う標本検査ではなく，すべてのLSIを検査する全品検査を行っている．これは，LSIが非常に故障が混入しやすい製品であるため，すべてのLSIに対して検査する必要があるからである．

半導体製造技術は微細加工技術とシリコンウェハの大口径化により日々成長を続けている．この恩恵を受ける形でLSI製造にかかるコストは年々下がる傾向にある．一方，LSIテストにおけるコストは上昇しており，LSIの製造コストに占めるテストコストの割合が高くなってきている．これは，近年のLSIの大規模化に伴い，従来のLSIテスト法におけるテスト実行時間が大幅に増加しているためである．また，製品コストを考える上で，テストに割くことのできるコストは限られており，なるべく低コストなテスト法が望まれている．

1.2　LSIテストと単一縮退故障モデル

LSIテストとは，故障が混入したLSIを識別するための工程のことである．LSIは，仕様をもとに設計者がLSIの設計図（設計データ）を作成し，その設計図に従い工場で製造される．このLSIの製造工程中に，故障が混入して正常に動作しないLSIが製造されることがある．このようなLSIを識別するために，製造時最終工程においてLSIテストが行われる．LSIテストを行う場合，製造不良による欠陥LSIを識別するための，回路の設計図をもとに生成された検査用データを用いる．LSIテストの結果，正常と判断されたLSIのみが出荷されることになる（図1.1）．

現在のLSIテストでは，0と1からなる論理値データをLSIに入力し，設計図から計算された出力応答と実際に得られた出力応答を比較し，その正誤によってLSIの故障を判別する論理テストが最も一般的に行われている．このようなLSI内の故障を検出する入力データ（0，1の系列）をテストベクトルと呼ぶ．

LSIの故障モデルはいくつか提案されているが，最も一般的に用いられている故障モデルは，単一縮退故障モデルである．これは，論理値が0や1に固定化される縮退故障（図1.2）がLSI内部にひとつだけあり，同時にふたつの縮退故障は存在しないという故障モデルである．この単一縮退故障の検出率が100%であればLSIテストの信頼度も充分に信頼できるものであると経験的に判断されている．また故障の中には，検出が容易なものと非常に困難なものが存在する．検出が困難な故障として，テストベクトル集合を入力しても一度しか検出できない故障を必須故障と呼ぶ．

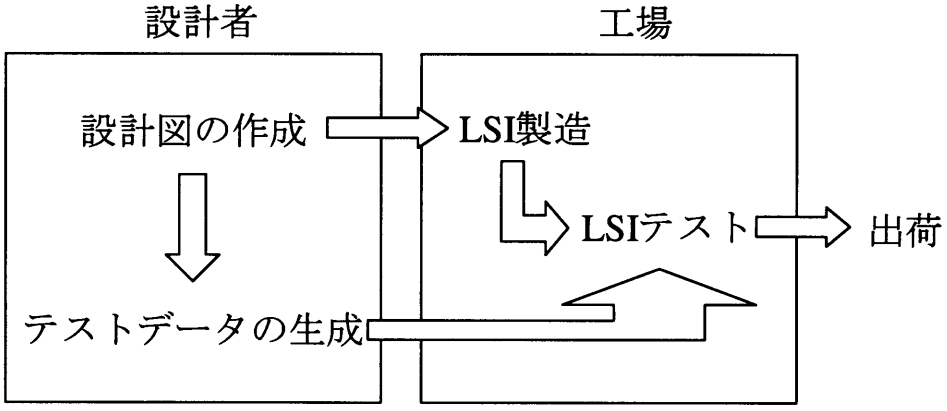

図 1.1: LSIの設計から出荷までの流れ

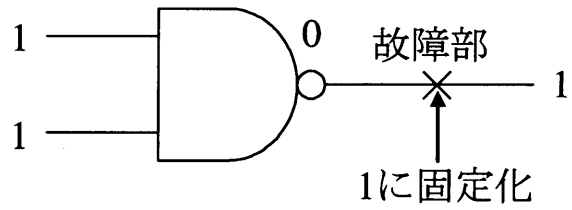

図 1.2: 縮退故障モデル

1.3　フルスキャンテスト

通常LSIは，組合せ回路部とフリップフロップ（FF）部から構成される順序回路である．LSIテストでは，動作中に起こる様々な論理値の組合せをテストする必要がある．しかし，前状態の論理値によって現在の論理値が変化する順序回路では，任意の論理値でテストを行うことは困難である．そこで，順序回路のテストを組合せ回路のテストとして行うフルスキャンテスト法が広く用いられている．フルスキャンテストのアーキテクチャを図1.3に示す．フルスキャンテストのアーキテクチャは図1.4のようなマルチプレクサ付のFFを数珠状につなぎシフトレジスタ（または，スキャンチェーンと呼ぶ）を構成し，通常の入力のほかにテストデータ入力を設けている．LSIテスト時には，S/L（シフト/ロード）ピンから制御信号を与え，マルチプレクサをテストデータ入力側Sに切り替えることで，FFの論理値を外部から任意の論理値に設定することができる．このようにして，フルスキャンテスト法では，LSIテスト時に組合せ回路としてテストすることが可能となる．

フルスキャンテストのテスト手順について説明する．組合せ回路用のテストベクトルをScan inピンから1ビットずつ入力し，ひとつのテストベクトルがすべてスキャンチェーンに格納された後，S/Lピンから制御信号Lを与え，被検査回路である組合せ回路を動作させる．回路動作後，出力応答（テスト応答）が再びスキャンチェーンに格納される．そして，次のテストベクトルを入力すると同時に，テスト応答を外部に出力し，そのテスト応答の正誤によって故障の有無を判別する．

このようにフルスキャンテストでは，テストデータ入力部がシフトレジスタであるため，テストベクトルを1ビットずつしか入力できない．そのため，テスト実行時間はテストベクトルの入力時間がほとんどであり，入力するテストデータ量（スキャンシフト量）とクロックサイクルの積にほぼ等しくなる．

一方，近年のLSIの大規模化に伴い，スキャンチェーン数とテストベクトル数は増大している．そのため，テストデータ量が大幅に増大しており，テスト実行時間の増加が問題となっている．

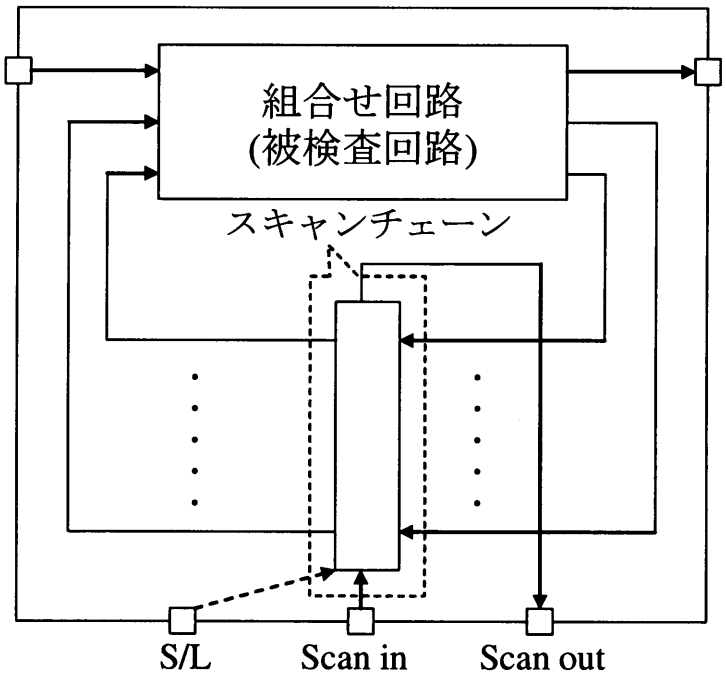

図 1.3: フルスキャンアーキテクチャ

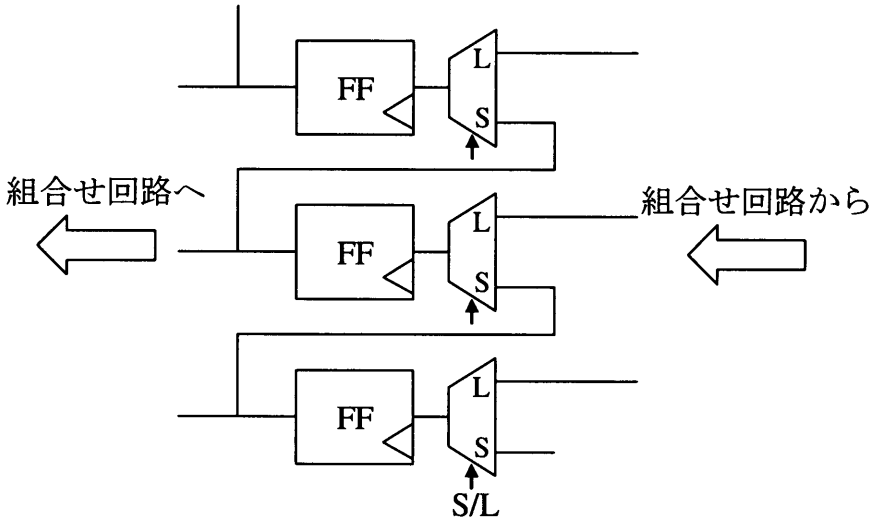

図 1.4: スキャンチェーンの内部構造

1.4　過去の研究

前述で述べたように，LSIの大規模化に伴い，フルスキャンテストのテスト実行時間の増加が問題となっている．そこで近年，フルスキャンテストの高速化を目的とした手法が数多く提案されている[10]．ドントケアビットを含むテストベクトルをエンコードしたデータ圧縮コードを用いLSIに内蔵された展開器により展開する手法[13, 14, 15]，XOR（排他的論理和）ゲートを用いた線形変換により圧縮データからテストベクトルに展開するリニア展開器を用いる手法[11, 12]，疑似乱数発生器を内蔵して外部テスタからのテストデータ供給量を削減する手法[16, 17, 18]などが提案されている．しかしながら，これらの手法はハードウェア追加のためのチップスペースが必要になるため，チップコストが上昇する．

それに対し，フルスキャンアーキテクチャに新規ハードウェアを一切追加しない手法として，テスト応答の一部をテストデータの一部として利用するフルスキャンテストの高速化手法[1, 2]がある．この手法は，データ展開圧縮回路を組み込む手法に比べテスト時間削減率は劣るものの，アーキテクチャが従来のフルスキャンテストと同じであるため，手軽に採用することができるという利点を有するとともに，データ展開圧縮回路を追加しないためチップコストの面でも優れている．この手法は，次のテストベクトルを入力する際，スキャンチェーンに格納されているテスト応答を，次のテストベクトルの一部として利用することにより，スキャンシフト量を削減するものである．しかしながら，ここでの提案手法は，スキャンチェーン中のFFの並べ換えが自由に行えることを前提としている．スキャンチェーンの並び順を変更するには，FF間の結線を変える必要があり，必ずしも並べ換えが常に自由に行えるわけではない[3]．

百万ゲートを越えるような大規模なLSIにおいて，縮退故障を対象としたテストベクトル集合は非常に高いドントケア率であることが報告されている[4, 5]．さらに近年，組合せ回路用ATPGで，縮退故障の故障検出効率を低下させることなく，ドントケア率（テストベクトル集合中の全テストデータに占めるドントケアビットの割合）の高いテストベクトル集合を生成する手法が報告されている[6, 7, 8]．

そこで，文献[9]では，正当化技術とテストベクトル中のドントケアビットを利用し，スキャンチェーン中のFFの並べ換えを前提としない，文献[1, 2]の手法のためのテスト入力系列の生成手法が提案されている．

2章では，新規ハードウェアを一切追加しないフルスキャンテスト高速化手法である文献[1, 2, 9]について述べる．

第2章

スキャンシフト量削減手法

本章では，文献[1, 2]で提案されているフルスキャンアーキテクチャのままでスキャンシフト量を削減する手法（Reduced Scan Shift：以後RSSと記述）と文献[9]で提案されている手法を紹介する．

2.1 Reduced Scan Shift 法[1, 2]

（ｉ）テスト応答を利用したスキャンシフト量削減手法

フルスキャンテストでは，テストベクトルを1ビットずつ入力してテストを実行し，テスト応答を全ビット出力している．次のテストベクトルを入力するときには，スキャンチェーン内部にひとつ前のテスト応答が格納されている．RSSの基幹アイディアは，これを利用するものである．すなわち，テスト応答の後部と次に入力するテストベクトルの前部が，同じビット列である場合，その部分は新しく入力する必要はなく，残りのビット列だけを入力すればよい．

これを模式的に表すと図2.1のようになる．最初のテストベクトルtv1が10111であり，それに対応するテスト応答が01010である．そして次に入力するテストベクトルtv2が01011である．このときtv1のテスト応答の後部010と次に入力するtv2の前部010が同じビット列である．そのため，このビット列（010）を利用して残りの2ビット（11）のみを入力すればtv2と同じベクトルを得ることができる．このとき，外部にスキャンアウトされるテスト応答は，2ビット（01）のみとなり，テスト応答の一部しかスキャンアウトされない．

このように，スキャンシフト量削減のため，スキャンチェーンに格納されているテスト応答の後部を，次に入力するテストベクトルの前部として利用し，残りの部分のみをスキャン入力する手法をテスト応答・テストベクトルオーバラッピング法（Test Response Test Vector Overlapping法：以後，TRTVO法と記述）と呼ぶことにする．

RSSでは，RSSの基幹アイデアである上記TRTVO法の性能向上のため，以下（ii）

（iii）で述べる手法が用いられている．

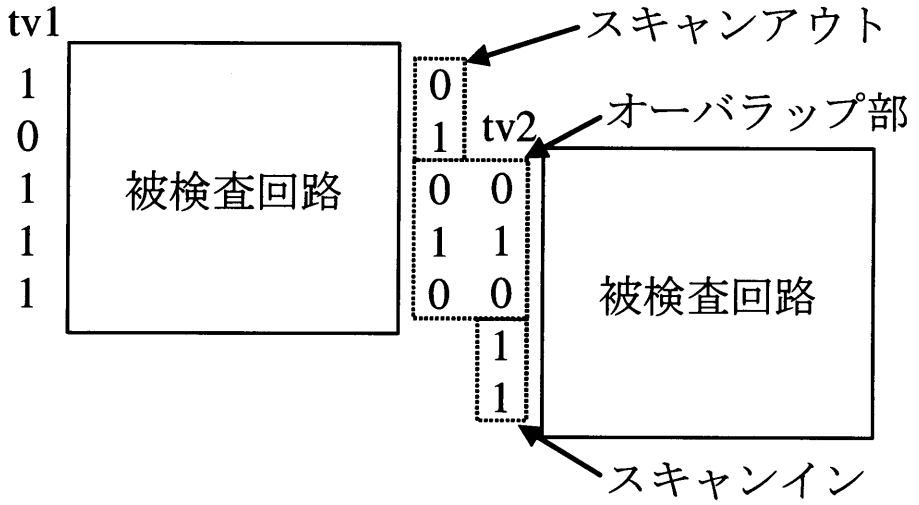

図 2.1: テスト応答・テストベクトルオーバラッピング

（ii）対象故障

TRTVO 法では，テストベクトル中のドントケアビットが多いほど，オーバラップしやすくなり，その結果スキャンシフト量が減る．そこで，多くのドントケアビットを確保するため，RSS では，対象故障を必須故障（故障検出効率100%のテストベクトル集合中で，ただ1つのテストベクトルでのみ検出される故障）に絞って生成したテストベクトル集合を使用している．これは，必須故障を対象として生成されたテストベクトル集合は，ほとんどすべての縮退故障を検出するという性質を利用している．

（iii）スキャンFFの並べ換え

RSS では，スキャンシフト量削減のためにスキャンチェーン中の FF の並べ換えを行っている．その並べ換え手法を図2.2を用いて説明する．図2.2は，3つのテストベクトルとそれぞれのテスト応答を表している．ここで，図中の D は，正常なテスト応答と故障時のテスト応答（故障応答）が異なる FF，すなわち故障を検出する FF を示している．テストベクトルにおいて，ケアビットが頻繁に現れる FF が存在する．また，テスト応答においても必須故障を頻繁に検出する FF が存在する．オーバラップ量を増やし，かつ，必須故障をスキャンアウトしやすくするためには，ケアビットが頻繁に現れる FF がなるべくスキャン入力ピンの近くになり，かつ，必須故障を頻繁に検出する FF をなるべくスキャン出力ピンの近くになるようにすると効果的である．これを実現するため，RSS では，(2.1) 式より各 FF の評価値を定義している．

$$E(FF_i) = VC(FF_i) - VO(FF_i) \tag{2.1}$$

ここで，(2.1) 式において，E は FF_i の評価値，VC は FF_i のケアビットの数，VO は FF_i

の必須故障の検出個数を表しており，評価値の大きなFFがスキャン入力ピン側になるようにFFの並べ換えを行っている．図2.2の例では，スキャン入力ピンからFF2，FF1，FF0（FF3），FF3（FF0）の順番になる．

	Test vectors					Faulty response			
	FF0	FF1	FF2	FF3		FF0	FF1	FF2	FF3
tv1	1	0	1	X	→	D	-	-	D
tv2	X	0	1	1	→	D	D	-	-
tv3	0	1	0	X	→	D	-	-	D
#care bits	2	3	3	1	#faulty responses	3	1	0	2

D: faulty response

図 2.2: テストベクトルとテスト応答

2.2　正当化技術を用いたスキャンシフト量削減手法 [9]

RSSでは，スキャンチェーン中のFFの自由な並べ換えが可能であることを前提としている．しかし，スキャンチェーン中のFFの自由な並べ替えは，FF間の結線を変更する必要があり，レイアウト上の制約から必ずしも並べ換えが常に自由に行えるわけではない [3].

そこで文献 [9] では，FFの並べ換えを前提としないTRTVO法の性能向上のための手法，すなわち，正当化技術とテストベクトル中のドントケアビットを利用しスキャンシフト量を削減する，TRTVO法のためのテスト入力系列の生成手法を，提案している．また，TRTVO法では，テスト応答を一部しかスキャンアウトできないため，検出できない故障（非検出故障）が現れる可能性がある．このような非検出故障に対する2つの異なった救済法を提案し，議論しており，縮退故障検出効率100%を保証している．文献 [9] で提案されている2つの非検出故障の救済法について，以下に述べる．

（ｉ）スキャンシフト量増加法

非検出故障を検出するには，オーバラップ部に隠れている故障応答がスキャンアウトされるようにオーバラップ量を減らし，すなわち，スキャンシフト量を増加することにより，可能である．そこで最も多くの非検出故障を検出するテスト応答に対し，そのテスト応答で検出できるすべての非検出故障をスキャンアウトするようにオーバラップ量を減少させる．以下この処理を非検出故障がなくなるまで繰り返す．

（ii）オーバラップベクトル追加法

オーバラップベクトル追加法は，非検出故障として残った故障集合全体を対象とした組合せ回路用テストベクトル集合を新たに生成し，元のTRTVO法用テスト入力系列の後方にオーバラップ追加することで非検出故障を検出する手法である．以下この処理を非検出故障がなくなるまで繰り返す．

なお，上記2つの救済法では（ii）オーバラップベクトル追加法の方が良好な結果を示しており，文献[9]では，正当化技術とオーバラップベクトル追加法を用いて，スキャンシフト量削減率において，RSSと同等，またはそれ以上の結果を示している．

第I部

線長制約化でのスキャンチェーンFF並べ換え

三重大学大学院　工学研究科

第3章

第I部の概要

文献 [9] では，スキャンチェーン中の FF の並べ換えを一切行わないという前提で，TRTVO 法のためのテスト入力系列生成手法を提案している．もし，FF を都合よく並べ換えれば，文献 [9] の手法により TRTVO 法におけるスキャンシフト量をさらに削減することができるが，FF の自由な並べ換えはレイアウト上の制約から困難である．そこで第I部では，スキャンチェーンの総結線長があまり長くならないような FF の並べ換えであれば許容されることを仮定し，スキャンチェーンの総結線長を制約として与え，その制約の下でスキャンシフト量を最小化することを目的とした効果的なスキャンチェーン中の FF の並び順を決定する手法を提案する．

以下，4 章で提案手法を述べ，5 章で実験結果を示す．6 章では第I部のまとめを行う．

第4章

スキャンチェーンFF並べ換え手法

本章では，スキャンチェーンの総結線長があまり長くならないようなFFの並べ換えであれば許容されることを仮定し，スキャンチェーンの総結線長を制約として与え，その制約の下でスキャンシフト量を最小化することを目的とした効果的なスキャンチェーン中のFFの並び順を決定する手法を提案する．

本手法の具体的な手順は次のようになる．まず，自由なFFの並べ換えを許し，総スキャンシフト量を最小にするためのFFの並び順を，ヒューリスティックな手法により求める（4.1節）．その後，その並び順を基にして，その総スキャンシフト量からの増加量ができるだけ小さくなるような，かつ，スキャンチェーン総結線長が制約として与えられた長さ以下になるような並び順を求める（4.2節）．このように得られたスキャンチェーンに対して，文献[9]で提案されているテスト入力系列生成手法を適用する．

以下，4.1節で総スキャンシフト量を最小にするためのFFの並び順決定法について，4.2節では，4.1節から求めたFFの並び順を基にして，その総スキャンシフト量からの増加量ができるだけ小さくなるような，かつ，スキャンチェーン総結線長が制約として与えられた長さ以下になるような並び順を求める手法について説明する．

4.1　制約なしでのスキャンFFの並べ換え

本節では，自由な並べ換えを許し，総スキャンシフト量を最小とするためのFFの並び順決定法について述べる．TRTVO法は，オーバラップ量が大きくなるほどスキャンイン時間を削減できる反面，テスト応答のスキャンアウト量が減るため，故障検出効率が悪化する．その場合，テストデータの追加等による救済が必要となる．それをできるだけ回避するためには，テスト応答が外部にスキャンアウトされるFFで効率的に故障を検出するように並べ換える必要がある[1, 2]．本提案手法では，各FFに対して，そのFFのドントケアビット数を基に与えた評価値と，そのFFが検出する故障を基に与えた評価値から，総合的な評価値を決定し，並べ換えを行う．RSSでは，対象故障を必須故

障にして，FFに対する評価式を定義している（(2.1) 式の E）のに対し，本提案手法では，すべての縮退故障を対象として評価式を定義する．

（ｉ）ドントケアビット数によるFFの評価値：DC

TRTVO法では，テスト応答の後部と，次に入力するテストベクトルの前部をオーバラップすることによりスキャンシフト量の削減を行っている．そのため，次に入力するテストベクトルの前部に多くのドントケアビットがあれば，オーバラップに有利となる[1, 2]．本手法では，すべての縮退故障を対象として得られたドントケアを含むテスト集合について，各FFのドントケアビットの数を評価値として（4.1）式より与える．

$$DC(FF_i) = FF_i\text{のドントケアビットの数} \tag{4.1}$$

DC の値が大きいFFを，スキャン出力ピン側に配置した方がオーバラップに有利となる．

（ii）検出故障に基づくFFの評価値：FI

検出故障に基づく評価値の定義について図4.1を用いて説明する．図4.1はテスト応答の例を表している．ここで，図中の f_i は，単一故障 f_i が存在したとき，正常なテスト応答と故障時のテスト応答（故障応答）が異なるFF，すなわち故障 f_i を検出するFFを示している．各故障の検出回数は，それぞれの故障により異なる（図4.1で f_0 : 3回，f_1 : 1回，f_2 : 1回，f_3 : 2回）．テスト応答を一部しかスキャンアウトしないTRTVO法では，検出回数の少ない故障の故障応答はスキャンアウトされにくくなる．そのため，検出回数の少ない故障を数多く検出するFFをスキャン出力ピン側に配置すると故障応答出力に有利である．まず，故障毎の総検出回数に基づいて，その故障の検出難易度を（4.2）式より定義する．

$$\text{検出難易度}\,(f_i) \;=\; \frac{1}{(f_i\text{の総検出回数})^n} \tag{4.2}$$

n は正の整数で経験的に定める（実験では $n = 2$）．(4.2) 式では総検出回数が少ない故障ほど検出難易度は高くなる（図4.1で $f_0 : 1/3^2$，$f_1 : 1$，$f_2 : 1$，$f_3 : 1/2^2$）．次に，このようにして求めた故障検出難易度を基に（4.3）式より各FFの評価値 FI を決定する．

$$FI(FF_i) \;=\; \sum FF_i\text{が検出する故障の検出難易度} \tag{4.3}$$

(4.3) 式では，FI の値が大きいFFほど，検出難易度が高い故障を数多く検出していることになる（図4.1でFF0 : 0.36=$1/3^2 + 1/2^2$，FF1 : 1.00，FF2 : 0.36，FF3 : 1.00，FF4 : 0.11）．そのため，FI の値が大きいFFを，スキャン出力ピン側に配置した方が故障応答出力に有利となる．

	Faulty response				
	FF0	FF1	FF2	FF3	FF4
tr1	f_0		f_3		
tr2			f_0	f_1	
tr3	f_3	f_2			f_0

tr_i : test response

f_j : 故障

図 4.1: 故障検出表

上記の方法により求めた各FFのDCとFIの値を基に総合的な評価値を決定する．しかし，DCとFIはそれぞれ全く違う単位であり，単純な評価値の足し合わせには問題がある．そこで，まず（4.4）（4.5）式を用いてDCとFIの値を0〜1に正規化を行う．

$$DC'(FF_i) = \frac{DC(FF_i) - DC_{min}}{DC_{max} - DC_{min}} \tag{4.4}$$

$$FI'(FF_i) = \frac{FI(FF_i) - FI_{min}}{FI_{max} - FI_{min}} \tag{4.5}$$

ここで，DC_{max}，DC_{min}はDCの最大値と最小値，同様にFI_{max}，FI_{min}はFIの最大値と最小値を表している．正規化を行ったDC'，FI'を用いて，各FFの総合的な評価値を（4.6）式により決定する．

$$E(FF_i) = \alpha \times DC'(FF_i) + (1-\alpha) \times FI'(FF_i)$$

ここでEはFF_iの評価値，α，$1-\alpha$はDC'，FI'の重み値であり，αは0〜1の間の値をとる．このようにして求めたEの大きい順に，FFをスキャン出力ピン側からスキャン入力ピンの方に配置する．

4.2 スキャンFFの並び順の変更手法

前節で述べた手法により，総スキャンシフト量は短縮されるが，スキャンチェーンのFF間の総結線長が大幅に増大してしまう．本節では，4.1節で決定した総スキャンシフト量を最小にするためのFFの並び順を基にして，その総スキャンシフト量からの増加量ができるだけ小さくなるような，かつ，スキャンチェーン総結線長が制約として与えられた長さ以下になるような並び順を求める手法について述べる．

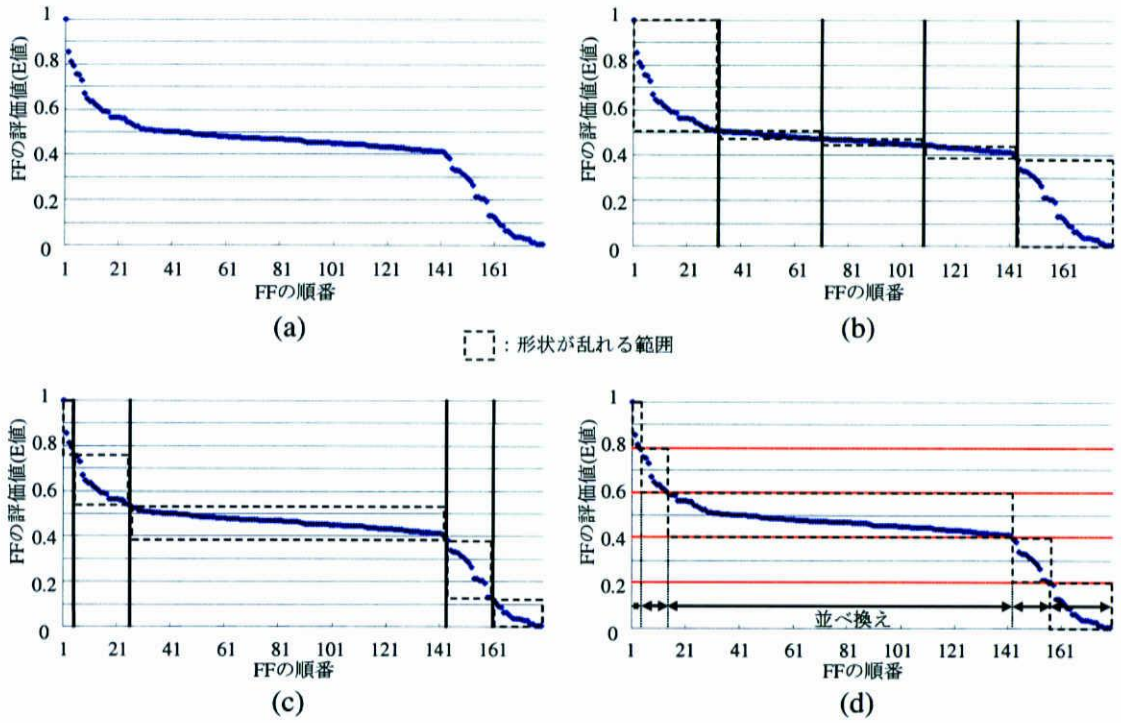

図 4.2: グラフ形状に基づくスキャンチェーン分割法

（ｉ）グラフ形状に基づくスキャンチェーン分割法

前節で評価式（4.6）によりFFの並び順が定まったとき，各FFの評価値の推移が図4.2（a）のようになったとする．図4.2において，評価値の高い左方がスキャン出力ピン側，右方がスキャン入力ピン側に対応している．この回路例について，このグラフの形は，総スキャンシフト量が最小になる（準）最適な形であるといえる．そのため，このグラフの形状をできるだけ乱さずに，スキャンチェーンの総結線長を短くするような並び順に変更すれば，総スキャンシフト量の増加を小さく抑えることができると考えられる．そこで，スキャンチェーンを複数のブロックに分割し（図4.2（b)），各ブロック毎に，ブロック内部で総結線長ができるだけ短くなるようなFFの並び順に変更する手法をとることにする．そのような変更であれば，グラフの形状の乱れは，図4.2（b）内に示す波線長方形の内部に限定される．

図4.2（b）では，各ブロックのFF数が等しくなるようにスキャンチェーンを分割しているが（横軸等分割法と呼ぶ），このように各ブロックのFF数が等しくなるよう分割するのではなく，各ブロックのFF数の間に偏りを持たせて，多数のFFをもつブロックが存在する方が，並べ換えによる全体の総結線長の短縮に有利である．なぜなら，多数のFFを持つブロック内部での自由な並べ換えにより，そのブロックで結線長を大幅に短縮できるからである．これは，極端な例として，100個のFFを2つのブロックに分割する場合，FFを50:50に分割するよりも99:1に分割した方が，総結線長をより短縮できるこ

とを考えれば，直感的にも明らかであろう．

しかし，一般に，多くのFFをもつブロックでは，並べ換えによりグラフの形状が大きく乱れてしまう可能性が高く，スキャンシフト量を大きく増大させてしまうおそれがある．これについては，グラフの傾きの緩やかな箇所が広くなるようにブロック分けすれば，グラフの乱れを少なくすることができる．すなわち，グラフの傾きが急な箇所は狭く，緩やかな箇所は広くなるようにグラフを分割すれば（図4.2（c）），評価値の差の小さな箇所のブロックが広く設定されるため，ブロック内では自由に並べ換えてもグラフの形状の乱れは小さく，そのブロック内での総結線長を大幅に短縮できる．これを実現する具体的な単純な手法として，縦軸を等分割する手法を用いる．例えば図4.2（d）のように，評価値（縦軸）が0〜0.2，0.2〜0.4，…，0.8〜1.0のFFをもつ各ブロックに分割する．このように，縦軸等分割によるブロック分けを行えば，グラフの傾きの緩やかな箇所を，横軸に関して広いブロックに設定することができる．以後，この手法を“縦軸等分割法”と呼ぶ．

（ii）グリーディ法によるFFの並び順変更

グラフの分割が完了したら，分割された各ブロック内で，総結線長ができる限り短くなるようにFFの並び順を変更する．しかし，総結線長が最短となるFFの順番を求める問題は，巡回セールスマン問題と同一であり，最短解を求めることは，一般に困難である．ここでは，最も単純な手法であるグリーディ法を用いて近似的にFFの並び順を決定する．順番の決定は，スキャン出力ピン側である最も評価値の高いブロックから順に行っていく．まず，最も評価値の高いブロック内のFFで，スキャン出力ピンから最も近いFFを選び，一番目のFFとする．その後，選ばれたFFから，同ブロック内で最も近いFFを次のFFとして順次決定していく．ブロック内のFFがすべて選択されたら，次のブロックに移り，前のブロックの一番後ろのFFから最も近いFFを次のFFとする．以上の処理をすべてのブロックが終了するまで繰り返す（図4.3）．

（iii）二分探索法

これまでに本節（i）（ii）で述べた手法は，グラフの分割数（ブロック数）を大きくすると，各ブロック内に含まれるFFが少なくなり並べ換えの自由度が低くなるため，総結線長はあまり短くならない．しかしながら，グラフの形状はほぼ維持されるため，総スキャンシフト量は小さいまま，あまり増加しない．すなわち，グラフの分割数の増加に対して，総結線長は増加傾向，総スキャンシフト量は減少傾向を示す．そこで，与えられた総結線長制約を満足する，できるだけ大きな分割数を効率よく見つけるために，図4.4に示す二分探索法を用いる．

図4.4の#bestsubchainsは，現在までに得られた，総結線長制約を満足する最も大きな最良分割数を記憶しておくもので，初期値は0を入れておく．#scanChainFFsはスキャンチェーン全体を構成する全FFの数で，二分探索法で用いる分割数の下界値と上界値

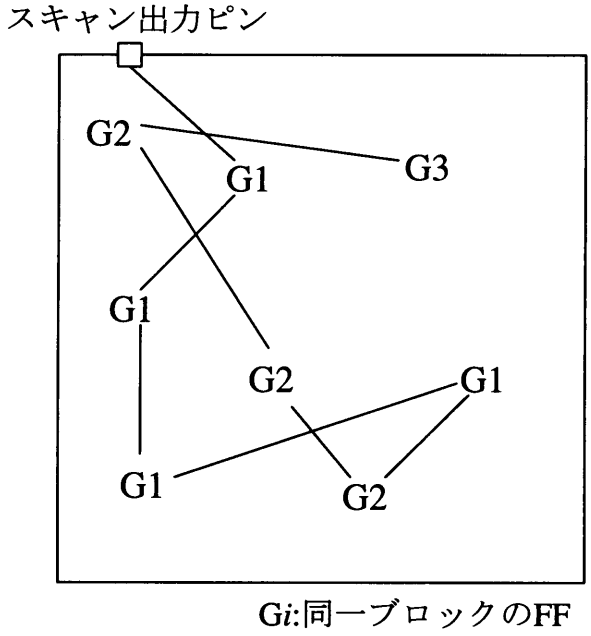

図 4.3: ブロック別グリーディ法

をそれぞれ#subchains_lower（初期値1）と#subchains_upper（初期値#scanChainFFs）に保持し，その中間値#sub_chainsの分割数でチェーン分割・並び順変更を行い，総結線長を計算する．総結線長が制約総線長以下であれば，#bestsubchainsをその分割数の値に更新する．これを二分探索法により，下界値と上界値の差が0になるまで狭めながら，実行していく．アルゴリズムが終了したとき，最良分割数#bestsubchainsが0のままであったときは，総結線長制約を満足する解が一つも得られなかったことを示す．

分割数に対して，総結線長や総スキャンシフト量が厳密に単調増加関数や単調減少関数になるわけではないので，これは，必ずしも，総結線長制約を満足する並び順のうちで，総スキャンシフト量を厳密に最小にするものを常に得ることができるアルゴリズムではないが，そのための近似手法となっている．

本章（4章）で述べた手法で得られたスキャンチェーンに対して，文献[9]のTRTVO法のためのテスト入力系列生成手法をそのまま適用する．

```
#bestsubchains = 0;
#subchains_lower = 1;
#subchains_upper = #scanChainFFs;

while ( #subchains_lower ≦ #subchains_upper ){

  #sub_chains = ( #subchains_lower + #subchains_upper ) / 2;

  分割数#sub_chainsで，グラフ形状に基づくスキャンチェーン分割;
                        /* 3.2(a)節 */
  各ブロック別にグリーディ法によるFFの並び順変更;
                        /* 3.2(b)節 */

  if ( 総結線長 ≦ 制約総結線長 ){
    #bestsubchains = #sub_chains;
    #subchains_lower = #sub_chains + 1;
  }
  else {
    #subchains_upper = #sub_chains − 1;
  }
}
if ( #bestsubchains == 0 ) print ( "NO RESULT" );
```

図 4.4: 二分探索法によるスキャンFFの並び順決定フロー

第5章

結果と考察

上記の手法をUNIX上にC言語で実装し，ISCAS '89ベンチマーク回路に対して実験を行った．使用した計算機は，1.9 GHzのIntel Pentium 4プロセッサと514 MBの主メモリを持つものである．表5.1に実験に用いたテスト集合を示す．#TV，X ratioは，テストベクトル数とテスト集合のdon't care率を示している．また，このテスト集合は文献[8]のシステムにより生成されたものである．

表 5.1: テストベクトル集合

curcuits	#TV	X ratio [%]
s5378	99	74.8
s9234	110	73.6
s13207	233	93.5
s15850	97	82.8
s35932	14	47.5
s38417	100	82.3
s38584	115	83.7

5.1　提案手法の実験結果

表5.2に文献[9]で提案されているFFの並べ換えを行わない手法の結果，および総結線長制約の下で並べ換えを行う本提案手法の結果，また比較のために，自由にスキャンFFの並べ換えを行った場合（4.1節の手法を用いる）の結果を示す．表5.2のLf，Wfはそれぞれ，フルスキャンテストによるテスト実行時間（クロックサイクル），FFの並べ換えを行わない場合のスキャンチェーンの総結線長を示している．また，L，Wは各手法でのテスト実行時間とスキャンチェーンの総結線長を示しており，#SCは二分探索法で得られたブロック数を示している．Tは3章で提案した手法のための計算時間である．

スキャンチェーンの総結線長を評価するためには，各FFのLSI上の位置情報とFF並

表 5.2: 実験結果（制約付 FF 並べ換え）

		FF並べ換えなし(文献[9])			自由なFF並べ換え手法					
circuit	Lf	L	R [%]	Wf	α	L	R [%]	W	W/Wf	T [sec]
s5378	17,999	10,139	43.7	15,164	1.0	5,980	66.8	90,302	5.96	281
s9234	25,418	16,034	36.9	17,238	0.2	10,615	58.2	120,752	7.00	859
s13207	156,779	61,765	60.6	29,064	0.5	28,986	81.5	295,149	10.16	12,113
s15850	58,603	33,615	42.6	28,874	0.2	24,694	57.9	294,480	10.20	6,176
s35932	25,934	22,067	14.9	46,225	0.4	20,433	21.2	1,095,020	23.69	1,014
s38417	165,336	126,464	23.5	45,962	0.5	91,408	44.7	938,897	20.43	31,827
s38584	168,547	103,854	38.4	42,649	0.8	77,834	53.8	895,654	21.00	30,452
Ave.			37.2				54.9			

	線長制約下でのFF並べ換え手法											
	制約 : W/Wf = 1.5						制約 : W/Wf = 2.0					
circuit	#SC	L	R [%]	W	W/Wf	T [sec]	#SC	L	R [%]	W	W/Wf	T [sec]
s5378	5	6,822	62.1	21,353	1.41	307	11	6,380	64.6	28,364	1.87	324
s9234	9	12,814	49.6	23,504	1.36	934	32	10,973	56.8	33,674	1.95	924
s13207	25	31,964	79.6	42,946	1.48	13,285	64	31,860	79.7	57,720	1.99	13,389
s15850	13	26,879	54.1	42,686	1.48	6,403	39	25,152	57.1	56,542	1.96	6,473
s35932	4	22,283	14.1	64,558	1.40	1,151	11	21,790	16.0	91,490	1.98	1,103
s38417	4	103,315	37.5	64,359	1.40	34,590	11	93,992	43.2	86,275	1.88	36,804
s38584	3	87,456	48.1	60,690	1.42	36,358	10	80,669	52.1	82,348	1.93	34,485
Ave.			49.3						52.8			

べ換えなしのスキャンチェーンが必要である．本実験では，すべての FF を配置する物理的位置をランダムに定めた．FF 並べ換えなしのスキャンチェーンはできるだけ短いものになっているものとし，具体的には，スキャン出力ピンに最も近い位置の FF を FF0，FF0 に最も近い位置の FF を FF1，一般に，FF 番号がまだ割り当てられていない位置の FF の中で FF_i から最も近い位置の FF を FF_{i+1} とし，この順に FF がつながっているものとした．FF 間の距離はマンハッタン距離とした．

R はフルスキャンテストに対するテスト時間削減率を示しており，(5.1) 式で定義される．

$$R = \frac{Lf - L}{Lf} \times 100 \quad [\%] \tag{5.1}$$

(4.6) 式の α は，各回路で，それぞれ α=0.0，0.2，0.4，0.5，0.6，0.8，1.0 の 7 通りを行い，最もテスト時間削減率の良い値を採用した．表 5.2 にその α の値を示す．

表 5.2 より，文献 [9] と自由な FF 並べ換え手法を比較すると，テスト時間削減率は平均 37%から 55%に向上しているが，スキャンチェーンの総結線長は大幅に増大しており，最大の 3 つの回路（s35932，s38417，s38584）では約 20 倍となっている．次に，総結線長制約下で FF を並べ換える本提案手法と，自由な FF の並べ換え手法を比較する．総結線長制約として FF 並べ換えなしの元のスキャンチェーンの総結線長の 2 倍を与える（表 5.2 内　制約:W/Wf=2.0）．この制約線長は，最大の 3 つの回路において，自由な FF 並べ換え手法を適用したときに得られる総結線長のほぼ十分の一の長さであるが，テスト時間削減率は平均約 53%であり，自由な FF 並べ換え手法を適用したときとほぼ同等程度の値を示している．総結線長制約として FF の並べ換えを行わないスキャンチェーン長の 1.5 倍を与えたときと，自由な FF 並べ換え手法とのテスト時間削減率の平均値の差は

5%程度である．これらのことから，本提案手法がスキャンチェーンの総結線長制約下でのFFの並べ換え手法として有効であることがわかる．

5.2　縦軸等分割法と横軸等分割法の性能比較

本提案手法の縦軸等分割法の効果を調べるために，グラフの傾きを考えずに各ブロック内のFF数が均等になるように等分割（横軸等分割）した場合との比較を行った．その結果を表5.3に示す．これは二分探索法は使わず，ブロック数を10に固定し，4.2（i）（ii）節の手法を適用した結果である．両手法を比べると，テスト時間削減率はほぼ同等程度であるが，スキャンチェーンの総結線長はすべての回路において，本提案手法である縦軸等分割法の方が良好な結果を得ていることがわかる．

表　5.3: 縦軸等分割法と横軸等分割法の比較

			縦軸等分割法(提案法)				横軸等分割法			
circuit	Lf	Wf	L	R [%]	W	W/Wf	L	R [%]	W	W/Wf
s5378	17,999	15,164	6,358	64.7	28,802	1.90	6,344	64.8	38,874	2.56
s9234	25,418	17,238	11,181	56.0	26,476	1.54	11,263	55.7	41,944	2.43
s13207	156,779	29,064	34,213	78.2	39,440	1.36	35,153	77.6	70,967	2.44
s15850	58,603	28,874	25,536	56.4	43,544	1.51	25,071	57.2	64,594	2.24
s35932	25,934	46,225	20,346	21.5	89,555	1.94	22,752	12.3	117,461	2.54
s38417	165,336	45,962	97,358	41.1	90,349	1.97	91,977	44.4	117,875	2.56
s38584	168,547	42,649	80,669	52.1	82,348	1.93	79,301	53.0	125,545	2.94
Ave.				52.9				52.1		

(#SC=10)

5.3　スキャンFFの評価関数の性能評価

本提案手法では，4.1節で述べた制約なしでFFの自由な並べ換えを行う際，すべての故障を対象にし，各故障の検出難易度を（4.2）式を用いて計算し，それに基づき，最終的に（4.6）式により各FFの評価値を求め，各FFのスキャンチェーン中の位置を定めている．それに対し，各故障の検出難易度を，RSSで行っているように必須故障，すなわち，一回しか検出されない故障のみで評価した場合の実験を行った．(4.2) 式の検出難易度を，一回検出故障は1，複数回検出故障は0と定義する．表5.4に制約なしでFFの自由な並べ換えを行った実験結果を示す．提案法の方が，わずかではあるが良い結果を示している．

表 5.4: 異なる評価関数による実験結果

回路	Lf	提案法		一回検出法	
		L	R [%]	L	R [%]
s5378	17,999	5,980	66.8	6,059	66.3
s9234	25,418	10,615	58.2	11,488	54.8
s13207	156,779	28,986	81.5	29,257	81.3
s15850	58,603	24,694	57.9	26,814	54.2
s35932	25,934	20,433	21.2	20,337	21.6
s38417	165,336	91,408	44.7	98,584	40.4
s38584	168,547	77,834	53.8	80,075	52.5
Ave.			54.9		53.0

5.4　FFの評価値グラフの特性

図5.1に，制約なしでFFの自由な並べ換えを行った結果の各FFの評価値が大きいものから順にプロットしたグラフを示す．ほとんどの回路において図4.2と同じような形状を示しており，グラフの傾きを考慮してブロック分割する本提案手法が有効的に働く形状をしている．

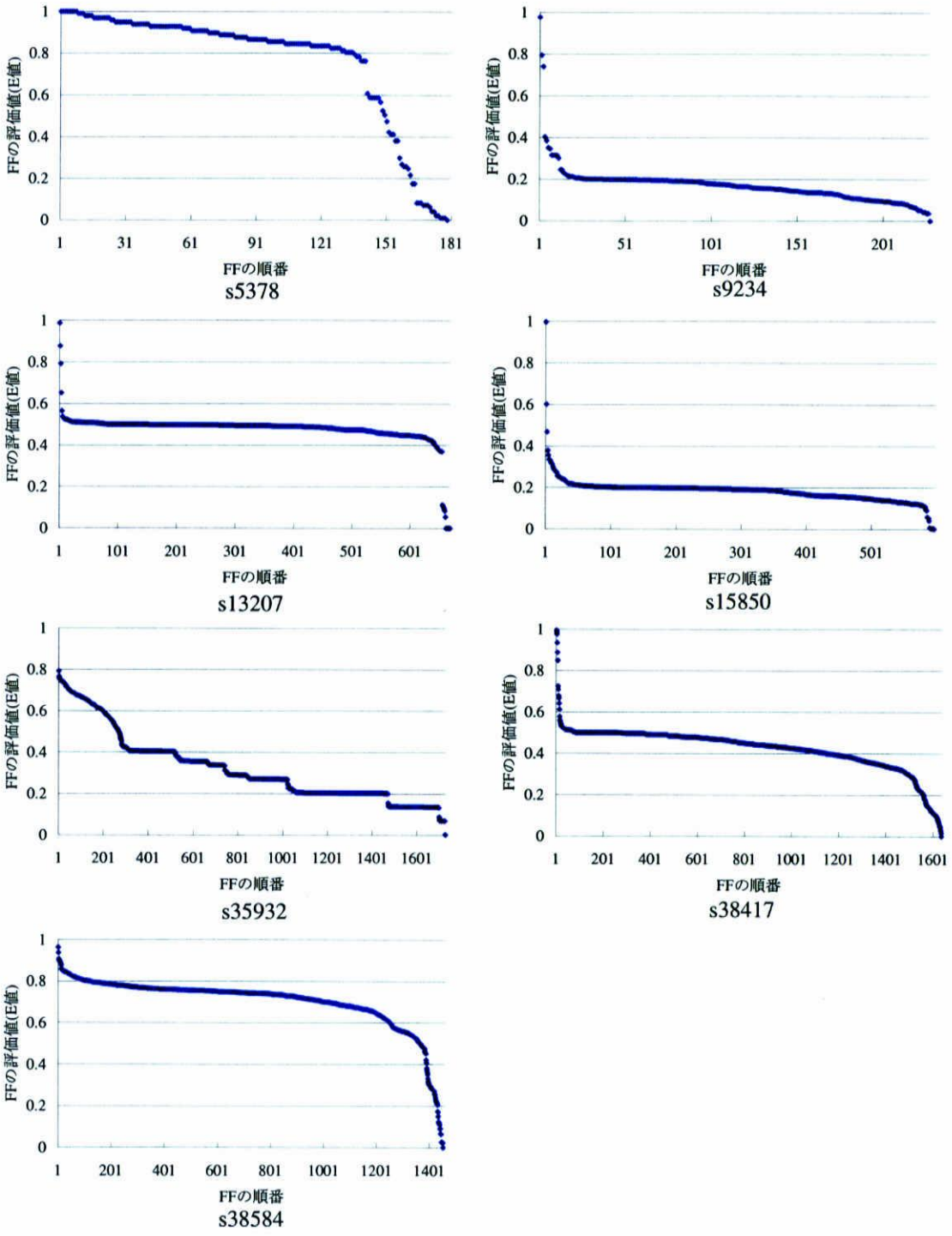

図 5.1: FF の評価値グラフ

第6章

第I部のまとめ

LSIテスト高速化のために，テスト応答の後部を次のテストベクトルの前部として利用することでスキャンシフト量を削減するテスト応答・テストベクトルオーバラッピング法（TRTVO法）が提案されている[1, 2, 9]．この手法は，データ展開圧縮回路を組み込む手法に比べテスト時間削減率は劣るものの，アーキテクチャが従来のフルスキャンテストと同じであるため，手軽に採用することができるという利点を有するとともに，チップコストの面でも優れている．

文献[1, 2]ではスキャンチェーン中のFFの並べ換えを行うことを利用したTRTVO法が提案されている．しかし，FFの並べ換えはLSIのレイアウト上の制約から常に自由に行えるわけではない．そこで文献[9]では，正当化技術とテストベクトル中のドントケアビットを利用し，スキャンチェーン中のFFの並べ換えを前提としない，TRTVO法の高速化のためのテスト入力系列の生成手法が提案されている．それらに対して，第I部では，スキャンチェーンの総結線長があまり長くならないようなFFの並べ換えであれば許容されることを仮定し，スキャンチェーンの総結線長を制約として与え，その制約の下でスキャンシフト量を最小化することを目的としたTRTVO法のための効果的なFFの並び順決定法を提案した．

本提案手法では，まず，自由なFF の並べ換えにより，総スキャンシフト量を最小にするスキャンチェーンFFの並び順を求める．ここでは，並び順を決定するための，文献[1, 2]とは異なる新しい評価関数を導入した．その後，その並び順を基にして，その総スキャンシフト量からの増加量ができるだけ小さくなるような，かつ，スキャンチェーン総結線長が制約として与えられた長さ以下になるような並び順を求める．ここでは，グラフ形状の乱れ具合に基づく新しい手法を考案した．このようにして得られたスキャンチェーンに対して，文献[9]で提案されているテスト入力系列生成手法をそのまま適用する．

ISCAS'89の大きい方の回路を用いた実験では，総結線長制約として元のスキャンチェーンの総結線長の2倍を与えた場合，制約なしに自由に並べ換えを行った場合に比べ，スキャンシフト量はほぼ同等程度で，スキャンチェーンの総結線長は1桁短縮されるとい

う結果を得ることができ，提案手法の有効性を確認した．

第II部

分割スキャンチェーンへの拡張

三重大学大学院　工学研究科

第7章

第II部の概要

第I部では，スキャンチェーン線長があまり長くならないようなFFの並べ換えであれば許容されると仮定し，総結線長の制約下での効果的なFFの並び順決定手法について述べた．それに対し第II部では，FFの並べ換えを行うことができないという前提で，TRTVO法の性能向上を目的とした手法を提案する．具体的な内容について以下に示す．

フルスキャンテストアーキテクチャでは1.3節の図1.3にあるように，スキャンチェーンは1本で構成されている（シングルスキャンチェーン）．そこで，このスキャンチェーンを複数本に分割し，分割された各スキャンチェーンに対してTRTVO法を用いることにより，その性能を向上させる．

また，TRTVO法ではテスト応答を一部しかスキャンアウトできないため，検出できない故障（非検出故障）が現れる可能性がある．文献[9]では，このような非検出故障に対する2つの救済法を提案し，どちらが効果的な救済法であるかについて議論している(2.2節)．しかしながら，これはシングルスキャンチェーンの場合を前提に議論を進めている．そこで，文献[9]で提案されている非検出故障に対する2つの救済法について，本提案手法の下での有効性について比較する．

以下，8章で提案手法について述べ，9章で実験結果を示す．10章では第II部のまとめを行う．

第8章

分割スキャンチェーンにおけるTRTVO法

本章では，スキャンチェーン中のFFの並べ換えを一切行わずに，TRTVO法の性能向上を目指した手法を提案する．

フルスキャンテストアーキテクチャでは1.3節の図1.3に示すように，スキャンチェーンは1本で構成されており，本論文で実験に用いている回路を例に挙げても，スキャンチェーン長（FF数）は10^2～10^3程度のオーダーである．TRTVO法では，テスト応答の後部（Scan inピン側）をオーバラップに利用するため，スキャンチェーン長が長い場合，Scan outピンに近い側のFFはオーバラップに有効利用されていないと考えられる．そこで，1本の長いスキャンチェーンをいくつかに分割し，分割されたそれぞれのスキャンチェーンに直接データ入力を行えるようなテストアーキテクチャ（以後，分割スキャンチェーンと記述）を提案する．分割スキャンチェーンに拡張することにより，シングルスキャンチェーンでは利用されていなかった，Scan outピンに近い側のFFもオーバラップに利用できるようになる．その結果，オーバラップ量が増加し，スキャンシフト量を削減できると考えられる．

以下，8.1節で基本原理について述べ，8.2節で分割スキャンチェーンのテストアーキテクチャとその動作について述べる．

8.1　基本原理

本節では，提案手法の基本原理について述べる．0，1，Xから構成される2本の長いベクトル列同士のオーバラップについて考える．

図8.1に示すように，ベクトルAとベクトルBをオーバラップさせる場合（1）と，ベクトルA，Bをそれぞれ2つに分割した，ベクトルA1とB1，A2とB2をオーバラップさせる場合（2）を考える．例えば3bitオーバラップする場合を考えると，(1）では，図8.2に示すように，ベクトルAの右方の3bitとベクトルBの左方の3bitがオーバラップする場合の1通りしかない．それに対して，(2）では，ベクトルA1とB1，及びベクト

ルA2とB2のオーバラップ量の合計が3bitになればよい．すなわち，図8.2に示すように，3bitオーバラップする方法は4通り考えることができる．このようにベクトルを分割し，それぞれでオーバラップさせる方が，オーバラップする「場合の数」が増えるため，オーバラップ量は増加すると考えられる．

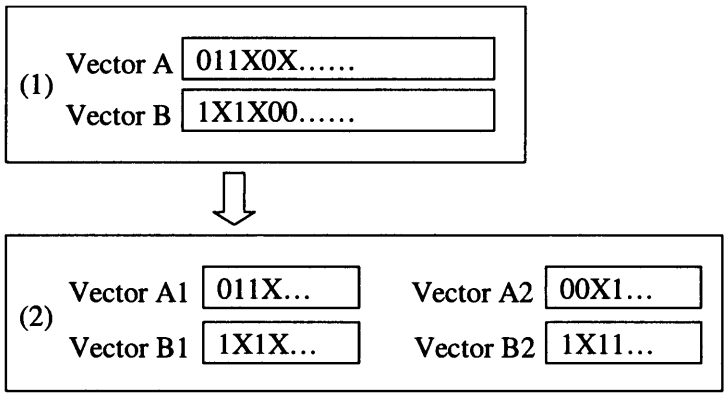

図 8.1: ベクトルの分割

8.2　テストアーキテクチャと動作

図8.3に本提案手法のテストアーキテクチャを示す．ここで，スキャンチェーンを分割すると，スキャンチェーンの分割位置に対応して，テストベクトルも分割される．スキャンチェーンとテスト集合との対応について図で表したものを図8.4に示す．また，以下にテスト時の動作について述べる．

各スキャンチェーンにデータを入力するときは，制御信号を用いてデータ入力の対象となるスキャンチェーンの切り替えを行う．このスキャンチェーンの切り替え，及びテストの実行の制御については，1本のS/C（シフト/チェンジ）信号線を用いる．具体的には，一つのスキャンチェーンに対してS信号（信号値0）を与えてデータ入力を行い，データ入力が完了したらC信号（信号値1）により次のスキャンチェーンに切り替え，次のスキャンチェーンのデータ入力に移る．このような動作を繰り返すことにより，すべてのスキャンチェーンへの入力を完了することができ，データ入力が完了したら，次のC信号で被検査回路である組合せ回路を動作させてテストを行う．図8.5にテスト時のタイムチャートを示す．

ここで，上記のテスト動作ではスキャンチェーンの切り替え毎に制御信号を1bit必要とする．すなわち，スキャンチェーンの分割数の増加に伴い，制御信号入力量によるオーバヘッドも増加する．

・3bitオーバラップする場合

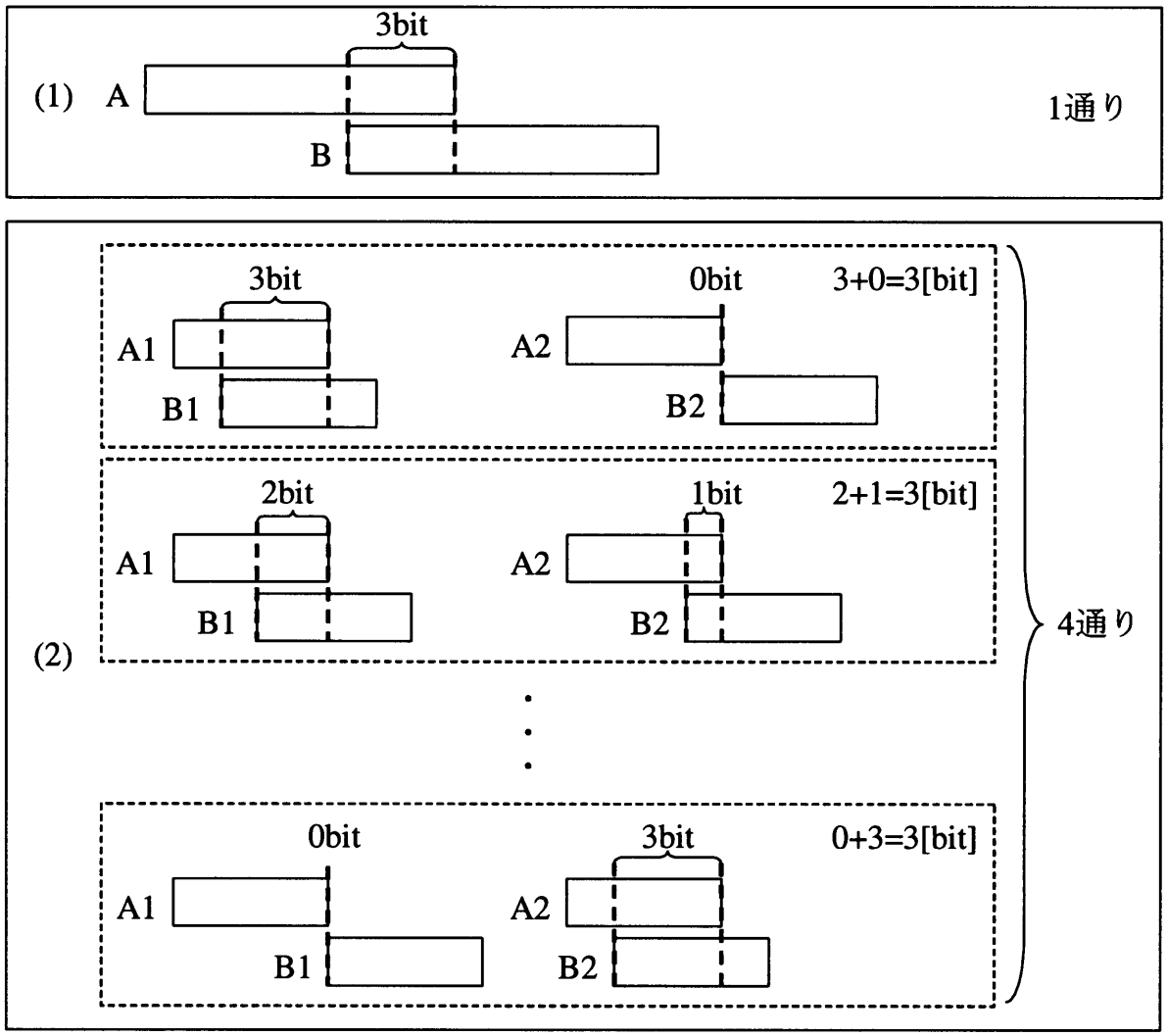

図 8.2: オーバラップ量増加

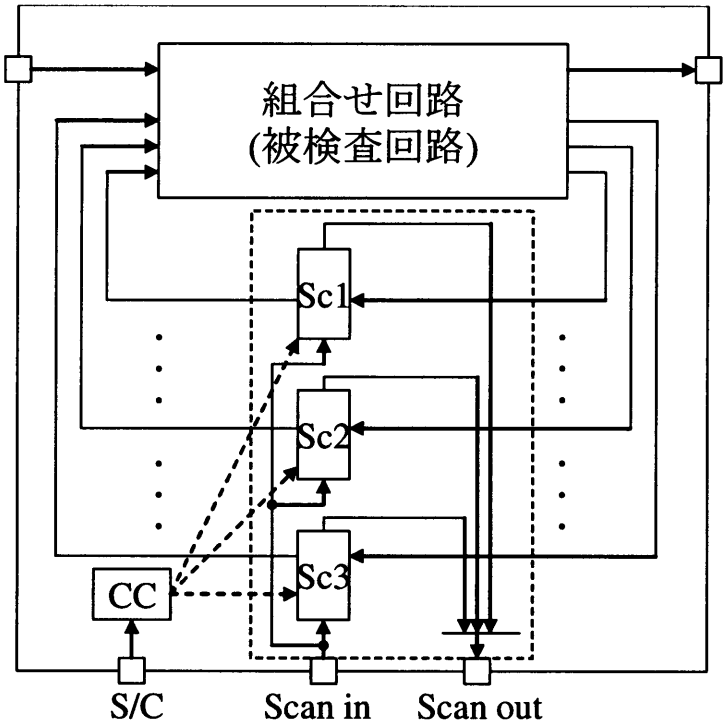

図 8.3: 分割スキャンチェーン用テストアーキテクチャ

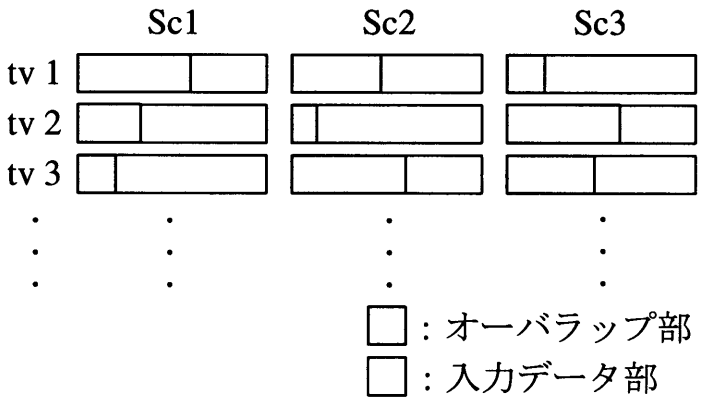

図 8.4: テスト集合とスキャンチェーン

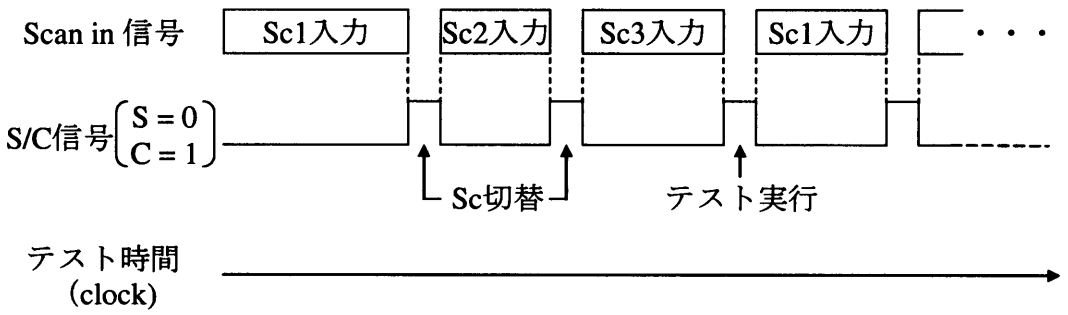

図 8.5: テスト時のタイムチャート

第9章

結果と考察

上記の手法をUNIX上にC言語で実装し，ISCAS'89ベンチマーク回路に対して実験を行った．使用した計算機は，第I部と同様に1.9 GHzのIntel Pentium 4プロセッサと514 MBの主メモリを持つものである．実験に用いたテスト集合についても同様に，文献[8]のシステムにより生成されたテスト集合を用いた（表5.1）．

9.1　非検出故障の救済法の比較

2.2節で述べた非検出故障の救済法であるスキャンシフト量増加法とテストベクトル追加法の2つに対して分割スキャンチェーンでの実験結果を表9.1に示す．ここで，今回の結果はスキャンチェーンの分割数は20で実験を行い，テスト時間削減率Rは（5.1）式を用いて算出した値である．

表 9.1: 非検出故障の各救済法によるテスト時間削減率R[%]

回路	シングル スキャンチェーン		分割スキャンチェーン 分割数:20	
	シフト量増加	ベクトル追加	シフト量増加	ベクトル追加
s5378	41.3	43.1	67.4	44.3
s9234	30.1	36.0	59.4	40.3
s13207	55.6	60.2	86.3	81.4
s15850	32.0	46.4	70.6	54.6
s35932	15.8	14.7	34.9	27.9
s38417	20.3	24.3	48.0	38.1
s38584	23.6	40.2	57.8	51.6
Ave.	31.2	37.8	60.6	48.3

実験結果から，シングルスキャンチェーンでは文献[9]にあるように，オーバラップベクトル追加法が良好な結果を得ている．しかしながら，スキャンチェーンを分割する本提案手法では，スキャンシフト量増加法が良好な結果となった．

非検出故障の救済時には，テストベクトルのすべてのドントケアビットには0か1の論理値が割り当てられている．そのため，スキャンシフト量増加法では，スキャンチェーンのFF数が大きい場合，シフト量を増加させる際にマッチングが成功しにくいため，大幅にシフト量が増加する．しかしながら，スキャンチェーンを分割する本提案手法では，一つ一つのスキャンチェーンのFF数が小さくなるため，シフト量増加の影響が小さくなり，良好な結果となったと考えられる．

以後，分割スキャンチェーンでの実験では，非検出故障の救済法としてシフト量増加法を用いた結果を表記する．

9.2　実験結果

本提案手法の実験結果を表9.2に示し，その結果をグラフで表したものを図9.1に示す．

表 9.2: 実験結果（スキャンチェーン分割）

分割数	テスト時間削減率R[%]						
	回路						
	s5378	s9234	s13207	s15850	s35932	s38417	s38584
1	41.3	30.1	55.6	32.0	15.8	20.3	23.6
2	43.5	36.0	71.3	45.8	19.5	25.4	32.5
4	55.2	49.6	77.4	56.1	22.4	37.4	46.3
8	67.0	54.5	82.3	61.6	25.0	41.9	49.0
16	66.7	61.1	83.9	70.5	33.0	48.3	54.2
32	63.1	61.9	85.2	72.1	39.8	51.8	58.8
50	55.2	57.9	84.5	72.0	42.2	53.5	59.1
100	30.7	39.6	78.5	68.1	42.9	57.4	65.0

小さな回路（s5378，s9234）では，8.2節で述べたスキャンチェーンの分割数の増加に伴う，スキャンチェーン切り替えのための制御信号の増加の影響が顕著に表れており，分割数が大きいとテスト時間削減率が非常に悪くなっている（図9.1）．最大の3つの回路（s35932，s38417，s38584）では，分割数30程度からテスト時間削減率はほぼ一定値となり，分割数をさらに増加させても結果はあまり良くなっていない．

次に分割スキャンチェーンでのテスト時間削減率Rを評価するために，s38417，s38584の2つの回路において，第I部で提案したシングルスキャンチェーンでのFF並べ換えとの結果を表にまとめたものを表9.3に示す．ここで，スキャンチェーンの分割数は20分割の結果を表記している．

表9.3より，最大の2つの回路（s38417，s38584）において，本提案手法ではFFの並

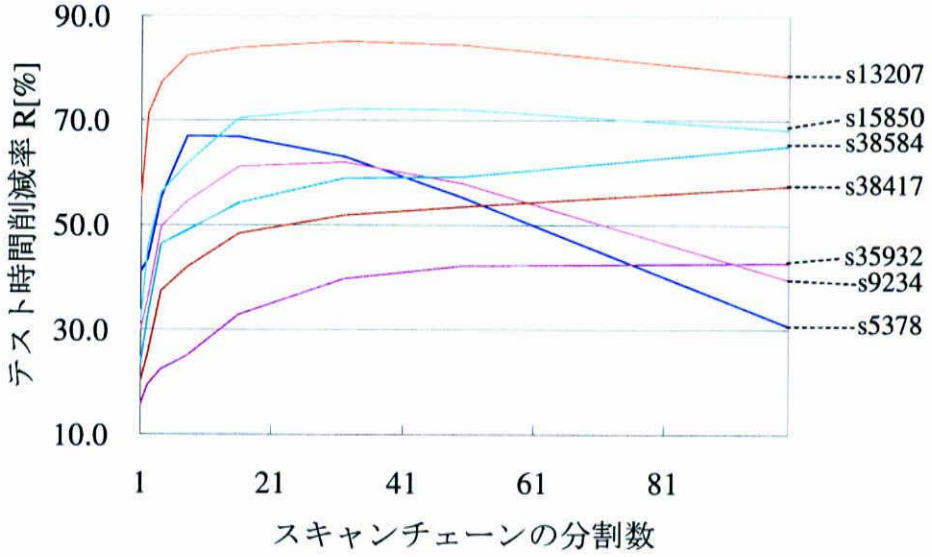

図 9.1: スキャンチェーンの分割数とテスト時間削減率

べ換えなしで，シングルスキャンチェーンでの自由なFF並べ換えよりも良好な結果を得ており，本提案手法の有効性を確認できた．

表 9.3: シングルスキャンチェーンでのFF並べ換え手法（4章）との比較

(a)s38417

FF並べ換え	シングル スキャンチェーン	分割スキャンチェーン 分割数:20
	R[%]	R[%]
なし	23.5	48.0
制約付	42.2	—
あり（自由）	44.7	—

(b)s38584

FF並べ換え	シングル スキャンチェーン	分割スキャンチェーン 分割数:20
	R[%]	R[%]
なし	38.4	57.8
制約付	52.1	—
あり（自由）	53.8	—

第10章

第II部のまとめ

LSIテスト高速化のために，テスト応答の後部を次のテストベクトルの前部として利用することでスキャンシフト量を削減するテスト応答・テストベクトルオーバラッピング法（TRTVO法）が提案されている[1, 2, 9]．この手法は，データ展開圧縮回路を組み込む手法に比べテスト時間削減率は劣るものの，アーキテクチャが従来のフルスキャンテストと同じであるため，手軽に採用することができるという利点を有すると共に，チップコストの面でも優れている．

文献[1, 2]ではスキャンチェーン中のFFの並べ換えを行うことを利用したTRTVO法が提案されている．しかし，FFの並べ換えはLSIのレイアウト上の制約から常に自由に行えるわけではない．そこで文献[9]では，正当化技術とテストベクトル中のドントケアビットを利用し，スキャンチェーン中のFFの並べ換えを前提としない，TRTVO法の高速化のためのテスト入力系列の生成手法が提案されている．本論文の第I部では，スキャンチェーン線長があまり長くならないようなFFの並べ換えであれば許容されると仮定し，総結線長の制約下での効果的なFFの並び順決定手法について述べた．それに対して，第II部では，FFの並べ換えを全く行わずに更なるTRTVO法の性能向上を目的とし，シングルスキャンチェーンから分割スキャンチェーンへのテストアーキテクチャの拡張を提案した．

また，文献[9]で提案されている非検出故障に対する2つの救済法について，分割スキャンチェーン環境下での有効性の比較を行った．

その結果，本提案手法では，非検出故障の救済法ではスキャンシフト量増加法の方が良好な結果を得ることができた．テスト時間削減率では，FF並べ換えなしでも，第I部で提案したFFの総結線長制約下でのFF並べ換え手法よりも良好な結果を得ることができ，本提案手法の有効性を確認した．

第11章

むすび

本論文では，フルスキャンテスト高速化のための，テスト応答の後部を次のテストベクトルの前部として利用することによってスキャンシフト量を削減するTRTVO法の更なる性能向上を目指した手法として，2つの内容を第I部と第II部にわけて提案した．

第I部で提案したFFの総結線長制約下でのFF並べ換え手法では，ISCAS'89の大きい方の回路を用いた実験では，総結線長制約として元のスキャンチェーンの総結線長の2倍を与えた場合，制約なしに自由に並べ換えを行った場合に比べ，スキャンシフト量はほぼ同等程度で，スキャンチェーンの総結線長は1桁短縮されるという結果を得ることができ，提案手法の有効性を確認した．

第II部で提案した分割スキャンチェーンへの拡張では，FF並べ換えなしでも，第I部で提案したFFの総結線長制約下でのFF並べ換え手法よりも良好な結果を得ることができ，提案手法の有効性を確認した．

本論文で提案した分散スキャンチェーンへの拡張は，FFの並べ換えを全く行っていない．そのため，本手法にFFの並べ換えを適用することにより，スキャンシフト量をさらに削減できると考えられる．今後の課題として，分割スキャンチェーン環境化で，総結線長制約下でのFF並べ換え手法を適用することが挙げられる．

参考文献

[1] Y.Higami, S.Kajihara and K.Kinoshita, " Reduced Scan Shift: A New Testing Method for Sequential Circuits ", International Test Conference, pp.624-630, 1994.

[2] Y.Higami, S.Kajihara and K.Kinoshita, " A Reduced Scan Shift Method for Sequential Circuit Testing ", IEICE Transaction on Fundamentals, Vol.E77A, No.12, pp.2010-2016, 1994.

[3] S.Makar, " A Layout-Based Approach for Ordering Scan Chain Flip-Flops ", International Test Conference, pp.341-347, 1998.

[4] C.Barnhart, V.Brunkhorst, F.Distler, O.Farnsworth, B.Keller and B.Koenemann, " OPMISR:The Foundation for Compressed ATPG Vectors ", International Test Conference, pp748-757, 2001.

[5] T.Hiraide, K.O.Boateng, H.Konishi, K.Itaya, M.Emori, H.Yamamura and T.Mochiyama, " BIST-Aided Scan Test - A New Method for Test Cost Reduction ", 21st VLSI Test Symposium, pp.359-364, 2003.

[6] S.Kajihara and K.Miyase, " On Identifying Don't Care Inputs of Test Patterns for Combinational & Full-Scan Sequential Circuits ", International Conference on Computer-Aided Design, pp.364-369, 2002.

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

[8] T.Hayashi, Y.Morimoto, T.Shinogi, H.Kita and H.Takase, " X-Maximal Test Set Generation for Combinational Circuits ", 3rd Workshop on RTL and High Level Testing, S4-2, 2002.

[9] 山田宏行，篠木剛，京谷忠雄，林照峯，鶴岡信治，"テスト応答・テストベクトルオーバラッピングLSI検査法のためのテスト入力系列生成手法"，電子情報通信学会和文論文誌D，Vol.J89-D，No.8，August 2006.

[10] Nur A. Touba, " Survey of Test Vector Compression Techniques " , IEEE Design & Test of Computers, Vol.23, No.4, pp.294-303, 2006.

[11] I. Bayraktaroglu and A. Orailoglu, " Concurrent Application of Compaction and Compression for Test Time and Data Volume Reduction in Scan Designs, " IEEE Trans. Computers, Vol.52, No.11, pp.1480-1489, 2003.

[12] C.V. Krishna and N.A. Touba,"Adjustable Width Linear Combinational Scan Vector Decompression, " Proc. Int ' l Conf. Computer-Aided Design (ICCAD 03), pp.863-866, 2003.

[13] A. Chandra and K. Chakrabarty, " System-on-a-Chip Test-Data Compression and Decompression Architectures Based on Golomb Codes " , IEEE Trans. Computer-Aided Design, Vol.20, No.3, pp.355-368, 2001.

[14] P.T. Gonciari, B.M. Al-Hashimi, and N. Nicolici, " Variable-Length Input Huffman Coding for System-on-a-Chip Test " , IEEE Trans. Computer-Aided Design, Vol.22, No.6, pp.783-796, 2003.

[15] S.M. Reddy et al.,"On Test Data Volume Reduction for Multiple Scan Chain Designs " , Proc. 20th VLSI Test Symp. (VTS 02), pp.103-108, 2002.

[16] B. Koenemann, " LFSR-Coded Test Patterns for Scan Designs " , Proc. European Test Conf. (ETC 91), pp.237-242, 1991.

[17] B. Koenemann et al., " A SmartBIST Variant with Guaranteed Encoding " , Proc. 10th Asian Test Symp. (ATS 01), pp.325-330, 2001.

[18] C.V. Krishna, A. Jas, and N.A. Touba, " Test VectorEncoding Using Partial LFSR Reseeding " , Proc. Int ' l Test Conf. (ITC 01), pp.885-893, 2001.

謝辞

本研究の遂行および修士論文の作成にあたり，丁寧なご指導とご助言を頂きました本学工学部電気電子工学科の，鶴岡信治教授，篠木剛助教授，川中普晴助手に感謝致します．そして貴重な時間を割いて本論文を査読して頂いた本学工学部電気電子工学科 林照峯教授に深く感謝致します．また，共に研究に携わり御助言をいただいた本学博士前期課程電気電子工学専攻 東海林正和氏，坂口敏雄氏に感謝致します．さらに，違う研究分野ながらも様々な意見を交えた 伊藤聖太郎氏，大谷芳弘氏，岡山陽介氏，安井良氏，吉田大祐氏，伊藤哲也氏，大國聖治，柴田彰洋氏，水谷洋輔氏，梁修孝氏，Premachandra Halpage Chinthaka 氏に深く感謝致します．本論文は多くの方々の多大な御協力をいただいた結果完成したものであり，これらの方々なしでは完成できませんでした．御協力いただいたすべての方々に感謝致します．

発表論文リスト

学術論文

(1) 山田宏行，篠木剛，京谷忠雄，林照峯，鶴岡信治：“テスト応答・テストベクトルオーバラッピングLSI検査法のためのテスト入力系列生成手法”，電子情報通信学会和文論文誌D，Vol.J89-D，No.8，August 2006

(2) 京谷忠雄，篠木剛，山田宏行，東海林正和，林照峯，川中普晴，鶴岡信治：“テスト応答・テストベクトルオーバラッピングLSI検査法のためのスキャンチェーン線長を考慮したスキャンFFの並べ換え手法”，情報処理学会論文誌（条件付採録）

国際会議

(1) Tsuyoshi Shinogi, Tadao Kyotani, Masakazu Tokairin, Terumine Hayashi, Hiroharu Kawanaka, Shinji Tsuruoka, "Scan Chain Flip-Flop Reordering Method of Considering Wiring Length for Test Response Test Vector Overlapping Testing", IEEE 7th Workshop on RTL and High Level Testing(WRTLT06), pp.63-68, November 2006

国内会議

(1) 京谷忠雄，山田宏行，篠木剛，鶴岡信治：“ベクトルオーバラップによるLSIテスト時間短縮手法”，平成17年度 電気関係学会東海支部連合大会，O-296，2005.9

(2) 京谷忠雄，山田宏行，篠木剛，鶴岡信治：“ベクトルオーバラッピングLSIテスト法における非検出故障の救済手法”，平成17年度 三重地区計測制御研究講演会 講演論文集，(B13-1)-(B13-4)，2005.12

(3) 京谷忠雄，篠木剛，川中普晴，鶴岡信治：“LSIテスト高速化のための線長制約を考慮したスキャンFFの並べ換え手法”，平成18年度 電気関係学会東海支部連合大会，O-357，2006.9，連合大会奨励賞受賞

(4) 京谷忠雄，篠木剛，林照峯，川中普晴，鶴岡信治：“テスト応答・テストベクトルオーバラッピング法によるスキャンシフト量削減手法について”，平成19年度 第56回FTC研究会，2007.1