

テスト応答・テストベクトルオーバーラッピング 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

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[†]

LSI の大規模化にともない、フルスキャンテストの実行時間の増加が問題となっており、フルスキャンテストの高速化が望まれている。そのための手法として、フルスキャンアーキテクチャのままハードウェアを新しく追加することなく、高速化する手法が提案されている。これは、テスト応答の一部を次のテストベクトルの一部として利用することでスキャンシフト量を削減する手法である。この手法の性能向上のため、スキャンチェーン中の FF の並べ換えを利用した手法が提案されているが、FF の並べ換えは LSI のレイアウト上の制約からつねに自由に行えるわけではない。本論文では、スキャンチェーンの総結線長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 FF の並び順の決定法を提案する。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た。

Scan-FF Reordering Method for Reducing LSI Test Application Time Considering Wire Length in Test Response Test Vector Overlapping Testing

TADAO KYOTANI,[†] TSUYOSHI SHINOBI,[†] HIROYUKI YAMADA,[†]
MASAKAZU TOKAIRIN,[†] TERUMINE HAYASHI,[†]
HIROHARU KAWANAKA[†] and SHINJI TSURUOKA[†]

Researches for reducing the manufacturing testing cost of full-scanned LSI circuits are actively carried out. A method to reduce the test application time needed in full-scan testing without any additional LSI hardware cost has been proposed. The basic idea is utilizing a part of the current test response on the scan chain as a part of the next test vector so as to reduce the shift amount. However, their method relies on scan chain FF reordering. The unrestricted reordering is not always applicable owing to the area overhead of stitching wires for connecting flip-flops to form the scan chain. In this paper, we propose a method of restricted scan chain FF reordering for reducing the scan shift amount further under constraint of the wiring length of the scan chain. In each of the three largest ISCAS'89 circuits, the total wiring length of the scan chain obtained by the restricted reordering gets an order of magnitude shorter than by the unrestricted reordering, while the difference of the scan shift amounts by these two reordering methods is very small.

1. ま え が き

現在、論理回路 LSI の製造時検査 (LSI テスト) では、フルスキャンテスト法による論理テストが広く行われている。フルスキャンテストのアーキテクチャは、被検査回路とテストデータ (テストベクトル) を入力するためのスキャンチェーンと呼ばれるシフトレジスタで構成されている (図 1)。データ入力部がシフトレジスタであるため、テスト実行時間はテストデータ

[†] 三重大学

Mie University

現在、ブラザー工業株式会社

Presently with BROTHER INDUSTRIES, LTD.

現在、ソニー株式会社

Presently with SONY CORPORATION

現在、株式会社デンソー

Presently with DENSO CORPORATION

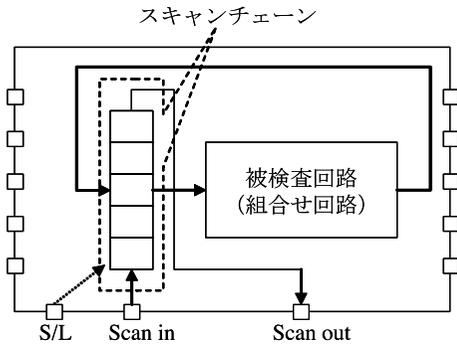


図1 フルスキャンテストアーキテクチャ
Fig.1 Full-scan test architecture.

の入力時間とほぼ等しくなり、スキャン入力ピン数が1である場合、入力するテストデータ量（スキャンシフト量）とほぼ等しくなる。一方、近年のLSIの大規模化にともないスキャンチェーン長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。また、製品コストを考えると、テストに割くことができるコストは限られており、なるべく低コストのテスト手法が望まれている。

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)~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トからのテストデータ供給量を削減する手法^{16)~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いっさい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て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自由に自由に行えるわけではない³⁾。

百万ゲートを超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)~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。

文献9)ではスキャンチェーン中のFFの並べ換えをいっさい行わないという条件の下で議論を展開しているが、それに対して、本論文では、スキャンチェーンの総結線長があまり長くないような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。

具体的な手法は次のようなものである。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。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適用する。

以下、2章では、文献1),2)で提案されているデータ展開圧縮回路を追加しない高速化手法（スキャンシフト量削減手法）を述べた後、それを基礎とした、文献9)で提案されている正当化技術とドントケアを利用した手法にふれる。3章で我々の提案手法を述べ、4章で実験結果を示し、5章でまとめを行う。

2. スキャンシフト量削減手法^{1),2),9)}

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スキャンシフト量を削減す

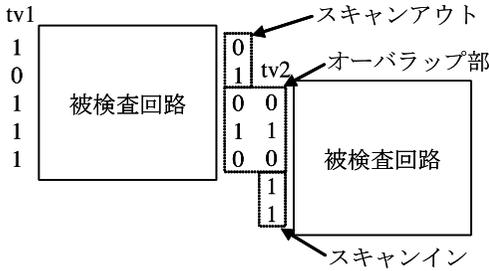


図2 テスト応答・テストベクトルオーバーラッピング
Fig.2 Test Response Test Vector Overlapping.

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)}

(a) 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 1 つ前のテスト応答が格納されている . RSS の基幹アイデアは, 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 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ーラッピング法 (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 (b) (c) で述べる手法が用いられている .

	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

図3 テストベクトルとテスト応答
Fig.3 Test vectors and faulty responses.

(b) 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 .

(c) 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 3 を用いて説明する . 図 3 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, 式 (1) より各 FF の評価値を定義している .

$$E(FF_i) = VC(FF_i) - VO(FF_i) \tag{1}$$

ここで, 式 (1) 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 3 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 .

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る、レイアウト上の制約から必ずしも並べ換えがつね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救済法を議論し、縮退故障検出効率 100%を保証している。この手法では、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。

3. スキャンチェーン FF 並べ換え手法

文献 9) では、スキャンチェーン中の FF の並べ換えをいっさい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、実験結果を示している。もし、FF を都合良く並べ換えれば、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減らすことができるが、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。本論文では、スキャンチェーンの総結線長があまり長くないような 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（3.1 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（3.2 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、3.1 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、3.2 節では、3.1 節

から求めた FF の並び順を、総結線長が制約として与えられた長さ以下になるように変更する手法について説明する。

3.1 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、オーバー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（式 (1) 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。

(a) 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ー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、オーバー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式 (2) より与える。

$$DC(FF_i) = FF_i \text{ のドントケアビットの数} \quad (2)$$

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ーラップに有利となる。

(b) 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 を用いて説明する。図 4 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 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、故障ごとの総検出回数に基づいて、その故障の

	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 故障検出表

Fig. 4 Fault detection table.

検出難易度を式 (3) より定義する.

$$\text{検出難易度}(f_i) = \frac{1}{(f_i \text{の総検出回数})^n} \quad (3)$$

n は正の整数で経験的に定める (実験では $n = 2$). 式 (3) 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 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式 (4) より各 FF の評価値 FI を決定する.

$$FI(FF_i) = \sum_{\text{検出難易度}} FF_i \text{が検出する故障の} \quad (4)$$

式 (4) 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 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.

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 はそれぞれまったく違う単位であり, 単純な評価値の足し合わせには問題がある. そこで, まず式 (5), (6) を用いて DC と FI の値を 0~1 に正規化を行う.

$$DC'(FF_i) = \frac{DC(FF_i) - DC_{min}}{DC_{max} - DC_{min}} \quad (5)$$

$$FI'(FF_i) = \frac{FI(FF_i) - FI_{min}}{FI_{max} - FI_{min}} \quad (6)$$

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式 (7) により決定する.

$$E(FF_i) = \alpha \times DC'(FF_i) + (1 - \alpha) \times FI'(FF_i) \quad (7)$$

ここで E は FF_i の評価値, $\alpha, 1 - \alpha$ 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.

3.2 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, 3.1 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.

(a) 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 (7) 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 5 (a) のようになったとする. 図 5 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 5 (b)), 各ブロックごとに, 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 5 (b) 内に示す波線長方形の内部に限定される.

図 5 (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 を持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 を持つブロックでは, 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が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

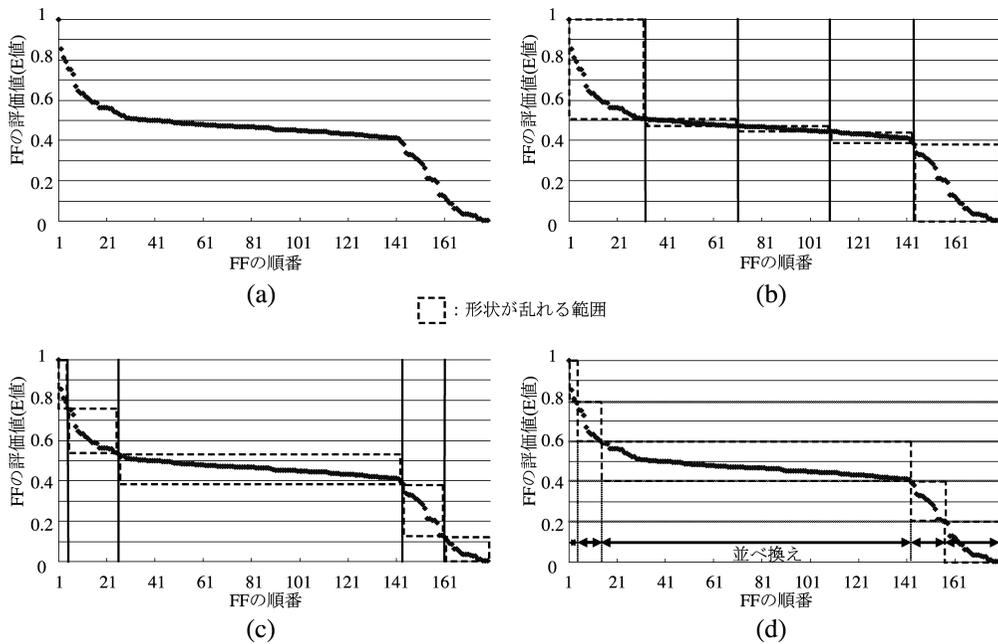
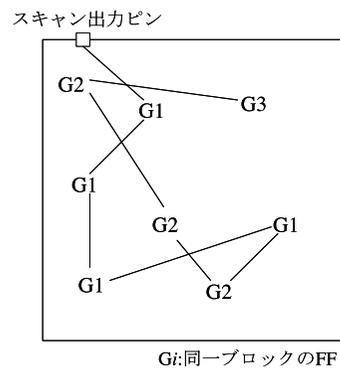


図 5 グラフ形状に基づくスキャンチェーン分割法
Fig. 5 Graph chart oriented scan chain partitioning method.

るようにグラフを分割すれば (図 5 (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. たとえば図 5 (d) のように, 評価値 (縦軸) が $0 \sim 0.2, 0.2 \sim 0.4, \dots, 0.8 \sim 1.0$ の FF を持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.

(b) 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, 1 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



Gi: 同一ブロックの FF
図 6 ブロック別グリーディ法
Fig. 6 Greedy reordering in each block.

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 6).

(c) 二分探索法

これまでに本節 (a) (b) 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, グラフの形状は乱れにくいので, 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

```

#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 7 二分探索法によるスキャン FF の並び順決定フロー
Fig. 7 Flow of scan-FF reordering by binary search algorithm.

割数を効率良く見つけるために、図 7 に示す二分探索法を用いる。

#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限値と上限値をそれぞれ #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、下限値と上限値の差が 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 1 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つね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。

4. 実験結果

上記の手法を 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、

表 1 テストベクトル集合

Table 1 Test vector set.

circuits	#TV	#don't-care [%]
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

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 1 に実験に用いたテスト集合を示す。これは文献 8) のシステムにより生成されたものである。表 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 (3.1 節の手法を用いる) の結果を示す。表 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、式 (8) で定義される。

$$R = \frac{Lf - L}{Lf} \times 100 \quad [\%] \quad (8)$$

式 (7) の α は、各回路で、それぞれ $\alpha=0.0, 0.2, 0.4, 0.5, 0.6, 0.8, 1.0$ の 7 通りを行い、最もテスト時間削減率の高い値を採用した。表 2 にその α の値を示す。

表 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 つの回路では約 20 倍となっ

表 2 実験結果
Table 2 Experimental results.

circuit	Lf	FF並べ換えなし(文献 ²⁾)			自由なFF並べ換え手法					
		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												
circuit	#SC	制約 : W/Wf = 1.5					制約 : W/Wf = 2.0					
		L	R [%]	W	W/Wf	T [sec]	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			

表 3 縦軸等分割法と横軸等分割法との比較
Table 3 Comparison of vertical division with horizontal division.

circuit	Lf	縦軸等分割法(提案法)					横軸等分割法				
		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)

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 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分かる。

また、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 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、3.2 節

表 4 異なる評価関数による実験結果

Table 4 Experimental result with different evaluation functions.

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

(a)(b) 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分かる。

我々の提案手法では、3.1 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式 (3) を用いて計算し、それに基づき、最終的に式 (7) 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

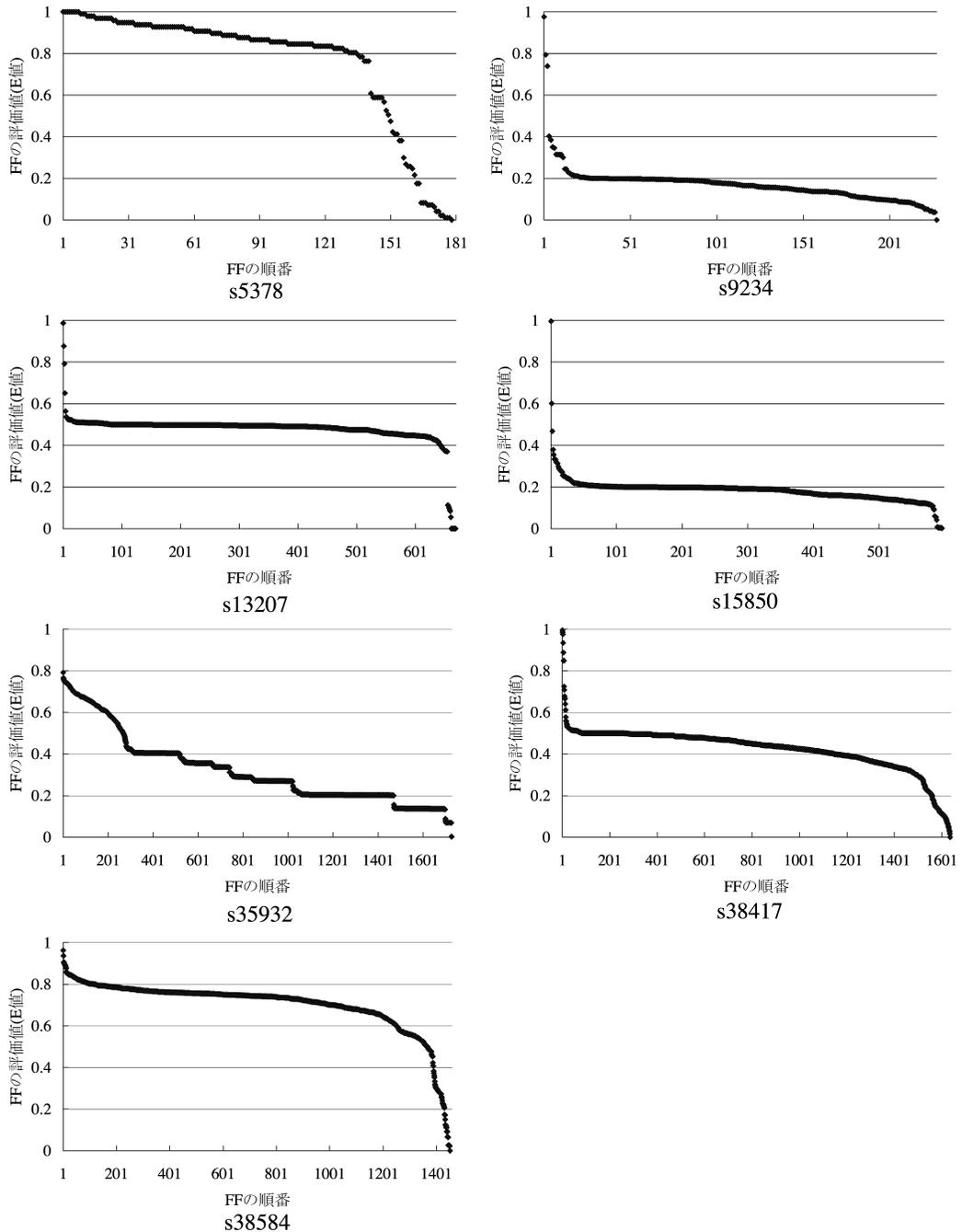


図 8 FF の評価値グラフ
Fig. 8 Graph chart of evaluation values.

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、1 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。式 (3) の検出難易度を、1 回検出故障は 1、複数回検出故障は 0 と定義する。表 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。

図 8 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 5 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。

5. む す び

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ーラッピング法 (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つね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 法の高速度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、本論文では、スキャンチェーンの総結線長があまり長くないような 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。

今回、グラフ分割法として単純な縦軸等分割法を採用したが、より効果的なグラフ分割法の検討が今後の

課題としてあげられる。

参 考 文 献

- 1) Higami, Y., Kajihara, S. and Kinoshita, K.: Reduced Scan Shift: A New Testing Method for Sequential Circuits, *International Test Conference*, pp.624–630 (1994).
- 2) Higami, Y., Kajihara, S. and Kinoshita, K.: A Reduced Scan Shift Method for Sequential Circuit Testing, *IEICE Trans. Fundamentals*, Vol.E77A, No.12, pp.2010–2016 (1994).
- 3) Makar, S.: A Layout-Based Approach for Ordering Scan Chain Flip-Flops, *International Test Conference*, pp.341–347 (1998).
- 4) Barnhart, C., Brunkhorst, V., Distler, F., Farnsworth, O., Keller, B. and Koenemann, B.: OPMISR: The Foundation for Compressed ATPG Vectors, *International Test Conference*, pp.748–757 (2001).
- 5) Hiraide, T., Boateng, K.O., Konishi, H., Itaya, K., Emori, M., Yamamura, H. and Mochiyama, T.: BIST-Aided Scan Test — A New Method for Test Cost Reduction, *21st VLSI Test Symposium*, pp.359–364 (2003).
- 6) Kajihara, S. and Miyase, K.: 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) El-Maleh, A. and Al-Suwaiyan, A.: An Efficient Test Relaxation Technique for Combinational & Full-Scan Sequential Circuits, *20th VLSI Test Symposium*, pp.53–59 (2002).
- 8) Hayashi, T., Morimoto, Y., Shinogi, T., Kita, H. and Takase, H.: 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ーラッピング 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 (Aug. 2006).
- 10) Toubas, N.A.: Survey of Test Vector Compression Techniques, *IEEE Design & Test of Computers*, Vol.23, No.4, pp.294–303 (2006).
- 11) Bayraktaroglu, I. and Orailoglu, A.: 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) Krishna, C.V. and Toubas, N.A.: Adjustable Width Linear Combinational Scan Vector Decompression, *Proc. Int'l Conf. Computer-Aided*

Design (ICCAD 03), pp.863–866 (2003).

- 13) Chandra, A. and Chakrabarty, K.: 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) Gonciari, P.T., Al-Hashimi, B.M. and Nicolici, N.: 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) Reddy, S.M., et al.: On Test Data Volume Reduction for Multiple Scan Chain Designs, *Proc. 20th VLSI Test Symp. (VTS 02)*, pp.103–108 (2002).
- 16) Koenemann, B.: LFSR-Coded Test Patterns for Scan Designs, *Proc. European Test Conf. (ETC 91)*, pp.237–242 (1991).
- 17) Koenemann, B., et al.: A SmartBIST Variant with Guaranteed Encoding, *Proc. 10th Asian Test Symp. (ATS 01)*, pp.325–330 (2001).
- 18) Krishna, C.V., Jas, A. and Toubia, N.A.: Test VectorEncoding Using Partial LFSR Reseeding, *Proc. Int'l Test Conf. (ITC 01)*, pp.885–893 (2001).

(平成 18 年 9 月 19 日受付)

(平成 19 年 3 月 1 日採録)



京谷 忠雄

平成 17 年三重大学工学部電気電子工学科卒業。平成 19 年同大学大学院工学研究科電気電子工学専攻博士前期課程修了。同年ブラザー工業株式会社入社。



篠木 剛 (正会員)

昭和 52 年東京工業大学理学部情報科学科卒業。昭和 54 年同大学大学院理工学研究科修士課程修了。同年(株)富士通研究所入社。昭和 60 年より 1 年間米国オレゴン大学客員研究員。平成 10 年三重大学大学院工学研究科博士後期課程修了。現在、三重大学大学院工学研究科助教授。工学博士。LSI のテストや設計支援、並列/分散計算機システム、画像処理等に興味を持つ。電子情報通信学会、IEEE 各会員。



山田 宏行

平成 16 年三重大学工学部電気電子工学科卒業。平成 18 年同大学大学院工学研究科電気電子工学専攻博士前期課程修了。同年ソニー株式会社入社。



東海林正和

平成 17 年三重大学工学部電気電子工学科卒業。平成 19 年同大学大学院工学研究科電気電子工学専攻博士前期課程修了。同年株式会社デンソー入社。



林 照峯 (正会員)

昭和 44 年名古屋大学工学部電気工学科卒業。昭和 46 年同大学大学院工学研究科修士課程修了。同年(株)日立製作所日立研究所入社。平成 5 年三重大学工学部電気電子工学科教授。平成 18 年より三重大学大学院工学研究科教授。工学博士。LSI の設計自動化、論理回路のテスト等の研究に従事。電子情報通信学会、IEEE 各会員。



川中 普晴

平成 11 年三重大学工学部電気電子工学科卒業。平成 16 年同大学大学院工学研究科博士後期課程修了。同年同大学院医学系研究科博士課程入学。同年(株)医用工学研究所取締役。平成 17 年(株)三重 TLO (NEDO 養成技術者), 三重大学社会連携フェロー。平成 18 年三重大学大学院工学研究科助手。ソフトコンピューティングとその応用, 医療情報学に興味と持つとともに, 産学官連携活動に従事。電子情報通信学会, 電気学会, 日本知能情報ファジィ学会, 日本医療情報学会各会員。



鶴岡 信治 (正会員)

昭和 52 年岐阜大学工学部電子工学科卒業, 昭和 54 年名古屋大学大学院工学研究科修士課程修了。同年三重大学工学部電子工学科助手, 平成元年同助教授, 平成 12 年より三重大学工学部電気電子工学科教授。平成 3~4 年米国ミシガン大学客員助教授。平成 18 年より三重大学大学院工学研究科教授。工学博士。文書理解, 動画像認識, 医用信号処理等のパターン認識問題に関心を持つ。電子情報通信学会, 映像情報メディア学会, 計測自動制御学会, IEEE 各会員。