

非冗長組合せ回路と極小テスト集合の同時生成について

梶原 誠 司[†] 樹 下 行 三[†]

組合せ回路の冗長部分は検出不可能な単一縮退故障として識別できるため、テスト生成による回路簡単化が可能である。テスト生成を用いた冗長除去には、非冗長回路とその回路のテスト集合が同時に得られるという利点があるが、得られるテスト集合の大きさは必ずしも小さなものでない。本論文では、冗長除去とテストパターンの圧縮を同時に実行することにより、効率的に非冗長回路と極小テスト集合を得る手法を提案する。小さなテスト集合を生成する手法として、得られたテスト集合からすべての不必要なテストベクトルを取り除き、極小テスト集合を得る手法が提案されている。冗長除去とこのテスト集合の圧縮においては、テストベクトルの生成や故障シミュレーションに基づくテスト生成の繰り返しなど、同様の処理が異なる目的で用いられている。本手法では、これらの目的を同時に達成するように処理を実行することにより、効率的に非冗長回路と極小テスト集合を得ることができる。ベンチマーク回路に対する実験では、本手法により、効率よく非冗長回路の生成と極小テスト集合の両方を得られることを示す。また、冗長を多く含む回路では、冗長除去により最小限必要なテストベクトル数が少なくできることを示す。

Concurrent Generation of Irredundant Combinational Circuits and Minimal Test Sets

SEIJI KAJIHARA [†] and KOZO KINOSHITA [†]

Since a redundant element of combinational circuits is identified as an undetectable single stuck-at fault, some logic optimization methods using test generation have been proposed. These methods have an advantage that an irredundant circuit and its test set are obtained concurrently, but the size of the test set is not necessarily small. In this paper, we propose a method that performs redundancy removal and test compaction efficiently to derive an irredundant circuit and a minimal test set for a combinational circuit. In order to generate a small test set, there is a method of removing all unnecessary test vectors from an obtained test set. Both redundancy removal and test compaction have the same processes such as generation of test vectors and repetition of fault simulation, but the purposes of them are different each other. In our method, these processes are done to accomplish the different purposes, thus an irredundant circuit and a minimal test set can be derived concurrently. Experimental results for benchmark circuits show that our method could produce an irredundant circuit and a minimal test set efficiently, and that the minimum number of test vectors for circuits with many redundancies can be reduced by redundancy removal.

1. ま え が き

組合せ回路のテスト生成の研究は、検出可能な縮退故障に対するテストベクトルを高速に生成することだけでなく、回路に含まれる冗長部分を識別することにも大きな成果を上げている^{1)~4)}。冗長判定の高速化は、テスト生成による冗長除去を回路簡単化の一手法として現実的なものとしており、それゆえ幾つかの手法が提案されている^{5)~7)}。テスト生成を用いた冗長除去は、非冗長回路とその回路のテスト集合が同時に得られる

という利点をもっている。しかしながら、一般に一つの冗長部分を除去した際に、他の冗長部分が非冗長になったり、それまでは非冗長だった部分が冗長になることがあるため、テスト生成処理の繰り返しが必要となる。2.1節で述べるように、冗長除去の高速化手法も提案されているが、全体的な処理時間は決して小さくなく、得られたテスト集合の大きさも必ずしも小さなものでない。

小さなテスト集合を求めることは、製造された回路のテストに要する時間を削減するため、また、LSI テスターに要求される記憶容量を少なくするため、重要な課題となっている。テスト集合の圧縮手法については2.2節で述べるが、一つのテストベクトルでより多く

[†] 大阪大学大学院工学研究科応用物理学専攻
Department of Applied Physics, Faculty of Engineering, Osaka University

の故障を検出するための動的圧縮^{12),13)}や静的圧縮^{8),9)}などが提案されている。また、逆順故障シミュレーション¹⁰⁾や二重検出法¹¹⁾のように、得られたテスト集合に含まれる不必要なテストベクトルを取り除くことにより、テスト集合を小さくする手法も提案されている。これらの圧縮手法では、故障シミュレーションの繰り返しなどで付加的な計算が必要となる。

本論文では、組合せ回路の単一縮退故障に対する冗長除去と極小テスト集合の生成を同時に行う手法を提案する。非冗長回路とその回路の小さなテスト集合は冗長除去とテスト圧縮の手順を逐次的に実行しても得ることができるが、冗長除去により得られたテスト集合が無駄になるため、時間的に効率が悪いものとなる。本手法は、文献6)の手順に基づいて冗長除去を行うが、検出可能な故障に対しては、動的圧縮の技法を用いて他の未検出故障をできるだけ多く検出するようにテストベクトルを生成する。また、各故障に対するテストベクトルが一度は生成されたなら、それまでに得られているテストベクトルを用いて故障シミュレーションを再度実行し、冗長除去後の回路に対するテストベクトルの有効性を確かめる。更にこの故障シミュレーションでは、二重検出法を用いて不必要となったテストベクトルをテスト集合から除去する操作を導入する。これらの処理の繰り返しにより、非冗長回路と極小テスト集合を効率よく、同時に得ることができる。

ベンチマーク回路に対する実験では、本手法は回路の冗長除去または極小テスト集合の生成のどちらか一方だけを行う場合と同程度の計算時間で、その両方の結果を得ることができることを示す。また、回路内の冗長の有無によるテスト集合の大きさの違いについても考察し、実験により冗長を多く含む回路では、テストベクトルが多く必要となる場合があることを示す。

2章では、本手法で用いる冗長除去と極小テスト集合の生成の手順をそれぞれ述べる。3章では、冗長除去と極小テスト集合の生成を同時に行う手法を提案する。4章では、本手法をベンチマーク回路に適用した結果を示し、5章でまとめを述べる。

2. 基本アルゴリズム

2.1 冗長除去

組合せ回路の冗長部分は、検出不可能な縮退故障としてテスト生成により識別でき、それらは回路の出力関数を変えずに取り除くことができる。しかし、一つの冗長部分(一つの検出不能な故障から冗長と判断される回路部分)を除去した際に、他の冗長部分が非冗長になることがあるため、先にすべての冗長部分

の識別のみを行い後でまとめて除去した場合、元と同じ出力関数は必ずしも保証されない^{5),6)}。そこで、一つの冗長部分の識別と除去を行った後、他の箇所について調べることが求められる^{5),6)}。また、ある冗長部分の除去によりそれまでは非冗長だった部分が冗長になることや、前に生成したテストベクトルが無効になり、新たなテストベクトルの生成が必要になる場合がある。そこで、冗長部分を除去した後は、テストベクトルがすでに生成されている故障についても、テスト生成処理を繰り返すことが必要となる^{5),6)}。そのため冗長部分が多く含まれる回路では、非常に計算時間が大きくなるという欠点を持つ。

文献6)では、除去しても他の冗長性を変えない冗長部分もあることに着目して、冗長除去の処理を効率化する手法が述べられている。ここでは、検出不能故障をクラス化することにより、冗長な信号線を信号値縮退信号線、ゲート入力縮退信号線、伝搬不能信号線の3種類に分類している。信号値縮退信号線とは、どのような入力ベクトルに対しても同じ信号値をとる信号線のことである。また、ゲート入力縮退信号線とは、ゲートの入力信号線の縮退故障がそのゲートの出力に伝搬しない場合、その入力信号線のことをいう。伝搬不能信号線は、一般の検出不可能な故障を持つ信号線のことである。信号値縮退信号線と一部のゲート入力縮退信号線については、同時に複数の冗長部分を除去できることが示されており、処理の高速化が可能になる。

また、テスト生成処理の繰り返し回数を削減するため、冗長部分を見つけるとすぐに除去を行い、各故障に対して少なくとも一度はテストベクトルが生成されるまで、繰り返し処理を行わない手順も導入している。図1にそのフローチャートを示す。まず、同時除去が

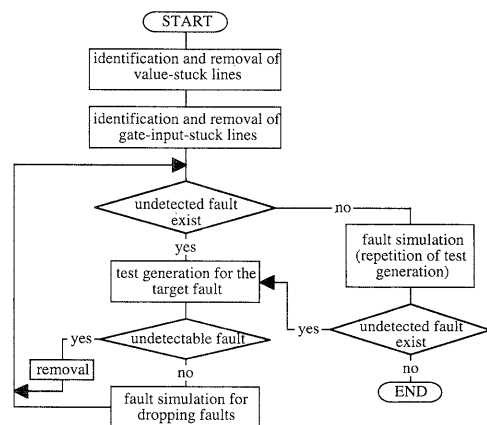


図1 冗長除去のアルゴリズム

Fig. 1 Algorithm of redundancy removal.

可能な信号値縮退信号線を，次にゲート入力縮退信号線を除去する．その後，伝搬不能故障を見つけるためのテスト生成を行う．このテスト生成により，冗長部分を見つけるとすぐに除去し，そうでなければ，得られたテストベクトルに対して故障シミュレーションを行う．生成されたテストベクトルの多くは冗長を除去した後の回路にも有効であるので，保存して，テスト生成の繰り返し処理の最初で故障シミュレーションに用いている．

これらの操作から，結果として非冗長な回路とその回路のテスト集合が得られる．

2.2 テスト圧縮¹¹⁾

小さなテスト集合を得るための手法には，一つのテストベクトルができるだけ多くの故障を検出するようにテスト生成を行う手法^{11)~14)}と，一度生成されたテスト集合を圧縮する手法^{8)~11)}とがある．前者には，ある対象故障（一次故障）に対して生成したテストベクトルの入力の“don't care”に，他の故障（二次故障）を検出するように値を割り当てる動的圧縮法¹²⁾や，動的圧縮法において，一次故障に対するテストベクトルの不必要な入力値を“don't care”に戻す極大圧縮法¹³⁾がある．また，テスト生成における後方追跡操作でゲート入力線の選択順序を動的に変更する巡回後方追跡法¹³⁾も，有効なテスト圧縮手法である．

一方，一度生成されたテスト集合を圧縮する手法として，逆順故障シミュレーション¹⁰⁾，二重検出法¹¹⁾がある．これらは，テストベクトルが生成された順序と異なる順序で故障シミュレーションを行うことにより，テスト集合に含まれる不必要なテストベクトルを取り除く手法である．逆順故障シミュレーションでは，幾つかの不必要なテストベクトルが残る可能性があるが，二重検出法ではすべての不必要なテストベクトルを識別できるため，得られるテスト集合は常に極小に

なる．

これらのテスト圧縮技法を用いてテスト集合を生成する場合の処理の流れを図2にフローチャートで示す．一つのテストベクトルの生成には，極大圧縮法，巡回後方追跡法を用いる．すべての検出可能故障に対してテストベクトルが生成された後，二重検出法により不必要なテストベクトルを識別し，除去する．不必要なテストベクトルがなくなるまで二重検出法を繰り返し適用することにより，得られるテスト集合は常に極小となる．

3. 冗長除去とテスト圧縮の統合

冗長除去におけるテスト生成処理の繰り返しの間には，幾つかのテストベクトルが追加される場合が多い．また，冗長を除去する前の回路に対して必要であったテストベクトルが不必要となっても，それらはテスト集合に取り残されている．このため，得られたテスト集合の大きさは必ずしも小さなものでない．本章では，冗長除去の際に得られるテスト集合が小さくなるように，非冗長回路と極小テスト集合を同時に生成する手法を提案する．

前章で述べた冗長除去とテスト圧縮の手順には，検出可能故障に対するテストベクトルの生成や一度得られたテスト集合に対する故障シミュレーションの繰り返しなど同じ内容の処理が存在する．これらの処理の役割は冗長除去とテスト圧縮の場合で異なっているが，一つにまとめることで，効率良く非冗長回路とその回路の極小テスト集合を生成することができる．図3に本手法のフローチャートを示すが，これは，図1と

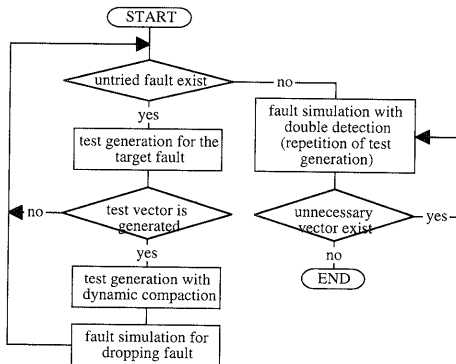


図2 テスト圧縮のアルゴリズム
Fig. 2 Algorithm of test compaction.

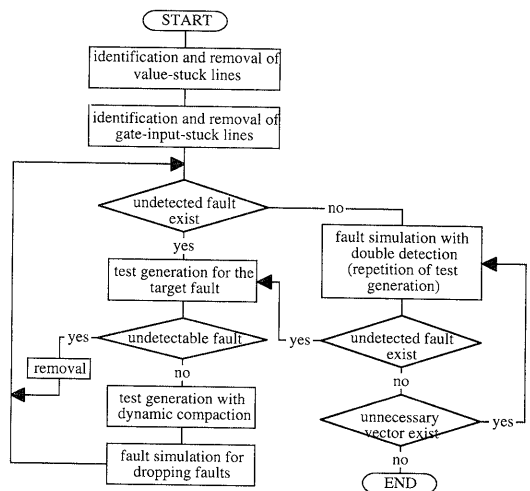


図3 提案手法のアルゴリズム
Fig. 3 Algorithm of the proposed method.

図2にそれぞれ示した冗長除去とテスト圧縮の手順を統合したものとなる。

まず、検出不能故障のクラス化に基づいて同時除去が可能な冗長部分を識別、除去する。そして、各対象故障に対するテストベクトルの生成は、冗長部分の識別に用いるが、故障が検出可能な場合、極大圧縮法と巡回後方追跡のテスト圧縮技法を用いて、できるだけ多くの未検出故障を検出するようにテストベクトルを生成する。また、検出不能故障である場合は、相当する冗長部分を除去する。各故障のテストベクトルが一度は生成されるまで、新たな故障のテスト生成か冗長除去を続ける。その後、それまでに生成されているテストベクトルを用いて故障シミュレーションを行い、冗長除去後の回路に対して、前に生成したテストベクトルの有効性を判断する。この故障シミュレーションでは、二重検出法を用いて不必要となったテストベクトルをテスト集合から除去する操作も導入する。故障シミュレーション後に未検出故障が存在すれば、それらは新しく冗長になっているか、前に生成したテストベクトルで検出できなくなった故障であるので、テスト生成による冗長除去の処理をやり直すことになる。このようにして、最終的には、非冗長な回路とその回路に対する極小テスト集合を得ることができる。

4. 実験結果

4.1 同時生成の効果

本手法をC言語を用いてプログラム化し、ISCAS'85のベンチマーク回路¹³⁾とフルスキャン回路を仮定したISCAS'89のベンチマーク回路¹⁰⁾に対して実験した。使用した計算機は富士通S-4/LCである。本手法に用いたテスト生成アルゴリズムは、SOCRATES³⁾を基本とするもので、また、故障並列型の故障シミュレータを用いた。表1は、各回路に対するテストベクトル数、故障検出率、計算時間を示す。故障検出率は、最終的に得られた回路に対して、

$$\text{coverage}(\%) = \frac{\text{the number of detected faults}}{\text{the number of target faults}} \times 100$$

の式により算出した。計算時間の単位は秒である。本手法の有効性を示すため、冗長除去やテスト圧縮を行わない場合の結果も表1に示す。“underlying test generator”に示される欄は、冗長除去とテスト圧縮のどちらも行わない場合、“redundancy removal”に示される欄は、冗長除去のみ行った場合、“test compaction”に示される欄は、テスト圧縮のみ行った場合、“proposed method”に示される欄は、冗長除去とテス

ト圧縮のどちらも行った場合の結果である。

テストベクトル数については、回路規模が大きくなるほどテスト圧縮の効果が大きくなっている。テストベクトル数と冗長の関係については4.2節で詳しく述べるが、c7552より小さな回路、すなわちゲート数3,000以下の規模の回路では、冗長部分の有無は本手法で生成したテストベクトル数にあまり大きな影響を与えていないように思われる。検出率については、いずれの場合も、検出可能な故障に対してはすべてテストベクトルを生成している。冗長除去やテスト圧縮には、通常のテスト生成に付加的な時間が必要となるが、その両方を同時に行ったとしても、冗長除去のみ、またはテスト圧縮のみを行うのと同程度の時間でよいことが分かる。また、c2670, c7552, c9234のように、本手法が冗長除去のみまたはテスト圧縮のみの処理より高速になっている場合もある。その理由として、冗長除去とテスト圧縮の処理がそれぞれテスト生成を高速にする効果を持っていることが挙げられる。ある冗長部分の除去は、テストベクトルの生成や他の冗長の識別を容易にする効果がある。一方のテスト圧縮では、一つのテストベクトルが多くの故障を検出するよう生成されるため、テストベクトルの生成にかかる検出可能な故障が、他の故障に対して生成したテストベクトルで検出されるようになったことが考えられる。

以上のことから、本手法は冗長除去とテスト圧縮に関して非常に有効であると考えられる。

4.2 非冗長回路のテストサイズ

冗長を含む回路と非冗長な回路に必要な最小テストベクトル数を評価するために、それぞれの回路におけるテストベクトル数の下限と実際にできるだけ小さなテスト集合を生成する実験を行った。結果を表2に示す。テストパターン数の下限は、独立故障集合の大きさから求めることができる^{11),15)~17)}。また、テストベクトルの生成では、一度極小テスト集合を生成した後、そのテスト集合に含まれる二つのテストベクトルを一つのテストベクトルに置き換えることにより、極小値をより小さくする手法を用いた¹⁷⁾。ただし元の回路に対しては、これまでに得られている最もよい数値を示した^{11),15)~17)}。

表2は、元の回路に含まれる検出不能故障の数により、上段から順に並べている。同じ関数を実現する組合せ回路であっても、100個以上の検出不能故障を含むような冗長回路では、より多くのテストパターン数が必要となることがわかる。例えば、s13207の元の回路のすべての検出可能な故障の検出には、少なくとも

表1 ベンチマーク回路に対する実験結果
Table 1 Experimental results for benchmark circuits.

Circuit		Underlying test generator	Redundancy removal	Test compaction	Proposed method
c 432	Vectors	86	60	38	38
	Coverage(%)	99.2	100	99.2	100
	Time(sec)	2.3	2.4	5.0	4.9
c 499	Vectors	94	89	52	52
	Coverage(%)	98.9	100	98.9	100
	Time(sec)	3.3	4.8	3.7	5.2
c 880	Vectors	78	70	27	30
	Coverage(%)	100	100	100	100
	Time(sec)	2.8	3.0	6.8	6.8
c 1355	Vectors	114	118	84	85
	Coverage(%)	99.5	100	99.5	100
	Time(sec)	12	21	14	22
c 1908	Vectors	142	150	110	106
	Coverage(%)	99.5	100	99.5	100
	Time(sec)	9.9	21	32	33
c 2670	Vectors	164	144	59	55
	Coverage(%)	95.7	100	95.7	100
	Time(sec)	41	33	100	72
c 3540	Vectors	196	211	116	114
	Coverage(%)	96.0	100	96.0	100
	Time(sec)	30	109	156	187
c 5315	Vectors	179	195	60	61
	Coverage(%)	98.9	100	98.9	100
	Time(sec)	33	147	221	331
c 6288	Vectors	47	33	21	22
	Coverage(%)	99.6	100	99.6	100
	Time(sec)	247	17	56	75
c 7552	Vectors	280	262	88	83
	Coverage(%)	98.1	100	98.1	100
	Time(sec)	993	1617	595	1108
s 5378	Vectors	317	335	112	109
	Coverage(%)	99.1	100	99.1	100
	Time(sec)	42	92	182	210
s 9234	Vectors	483	449	138	128
	Coverage(%)	93.5	100	93.5	100
	Time(sec)	4449	2050	4493	2440
s 13207	Vectors	563	615	239	178
	Coverage(%)	98.5	100	98.5	100
	Time(sec)	324	5570	1576	6239
s 15850	Vectors	531	520	115	108
	Coverage(%)	96.7	100	96.7	100
	Time(sec)	352	3363	1658	4446

233個のテストベクトルが必要であったが、非冗長回路にすることにより165個のテストベクトルで検出できるようになった。

具体的にどのような冗長を除去した場合にテストベ

クトルが少なくできるかの考察については、今後の課題であるが、この結果から、回路を非冗長にすることはテストパターン数を少なくすることにも貢献することが分かる。

表 2 非冗長回路のテストサイズ
Table 2 Test size of irredundant circuits.

Circuit	Original circuit			Irredundant circuit	
	Redundant faults	Lower bound	Generated tests	Lower bound	Generated tests
c 432	4	27	27	20	28
c 499	8	52	52	50	52
c 1355	8	84	84	82	84
c 1908	9	99	106	90	102
c 6288	34	6	14	6	14
s 5378	40	92	103	79	102
c 5315	59	37	44	24	45
c 2670	117	42	45	37	41
c 7552	131	52	80	34	55
c 3540	137	80	89	75	89
s 13207	151	233	235	160	165
s 15850	389	90	95	83	89
s 9234	452	90	107	69	94

5. む す び

本論文では組合せ回路の非冗長回路と極小テスト集合を生成する手法を述べた。冗長除去とテスト圧縮の双方に必要な処理を共有することで、処理の効率化を実現した。また、実験から、本手法は、非冗長回路の生成か、または、極小テスト集合の生成のどちらか一方だけを行う場合と同程度の計算時間で、その両方の結果が得られることを示した。また、同じ関数を実現する論理回路であっても、冗長が多く含まれる回路では、より多くのテストベクトルが必要となることを示した。

参 考 文 献

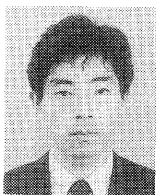
- 1) Goel, P.: An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits, *IEEE Trans. Comput.*, Vol. C-30, No. 3, pp. 215-222 (1981).
- 2) Fujiwara, H. and Shimono, T.: On the Acceleration of Test Generation Algorithms, *IEEE Trans. Comput.*, Vol. C-32, No. 12, pp. 1137-1144 (1983).
- 3) Schulz, M. H. and Auth, E.: Advanced Automatic Test Pattern Generation and Redundancy Identification Techniques, *FTCS-18*, pp. 30-35 (1988).
- 4) Waicukauski, J. A., Shupe, P. A., Giramma, D. J. and Matin, A.: ATPG for Ultra-Large Structured Designs, *1990 ITC*, pp. 44-51 (1990).
- 5) Bryan, D., Brglez, F. and Lisanke, R.: Redundancy Identification and Removal, *MCNC Workshop on Logic Synthesis* (1989).
- 6) Kajihara, S., Shiba, H. and Kinoshita, K.: Removal of Redundancy in Logic Circuits under

Classification of Undetectable Faults, *FTCS-22*, pp. 263-270 (1992).

- 7) Cheng, K. T.: An ATPG-Based Approach to Sequential Logic Optimization, *ICCAD-91*, pp. 372-375 (1991).
- 8) Chang, J. S. and Lin, C. S.: Test Set Compaction for Combinational Circuits, *ATS'92*, pp. 20-25 (1992).
- 9) Reddy, L. N., Pomeranz, I. and Reddy, S. M.: ROTCO: A Reverse Order Test Compaction Technique, *1991 IEEE EURO-ASIC Conference*, pp. 189-194 (1991).
- 10) Schulz, M. H., Trischler, E. and Sarfert, T.: SOCRATES: A Highly Efficient Automatic Test Pattern Generation System, *IEEE Trans. CAD.*, pp. 126-137 (1988).
- 11) 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, *30th DAC*, pp. 102-106 (1993).
- 12) Goel, P. and Rosales, B. C.: Test Generation and Dynamic Compaction of Tests, *Digest of Papers 1979 Test Conf.*, pp. 189-192 (1979).
- 13) Pomeranz, I., Reddy, L. N. and Reddy, S. M.: COMPACTEST: A Method to Generate Compact Test Sets for Combinational Circuits, *1991 ITC*, pp. 194-203 (1991).
- 14) Higuchi, H., Ishiura, N. and Yajima, S.: Compaction of Test Sets Based on Symbolic Fault Simulation, *SASIMI'92*, pp. 253 - 262 (1992).
- 15) Matsunaga, Y.: MINT—An Exact Algorithm for Finding Minimum Test Sets—, *IEICE Trans. Fundamentals*, Vol. E76-A, No. 10, pp. 1652-1658 (1993).
- 16) Chang, J. -S. and Lin, C. -S.: Test Set Compaction for Combinational Circuits, *First Asian Test Symposium (ATS'92)*, pp. 20-25 (1992).
- 17) Kajihara, S., Pomeranz, I., Kinoshita, K. and Reddy, S. M.: On Compacting Test Sets by Addition and Removal of Test Vectors, *12th IEEE VLSI Test Symp.*, pp. 202-207 (1994).
- 18) Brglez, F. and Fujiwara, H.: A Neutral Netlist of 10 Combinational Benchmark Circuits and A Target Translator in FORTRAN, Special Session on ATPG and Fault Simulator, *ISCAS'85* (1985).
- 19) Brglez, F., Bryan, D. and Kozminski, K.: Combinational Profiles of Sequential Benchmark Circuits, *ISCAS'89*, pp. 1929-1934 (1989).

(平成 6 年 9 月 2 日受付)

(平成 7 年 3 月 13 日採録)

**梶原 誠司 (正会員)**

1965年生まれ、1987年広島大学総合科学部総合科学科卒業。1989年広島大学大学院工学研究科情報工学専攻博士課程前期修了。1992年大阪大学大学院工学研究科応用物理学専攻博士後期課程修了。博士(工学)。同年大阪大学工学部応用物理学科助手。論理回路のテスト生成、テスト容易化設計などの研究に従事。電子情報通信学会、IEEE各会員。

**樹下 行三 (正会員)**

1959年大阪大学工学部通信工学科卒業。1964年同大学院工学研究科通信工学専攻博士課程修了。工学博士。同年同大学工学部電子工学科助手、1966年同学科助教授、1978年広島大学総合科学部教授、1989年10月より大阪大学工学部教授。論理回路およびメモリのテスト容易化設計、故障診断、テスト生成などの研究に従事。1986～88年情報処理学会設計自動化研究会主査、1987～89年電子情報通信学会フォールトトレラントシステム研究会専門委員長、1988年IEEE・FTCS-18プログラム委員長、1992年IEEE第1回アジアテストシンポジウム実行委員長など。電子情報通信学会会員、IEEEフェロー。