

スキャン極性調節とピンポイントテスト変換によるテスト圧縮

土井 康稔[†] 梶原 誠司[†] 温 暁青[†] Lei Li[‡] Krishnendu Chakrabarty[‡]

[†]九州工業大学 〒820-8502 福岡県飯塚市川津 680-4

[‡]デューク大学 Durham, NC 27708, U.S.A.

E-mail: [†]doi@aries30.cse.kyutech.ac.jp, [†]{kajihara.wen}@cse.kyutech.ac.jp, [‡]{ll.krish}@ee.duke.edu

あらまし 本論文では、ランレングス符号化のテストデータ量削減効率を向上させるテストコンプレッション手法を提案する。提案手法は、スキャン極性調節とピンポイントテストパターン変換で構成される。スキャン極性調節では、フルスキャン回路に対する与えられたテスト集合に対して、テストパターン中のいくつかのスキャンセルの論理値を選択的に反転する。これはスキャンセルの否定出力 \bar{Q} を次のスキャンセルに連結することで実現することができる。ピンポイントテストパターン変換は、テストパターン中の指定した論理値 1 のビットを 0 に反転する。これらの手法では、反転するスキャンセルやビットを決定するために gain-penalty 表を用いる。ベンチマーク回路に対する実験結果では、提案手法によりテストデータ量を 36% に削減でき、スキャンテスト時のスイッチングアクティビティも削減することができた。

キーワード ランレングス符号、テストパターン変換、ゲートッドスキャンチェーン

Test compression for scan circuits using scan polarity adjustment and pinpoint test relaxation

Seiji KAJIHARA[†] Yasumi DOI[†] Lei LI[‡] and Krishnendu CHAKRABARTY[‡]

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

[‡] Duke University Durham, NC 27708, U.S.A.

E-mail: [†]kajihara@cse.kyutech.ac.jp, [†]doi@aries30.cse.kyutech.ac.jp, [‡]{ll.krish}@ee.duke.edu

Abstract This paper presents a test compression method that effectively derives the capability of a run-length based encoding. The method is based on scan polarity adjustment and pinpoint test relaxation. Given a test set for a full scan circuit, scan polarity adjustment selectively flips values of some scan sells in test patterns. It can be realized by changing connections between two scan cells such that the inverted output of a scan cell \bar{Q} is connected to the next scan cell. Pinpoint test relaxation flips some specified 1s in the test patterns to 0s. Both techniques are applied with referring to the gain-penalty table to determine scan cells or bits to be flipped. Experimental results for ISCAS'89 benchmark circuits show that the proposed method could reduce test data volume by 36%, and could reduce switching activities (i.e. test power) during scan testing too.

Keyword run-length codes, test relaxation, gated scan chain

1. はじめに

SoC(System-On-Chip)テストにおいてテストデータ量とテスト電力の増加は大きな問題となっている。特に、テストデータ量は直接的にテストコストに影響するため、テストデータ量を削減する手法については多くの研究がなされている[1-15]。特にテストコンプレッションは、小さなハードウェアをチップ上に組み込むことでテストデータを強力に圧縮する。たとえば、LFSR のリシーディングは効果的なテストコンプレッション手法の一つである[4,5,6]。また、ランレングス符号や統計符号を用いたテストデータ量削減手法もあ

る[7-11]。これらの手法においてチップ上に組み込まれるハードウェアは順序回路であるが、デコンプレッサとして組合せ回路を用いるテストコンプレッション技術も提案されている[12,13]。ソフトウェアベースのデコンプレッションを行うために組み込まれたプロセッサを利用する手法もある[20]。

高度なコンパクション技術を用いて生成されたテストパターンでは、そのすべてのビットに 0 または 1 が割り当てられている。一方、テストコンプレッションでは高い圧縮率を得るためにテストパターン中の未設定値が必要とされる。

テストパターン中の値には、逆の論理値を割り当てても故障検出率に影響を与えないものがあり、そのような入力値はドントケア (X) として扱うことができる。テスト集合中のドントケア判定手法は[17,18]で提案されている。[13,21]ではこのドントケア判定手法を用いたテストデータ削減効率の向上手法が提案されている。しかし、テスト集合中にはXにならないビットもあり、そのようなビットがXを利用するテストコンプレッションの制約になる。

本論文では、FDR (Frequency-Directed Run-length) 符号化のようなランレングス符号化を利用したテストデータ量削減効率向上手法を提案する。本手法は、ピンポイントテスト変換とスキャン極性調節の2つの技術を利用する。スキャン極性調節では、スキャンセルの否定出力 \bar{Q} を次のスキャンセルに連結することで、テストパターン中のスキャンセルの論理値を選択的に反転する。ランレングススペースの符号化手法は、テスト集合中に0がより長く連続する個所が多いとき、より高い圧縮率を得ることができる。従って、スキャンセルにおける0の数が多くなるようにスキャン極性調節を行う。スキャン極性調節の考え方は、テスト電力を削減するスキャンチェーン上へのゲート挿入法[22]として開発されてきたが、本研究のスキャン極性調節では付加ゲートは必要としない。また、論理値を反転するスキャンセルは、gain-penalty表を用いて反転したときの利得の合計を計算することで選択する。

スキャン極性調節の後に、さらに高い圧縮率を得るためにピンポイントテスト変換[21]を行う。ピンポイントテスト変換では、X判定されるビットは、論理値を反転したときの利得が最大となるようにピンポイントに選択される。結果として、与えられたテストパターンはテストコンプレッションに最適なテストパターンに変換される。その上、ランレングス符号化を用いて効果的に圧縮されたテストパターンは、テスト時の消費電力が低いことが知られており、提案手法は低消費電力テスト手法[22,23]の特徴も兼ね備えている。

ISCAS ベンチマーク回路に対する実験では、提案手法によりテストデータ量を平均 36% に削減でき、スキャンテスト時のスイッチングアクティビティも効果的に削減することができた。

以下、本論文は次のように構成される。2章では、提案手法の要素技術を説明し、3章で、提案手法についての詳細を述べる。4章で実験結果を示し、最後に5章で本論文をまとめる。

2. 関連研究

2.1 ランレングスに基づく符号化

ランレングスに基づくテストパターンの符号化手法として、FDR(Frequency Directed Run-length)符号化があ

表1: FDR符号の符号語割当

Group	Run-length	prefix	Tail	Codeword
A ₁	0	0	0	00
	1		1	01
A ₂	2	10	00	1000
	3		01	1001
	4		10	1010
	5		11	1011
A ₃	6	110	000	110000
	7		001	110001
	8		010	110010
	9		011	110011
	10		100	110100
	11		101	110101
	12		110	110110
13	111	110111		

t_1 : 00100

t_2 : 10101

t_3 : 00011

TD: 001001010100011

TE: 100010000101100100

(a) 与えられたテスト集合

(b) テストストリーム

図1: FDR符号化の例

る。この手法は、テストパターン中の連続する0を符号語に変換することによってテストデータ量を削減する。以降本論文では、符号化前のテストパターンと符号化後のテストパターンをそれぞれ T_D , T_E とする。

表1にFDR符号の例を示す。図1(a)のようなテスト集合が与えられたとき、図1(b)のようにすべてのテストベクトルを連結し、TDのような1つのテストストリームとして扱う。その後TEのように符号語に変換される。

ランレングスに基づく符号化は、テストパターン中の0が多く、それぞれの連続する0が長いとき、その効果が上がる。従って、0が長く連続する個所が多くなるようにテストパターンを変換することによってテストデータ量をより小さくすることができる。

2.2 ピンポイントテストパターン変換

ランレングスに基づく符号化によってテストデータ量を削減するためには、テスト集合中により多くの0が必要である。ピンポイントテストパターン変換手法は[21]で提案されている。この手法では、与えられたテストパターンを符号化する前に、テスト集合の故障検出率を下げることなく、ドントケア判定手法[18]を用いてテストパターン中のいくつかのビットの論理値を反転する。このとき論理値を反転するビットは、2つの連続する0が1つの長く連続する0に連結されるように選択される。例えば、図1(a)のテスト集合が圧縮されると仮定する。元のテストストリーム TD_0 から得られるXを含むテストストリームとして、図2(a)の TD_1 と TD_2 のようなXのビット位置が異なる2種類のテストストリームを考える。 TD_1 のXに0を割り当てた後に表1を用いてFDR符号化すると図2(b)の TE_1

TD_0 : 001001010100011
 TD_1 : 00x0010101000x1
 TD_2 : 0010010x01000x1

(a) テストストリーム

TE_0 : 100010000101100100
 TE_1 : 101101011010
 TE_2 : 1000100010011010

(b) Xに0を割り当てた後FDR符号化した例

図2: テストパターン変換の例

のようになる。 TD_0 から TD_1 に変換することによる利得は、 $|TE_0| - |TE_1| = 6$ ビットである。ここで $|TE_i|$ は、 TE_i のビット数を示す。一方 TD_2 の場合、利得は $|TE_0| - |TE_2| = 2$ ビットとなる。 TD_1 と TD_2 の X の数は等しいが、利得は異なることが分かる。連結による利得は、連結するビットの前後に連続する0の長さで決まる。 TD_1 の場合、最初の X に0を割り当てることによって、0の長さが2の2つの部分が連結され、0の長さは5となる。表1の符号語より、0の長さが2のときの符号語と0の長さが5のときの符号語は、どちらも4ビットなので、利得は4ビットとなる。一方 TD_2 の場合、最初の X に0を割り当てると、0の長さが1の2つの部分が連結される。しかし、0の長さが1のときの符号語は2ビットであり、0の長さが3のときの符号語は4ビットなので、利得は0となる。

このように、表1の符号の割当てに基づいて、変換する1の前後に連続する0の長さから利得をそれぞれ計算することができる。そこで、この手法では2つの0が連続する部分を連結して符号化することによって削減できるビット数を示す2次元の表をあらかじめ用意している。これを利得表と呼ぶ。FDR符号化の利得表を表2に示す。縦軸と横軸の数字は、それぞれ連結するビットの前後に連続する0の長さを示している。ドントケア判定を行うとき、利得表はテストパターン変

表2: FDR符号に対する利得表

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	0	2	2	2	0	2	2	2	2	2	2	2	0	2	2
1	0	0	2	2	0	0	2	2	2	2	2	2	0	0	2	2
2	2	2	4	2	2	2	4	4	4	4	4	2	2	2	4	4
3	2	2	2	2	2	2	4	4	4	4	2	2	2	2	4	4
4	2	0	2	2	2	2	4	4	4	2	2	2	2	2	4	4
5	0	0	2	2	2	2	4	4	2	2	2	2	2	2	4	4
6	2	2	4	4	4	4	6	4	4	4	4	4	4	4	6	6
7	2	2	4	4	4	4	4	4	4	4	4	4	4	4	6	6
8	2	2	4	4	4	2	4	4	4	4	4	4	4	4	6	6
9	2	2	4	4	2	2	4	4	4	4	4	4	4	4	6	6
10	2	2	4	2	2	2	4	4	4	4	4	4	4	4	6	6
11	2	2	2	2	2	2	4	4	4	4	4	4	4	4	6	6
12	2	0	2	2	2	2	4	4	4	4	4	4	4	4	6	6
13	0	0	2	2	2	2	4	4	4	4	4	4	4	4	6	6
14	2	2	4	4	4	4	6	6	6	6	6	6	6	6	8	6
15	2	2	4	4	4	4	6	6	6	6	6	6	6	6	6	6

換することでより多くの利得が得られる論理値1のビットをピンポイントに指定するために使われる。このようにピンポイントテストパターン変換を行うことで、与えられたテスト集合を最大限に圧縮することができる。

2.3 ゲーテッドスキャンチェーン[22]

スキャンセル間にインバーターや XOR のような論理ゲートを挿入するゲーテッドスキャンチェーンはスキャンイン動作時の遷移数を削減する目的で提案されている[22]。各スキャンセルの信号値変化回数を少なくすることでテスト時の消費電力を削減できる。図3にその例を示す。ゲート挿入をしていない普通のスキャンチェーンにテストベクトル“00110”をスキャンインすると、多くのスキャンセルで遷移が起こる。しかし、NOTゲート挿入したスキャンチェーンに“00000”をスキャンインすると、スキャンセルでは遷移は起こらず、テスト対象回路には実際のテストベクトル“00110”が設定される。この手法により、スキャンチェーンでの消費電力や、シフト時に論理回路内で生じる信号値変化回数を削減できる。

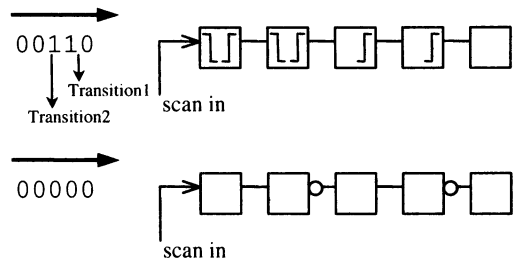


図3: テストベクトルの遷移がスキャンセルに及ぼす影響

3. 提案手法

3.1 スキャン極性調節

本研究では、テスト圧縮の効果を引き出すようにテストパターン変換を行う。図4のように、3つのテストベクトルからなり、スキャンセル数が5のテスト集合が与えられたとする。FDR符号化によってテストデータをより圧縮するためには、テストパターン中の0が長く連続する箇所をより多くする必要がある。テスト集合の3つ目の入力には0よりも1の数が多いので、高い圧縮率を得るために3つ目の入力の論理値を反転する。それに応じて、元のテスト集合と同じテストデータがテスト対象回路に印加されるように、スキャンチェーン上をシフトされる値の極性を変える必要がある。本手法では、図4のようにスキャンセルの出力 Q の代わりに否定出力 \bar{Q} を使うことで実現する。この手法をスキャン極性調節と呼ぶ。スキャン極性調節は論理ゲートの挿入を必要としないので、ハードウェアオーバーヘッドや遅延は通常のスキャン設計とほとんど同じである。

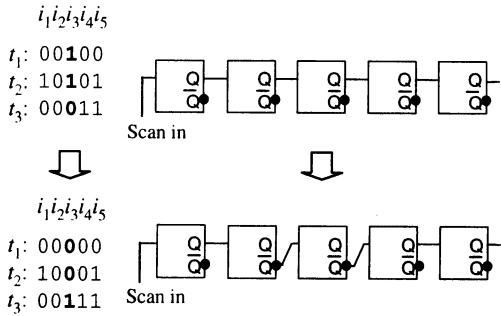


図4: スキャン極性調節の例

Original: TD: 001001010100011
TE: 100010000101100100

After flipping the 3rd input: TD': 000001000100111
TE': 1011100110000000

図5: FDR符号化の例

図5では図4のテスト集合を圧縮したテストストリームを示している。この例では、スキャン極性調節によって得られる利得は2ビットとなる。

3.2 gain-penalty 表による入力反転

スキャン極性調節は選択したスキャンセルのすべての論理値を反転するので、元のテストパターン中に含まれるいくつかの0を1に変えてしまう可能性がある。そのような場合、テスト圧縮において損失(テストデータ量の増加)となる。本研究では、どのスキャンセルを反転するかを決定するために、gain-penalty表を用いる。gain-penalty表によって、テスト集合中の各ビットを反転することによって削減できるビット数が分かる。その後、各入力反転による利得の合計を算出する。図4のテスト集合が与えられたとき、各スキャンセルの利得の合計は表3(a)ようになる。この表では、論理値を1から0に反転したときの利得は正、論理値0から1に反転したときの利得(つまり損失)は負で表している。3つ目と5つ目の入力の利得の合計は正なので、より高い圧縮率を得るためにこれらの入力を反転する。入力反転後、表3(b)のように各入力の利得の合計を再計算し、すべての入力の利得の合計が0以下になるまでこの処理を続ける。

3.3 処理手順

与えられたテスト集合に対して、2つのテスト圧縮手法を用意した。手法1の概略を以下に示す。

表3: 利得の合計

(a) Original tests TD						(b) Flipped tests TD'					
	i_1	i_2	i_3	i_4	i_5		i_1	i_2	i_3	i_4	i_5
t_1	0	0	4	0	0	t_1	-2	-2	-2	-2	2
t_2	2	-2	0	-2	2	t_2	2	0	0	-2	-2
t_3	-2	0	-2	2	2	t_3	0	0	2	0	-2
Sum	0	-2	2	0	4	Sum	0	-2	0	-4	-2

<手法1>

- Step 1: 与えられたテスト集合のすべてのスキャンセルの利得の合計を算出する。
- Step 2: もしすべてのスキャンセルの利得の合計が0以下ならばStep4の処理を行う。
- Step 3: 利得の合計が0より大きいスキャンセルの論理値をすべて反転し、Step1の処理を行う。
- Step 4: スキャン極性調節によって変換されたテスト集合に対してピンポイントテストパターン変換を行う。

手法1では損失をもたらす入力を反転するかもしれない。なぜなら入力を反転後、その近くにある入力の利得は異なるかもしれないからである。損失をもたらす入力を反転することを避けるために、以下のような手法2を提案する。

<手法2>

- Step 1: 利得の合計が0より大きいスキャンセルを見つめる。
- Step 2: もしそのようなスキャンセルが見つければ、スキャンセルの論理値を反転し、Step1の処理を行う。そうでなければStep3の処理を行う。
- Step 3: スキャン極性調節によって変換されたテスト集合に対してピンポイントテストパターン変換を行う。

4. 実験結果

提案手法をPC(Pentium IV 1.80GHz, 768MB)上にC言語で実装し、ISCAS'89フルスキャンベンチマーク回路に対して実験を行った。最初のテスト集合として、コンパクトテスト集合[2]を用い、FDR符号化に対する利得表として、510×510の行列を用いた。表4に手法1と手法2の実験結果を示す。表中の"#tests"と"#inputs"は、元のテスト集合のテストベクトル数とテストベクトルのビット数、"TD"は元のテスト集合のテストデータ量、"TE"は圧縮されたテストデータ量、"time"は実行時間を示している。テストデータ量と実行時間において、手法2よりも手法1の方が良い結果が得られた。

表5に、ピンポイントテストパターン変換前に行ったスキャン極性調節の実験結果の詳細を示す。"#scan cells"はスキャンセル数、"#flipping"は反転したスキャンセル数を示している。提案手法によって与えられたテスト集合中のおよそ50%のスキャンセルを反転した。手法1は手法2よりも多くのスキャンセルを反転しているが、実行時間は手法1の方が早かった。

提案手法のテスト電力削減効果を確認するために、スキャンチェーンでのスイッチングアクティビティを算出した結果を表6に示す。"average sw.-ac"は、各スキャンシフトにおいて値が変化したフリップフロップの割合の平均である。"maximum sw.-ac"は、一回のス

表4: スキャン極性調節とピンポイントテストパターン変換後の圧縮結果

circuit	#tests	#inputs	TD [bits]	TE [bits]		time [sec]	
				Method 1	Method 2	Method 1	Method 2
s5378	100	214	21400	8502	8430	24	25
s9234	111	247	27417	10608	10688	70	90
s13207	235	700	164500	18518	18754	708	950
s15850	97	611	59267	15900	15704	259	262
s35932	12	1763	21156	13880	14414	484	288
s38417	87	1664	144768	57118	57532	566	547
s38584	114	1464	166896	53774	54138	986	1043

表5: スキャン極性調節の実験結果の詳細

circuit	#scan cells	#flipping		time [sec]	
		Method 1	Method 2	Method 1	Method 2
s5378	179	99	91	0.41	0.45
s9234	228	143	138	0.58	0.63
s13207	669	248	234	1.00	1.31
s15850	597	237	234	0.98	1.12
s35932	1728	1053	776	1.55	1.69
s38417	1636	1265	1235	2.26	2.56
s38584	1452	488	473	2.35	2.60

表6: スキャンFFでのスイッチングアクティビティの削減率

circuit	average sw.-ac. [%]			maximum sw.-ac. [%]		
	original	ours	%imp	original	ours	%imp
s5378	35.34	19.83	43.89	47.66	45.33	4.89
s9234	31.08	18.46	40.60	50.20	51.01	-1.61
s13207	36.25	16.35	54.90	54.29	47.71	12.12
s15850	25.46	16.40	35.59	46.15	37.32	19.13
s35932	3.26	9.86	-202.45	87.80	72.26	17.70
s38417	35.20	24.15	31.39	44.23	40.93	7.46
s38584	30.68	17.08	44.33	63.59	57.58	9.45

表7: 実験結果の比較

circuit	ours	pinpoint [21]	EFDR [9]	VIHC [14]	RESPIN++ [15]	XORnet [12]
s5378	<u>8502</u>	10490	11419	11516	17332	N/A
s9234	<u>10608</u>	15454	21250	17736	17198	N/A
s13207	<u>18518</u>	22840	29992	27737	26004	25344
s15850	<u>15900</u>	18466	24643	30271	32226	22784
s35932	13880	18072	<u>5554</u>	9458	N/A	7128
s38417	<u>57118</u>	64078	64962	74938	89132	89356
s38584	53774	59262	73853	85674	63232	<u>38976</u>

キャンシフトで値が変化したスキャンセルの割合の最大値である。"original"と"ours"は、元のテスト集合と手法1によって変換されたテスト集合での実験結果を示している。また、"%imp"はスイッチングアクティビティの削減率を示している。s35932を除くすべての回路において、平均41%のスイッチングアクティビティを提案手法によって削減することができた。一方、最

大スイッチングアクティビティはそれほど多く削減することはできなかった。

表7に、手法1と他の手法との比較を示す。"pinpoint"はピンポイントテストパターン変換手法[21]の実験結果を示している。提案手法によって、ピンポイントテストパターン変換のみの場合より、テストデータ量を18%削減することができた。次に、EFDR

coding [9], variable-length Huffman coding [14], RESPIN++ [15], XOR network [12]の結果を示した。多くの回路において、提案手法によって最も小さいテストデータを得ることができた。

5. まとめ

本論文では、ランレングス符号化のテストデータ量削減効率を向上させるテストコンプレッション手法を提案した。与えられたテスト集合を符号化する前に、スキャン極性調節手法を用いてテスト集合の入力の値を反転した。値を反転するスキャンセルは、gain-penalty表を用いて選択された。その後、得られたテストパターンに対してピンポイントテストパターン変換を行った。ISCAS'89ベンチマーク回路に対する実験結果では、提案手法によりテストデータ量だけでなくテスト電力も削減することができた。

文 献

- [1] I. Pomeranz, L. N. Reddy and S. M. Reddy, "COMPACTEST: A Method to Generate Compact test Sets for Combinational Circuits," IEEE Trans. CAD, pp. 1040-1049, Jul. 1993.
- [2] S. Kajihara, I. Pomeranz, K. Kinoshita and S. M. Reddy, "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits," IEEE Trans. CAD, pp.1496-1504, Dec. 1995.
- [3] I. Hamzaoglu and J. H. Patel, "Test Set Compaction Algorithms for Combinational Circuits," Proc. Int'l Test Conf, pp. 283-289, Oct. 1998
- [4] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman and B. Courtois, "Built-in Test for Circuits with Scan Based on Reseeding of Multiple-polynomial Linear Feedback Shift Registers," IEEE Trans. Comp., pp. 223-233, Feb. 1995.
- [5] S. Hellebrand, H.-G. Liang, H. -J. Wunderlich, "A mixed-mode BIST scheme based on reseeding of folding counters," Int'l Test Conf., pp.778-784, Oct. 2000.
- [6] C. V. Krishna, A. Jas, and N. A. Toubia, "Test Vector Encoding Using Partial LFSR Reseeding," International Test Conf., pp. 885-893, Oct. 2001.
- [7] A. Chandra and K. Chakrabarty, "System-on-a-chip test data compression and decompression architectures based on Golomb codes", IEEE Transactions on CAD/ICAS, vol. 20, pp. 355-368, March 2001.
- [8] A. Chandra and K. Chakrabarty, "Test data compression and test resource partitioning for system-on-a-chip using frequency-directed run-length (FDR) codes", IEEE Transactions on Computers, vol. 52, pp. 1076-1088, August 2003.
- [9] A. El-Maleh, R. Al-Abaji, "Extended Frequency-directed Runlength Codes with Improved Application to System-on-a-Chip Test Data Compression," Int'l Conf. Electronics, Circuits and Systems, pp. 449-452, 2002.
- [10] A. Jas, J. Ghosh-Dastidar, and N. A. Tuba, "Scan Vector Compression/Decompression Using Statistical Coding," VLSI Test Symposium, pp. 114-120, April 1999.
- [11] H. Ichihara, K. Kinoshita, I. Pomeranz, S. M. Reddy, "Test Transformation to Improve Compaction by Statistical Encoding," International Conf. on VLSI Design, pp.294-299, Jan. 2000.
- [12] I. Bayraktaroglu, and A. Orailoglu, "Test Volume and Application Time Reduction Through Scan Chain Concealment," Design Automation Conference, pp.151-155, June 2001.
- [13] S. M. Reddy, K. Miyase, S. Kajihara and I. Pomeranz, "On Test Data Volume Reduction for Multiple Scan Chain Designs," VLSI Test Symposium, pp. 103-108, April 2002.
- [14] P. T. Gonciari, B. Al-Hashimi, N. Nicolici, "Improving compression ratio, area overhead, and test application time for system-on-a-chip test data compression/decompression," Design Automation and Test in Europe Conf., pp. 604-611, 2002.
- [15] L. Schafer, R. Dorsch, H. -J. Wunderlich, "RESPIN++ -Deterministic Embedded Test," European Test Workshop, pp.37-44, May 2002.
- [16] M. Schulz, E. Trischler, and T. Sarfert, "SOCRATES: A Highly Efficient Automatic Test Pattern Generation System," IEEE Trans. on CAD., pp. 126-137, Jan. 1988.
- [17] S. Kajihara, and K. Miyase, "On Identifying Don't Care Inputs of Test Patterns for Combinational Circuits," Proc. Int'l Conf. on Computer Aided Design, pp. 364-369, Nov. 2001.
- [18] K. Miyase, S. Kajihara, I. Pomeranz and S. M. Reddy, "Don't care identification on specific bits of test patterns," 2002 International Conference on Computer Design, pp. 194- 199, Sep. 2002.
- [19] P. Goel, and B. C. Rosales, "Test Generation and Dynamic Compaction of Tests," Digest of Papers 1979 Test Conf., pp. 189-192, Oct. 1979.
- [20] A. Jas and N. A. Toubia, "Using an embedded processor for efficient deterministic testing of system-on-a-chip," Int'l Conf. on Computer Design, pp.418-423, 1999.
- [21] S. Kajihara, Y. Doi, L. Li and K. Chakrabarty, "On combining Pinpoint Test Set Relaxation and Run-Length Codes for Reducing Test Data Volume," 2003 International Conference on Computer Design, pp.387-392, Oct. 2003.
- [22] O. Sinanoglu, I. Bayraktaroglu and A. Orailoglu, "Test Power Reduction Through Minimization of Scan Chain Transitions," VLSI Test Symposium, pp. 166-172, 2002.
- [23] S. Kajihara, K. Ishida, K. Miyase, "Test Vector Modification for Power Reduction during Scan Testing," IEEE VLSI Test Symposium, pp. 160-165, April 2002.