Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

情報処理学会　2014年度（平成26年度）

単一縮退故障用テストパターン生成を利用した多重縮退故障用テストパターン生成

藤田 昌宏, 田口 直樹

2015年2月4日 提出

Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

0.1 はじめに

近年、LSI の集積度はムーアの法則に従って、年々指数関数的に増加しており、それに伴い、要求される LSI テストの分量も増大している。単位面積当たりのゲート数が増加しているため、従来は一つの LSI に対して単一の故障のみを想定してテストを行えば実用上の問題はないとされてきたが、複数の様々な故障を想定したテストが必要になってきている。実際の製造において多重故障が確認されたことを報告しており、多重故障の検出は LSI テストにおける課題の１つとなっている。潜在的な単一故障の数は、発生箇所の定数倍のオーダーであるのに対して、多重故障の場合はその数が指数となるが、多くの多重故障は、単一故障の検出に必要なテストパターンにより検出可能であることは経験的に知られている。ただし、多重故障の組み合わせは回路規模に対して指数的に増大するため、多重故障に対して十分なテストパタンを生成することは容易ではない。

従来の単一縮退故障用のテストパターンは、このような故障シミュレーションを用いて行われてきた。しかし、多重縮退故障の場合にはテスト対象とする故障の数が膨大となるため、故障リストの生成が困難であり、一定規模以上の回路においては多重縮退故障用の ATPG は困難であった。しかしながら、最近になり、故障リストを用いない ATPG 手法 [] が提案され、1 万ゲート規模くらいまでの回路に対する多重縮退故障用の完全なテストパターン生成が可能になった。それに伴い、より少数のテストパターンでの多重縮退故障検出への要求も高まってきている。

本研究では、LSI の製造工程において発生しうる多重縮退故障を、より少数で検出するようなテストパターンの生成手法を検討する。本研究では、文献 [] の手法により生成された単一縮退用のテストパターンを、ATPG ツールである ABC[] の入力テストパターンとして用い、多重縮退故障を検出するテストパターンを生成する。すなわち、多重故障用のテストパターンを、単一縮退故障用のテストパターンに追加する方法で求める。また、多重故障の完全な検出のために追加のテストパターンを要した場合のテスト対象の回路及び生成されたテストパターンについて、具体的な回路を用いて検討する。単一故障検出に特化したテストパターンの組では、従来の圧縮手法のもとで生成された場合、どのようにテストパターンの組を生成したとしても、回路構造の特徴に起因する理由によって検出不可能な多重故障が存在する場合がある。そのため、単一縮退故障を最小数で検出するテストパターンの代わりに、各単一縮退故障を複数回検出する（N 回検出）テストパターン生成で得られたテストパターンを基にして多重故障用のテストパターン生成を行なう手法の評価も行なう。

本稿の構成を以下に示す。まず第 0.2 章で、本研究の土台となる充足可能性問題（SAT）及び自動テストパターン生成（ATPG: Automatically Test Pattern Generation) について説明する。次に第 0.3 章では、ATPG ツールである ABC を用いて生成した、単一故障用と多重故障用のテストパターンの評価について説明する。そして、第４ 0.4 章では単一故障では検出できない多重故障が存在する条件について説明する。最後に 0.5 章では結論と今後の課題を述べる.

Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

0.2 SAT を用いた ATPG 手法

0.2.1 SAT

充足可能性問題 (SAT:SATisfiability problem) とは、与えられた論理関数を 1 にするような変数の組み合わせの有無を探索する問題である。近年、SAT の解を求めるアルゴリズムを実装した SAT ソルバの性能が飛躍的に向上し、解ける問題のサイズが増大している。SAT 自体は 1960 年と昔から知られている問題であり、SAT に帰着できる問題は SAT ソルバにより解を求められる。現在では様々な分野での応用例が報告されており、LSI の設計・製造工程においては、特に検証・テストで用いられている。SAT を解く最も基本的な DPLL アルゴリズムは、変数の割り当ての場合分けを順番に探索するアルゴリズムである。

0.2.2 インクリメンタル SAT

ある論理式に対する SAT の解を求めた後に、その論理式に新たな項を追加した論理式に対しての SAT の解を求めたいとする。この時、直前に解いた SAT 問題における情報を、新しい方の SAT を解く際に利用できる。この時、古い方の式で解となりえない変数の組合せは、新しい方の式でも解となりえない。このように、新しい方の式に対して一から問題を解く場合に比較して、探索範囲を狭めることができる。

0.2.3 論理回路と SAT

任意の組合せ回路は CNF 式により表現可能である。CNF(conjunctive Normal Form) 式とは、各項が変数の論理和のみで構成されており、その項の論理式として表現された論理式である。論理回路における基本的なゲートは表のような CNF 式に変換できる。各ゲートを CNF 式で表現したものをマージすることで論理回路全体を CNF で表現できる。

0.2.4 SAT を用いた多重故障用の ATPG 手法

1 に示される ATPG フローにより、多重故障に対する ATPG が可能である。このアルゴリズムにより得られたテストパターンの組に、冗長的なテストパターンが含まれることが起こり得る。一見すると、フローにおいて冗長的な処理はないようにも見えるが、これは、フローの後半に得られたテストパターンが、それまでに得られたテストパターンで検出可能な故障を全てカバーする可能性があるためである。このフローでテストパターンが追加される条件を言葉に言い換えるとすれば、「それまでのテストパターンでは出力を変化させない故障を検出するようなテストパターンが見つかった場合」である。

0.2.5 ABC

ABC はオープンソースの CAD ツールであり、検証やテストのための様々なプログラムを揃えている。その中の&fftest はテストパターン生成を実装したものである。同プログラム内では 1 に示されるフローチャートにしたがって ATPG が実行される。ISCAS89 のベンチマーク回路に対して ATPG を行なう際の手順を説明する。まず ISCAS89 の回路は、bench 形式で与えられる。ABC で順序回路に対する ATPG を

行なう場合には、組み合わせ回路部分に対するテストパターンが生成される。また、順序回路を扱う際には structual hashing が行われる。&fftest では AIG(AIG: And-Inverter-Graph) 形式で表現された回路情報を受け取り ATPG を実行する。AIG とは、論理回路あるいは論理関数を AND と NOT のみで表した表現であり、全ての組合せ回路・論理関数は AIG に変換可能である。この AIG 形式に含まれる情報と同じ情報を ASCII 形式で表現した内容を含む aag 形式がある。

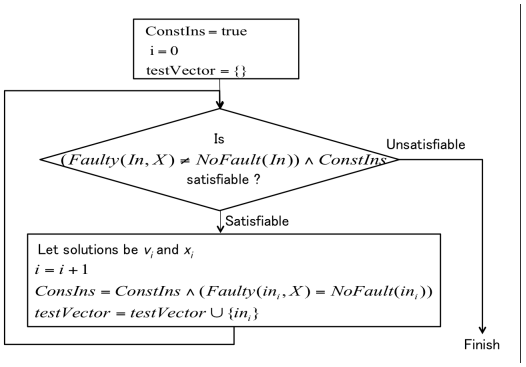

図 1: SAT を利用した ATPG のフロー

Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

0.3 単一縮退故障用と多重縮退故障用テストパターンの比較

0.3.1 実験環境

まず実験環境について述べる。ISCAS89 の各回路は、いずれも bench 形式で記述されている。この回路に対して structual hashing を行ない、AIG に変換したものをテスト対象の回路として扱い、ATPG の入力とした。初期の入力に用いる単一故障用のテストパターンには、文献 [] の手法で生成された、圧縮されたテストパターンを用いた。実験は、OS: LInux、CPU Intel Xeon 2.66GHz、メモリ 20GB の計算機上で行なった。

0.3.2 圧縮された単一用テストパターンを用いたテストパターン生成

ABC を用いて、圧縮された単一縮退用のテストパターンを元に行なった、多重用の ATPG の実験結果を示す。

実験結果をまとめたものを表 2 に示す。表 2 は、左から、回路の名称、圧縮された単一用のテストパターン数、生成された多重用のテストパターン数を示している。多重用のテストパターン生成において、圧縮されたテストパターンの導入により、入力なしの場合や、圧縮されてない単一用のテストパターンを用いた場合と比較して、いずれの回路においても、多重用のテストパターン数も抑えられていることが分かる。また、元の単一用のものと比較した多重用のテストパターン数は、比較的小規模の回路群においては、追加のテストパターンが不要であるか、１つまたは２つの追加のテストパターンが必要という結果になった。一方で大規模な回路群においては、追加のテストパターンを要したが、テストパターンの増分は、元のテストパターン数と比較して、３倍以内の個数に収まっている。

Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

表 1: 単一縮退故障用を用いた多重縮退故障用テストパターン生成

circuit	abc single no inputs	abc multiple no inputs	Additional test patterns
s27	12	11	-1
s208	50	82	32
s298	60	64	4
s344	59	87	28
s349	64	86	22
s382	77	87	10
s386	107	214	107
s400	63	85	22
s420	112	225	113
s444	65	73	8
s510	89	110	21
s526	86	159	66
s526n	94	170	66
s641	140	162	22
s713	127	148	19
s820	154	329	175
s832	162	334	172
s838	205	485	280
s953	138	197	59
s1196	257	398	141
s1238	247	548	301
s1423	179	182	3
s1488	232	284	52
s1494	211	357	146
s5378	595	517	-78
s9234	648	794	146
s13207	1181	-	-
s15850	-	793	-
s35932	-	-	-
s38417	-	-	-
s38584	-	-	-

Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

表 2: 圧縮された単一用を用いた多重用テストパターン生成

circuit	Number of compacted test patterns	tests	Additional test patterns
s27	5	5	0
s208	32	32	0
s298	28	28	0
s344	14	14	0
s349	15	15	0
s382	27	27	0
s386	64	64	0
s400	28	30	2
s420	70	70	0
s444	25	26	1
s510	58	58	0
s526	49	49	0
s526n	51	51	0
s641	25	25	0
s713	24	24	0
s820	99	99	0
s832	101	104	3
s838	142	142	0
s953	80	80	0
s1196	117	117	0
s1238	130	153	23
s1423	25	31	6
s1488	108	108	0
s1494	110	112	2
s5378	102	108	6
s9234	134	297	163
s13207	250	319	69
s15850	116	141	25
s35932	30	103	73
s38417	120	182	62
s38584	174	197	23

Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

0.4 単一用テストパターンでは検出できない多重故障

0.4.1 多重故障と単一故障の検出されやすさの比較

回路中に発生する多重故障は以下の２パターンに分類される。１．ある単一縮退故障と等価な多重故障

２．いずれの縮退故障とも非等価な多重故障

１．に分類される多重故障は、ある単一縮退故障と等価であるから、その単一縮退故障を検出するテストパターンにより検出が可能である。したがって、この分類の多重故障が検出不可能になるという場合はあり得ない。

次に、２，に分類される多重故障についてである。２．に分類される多重故障は、

0.4.2 単一用に追加のテストパターンが必要な場合

ここでは、具体的にどのような回路において追加のテストパターンが必要な状況が発生するかを、例題となる回路を用いながら議論する。

1. 単一縮退故障用のテストパターンをどのように選んでも、検出できない多重故障が回路中に存在する。

2. 単一縮退故障用のテストパターンの選び方が不適切なために、検出できない多重縮退故障が存在する。

1 の場合について説明を行う。この場合とは、回路中の全ての単一縮退故障の集合 $F\{f1,\cdots,fn\}$ に含まれる縮退故障の、２つ以上の組み合わせからなる多重故障が、F に含まれる故障を少なくとも１つ検出可能なテストパターンの集合に含まれる、いかなるテストパターンを用いても検出できない場合を指す。

ところが全ての入力は、期待された出力値と反対の値の、出力における縮退故障のいずれかを必ず検出可能である。つまり、いかなる単一故障も検出できないテストパターン、というものは存在しない。したがって、この場合については考えないこととする。

0.4.3 XOR における多重用テストパターン生成

ここでは、２入力 XOR ゲート単体における縮退故障を考えることにより、単一縮退用では検出できない多重故障が発生する場合について議論をする。以下、a での 0 縮退故障を sa0@a、 b での 1 縮退故障を sa1@b といったように表現する。sa1@a を検出可能なテストパターンは (0,*) であり、sa1@b を検出可能なのは (*,0) であるなお、「*」は don't care を示す。したがって、この二つの故障を最少のパターン数で検出するのは、(0,0) の１つの入力パターンから構成される集合である。ところが、このテストパターン（の集合）は、sa0@a と sa0@b が同時に発生した多重故障（以下、sa0@a, sa@b のように表す。）を検出できない。この例題を元に、問題を定式化する。

回路中における単一縮退故障の集合 $F\{f1,...fn\}$ を、集合のうちのいずれかのテストパターンにより検出可能なテストパターンの集合 T がある場合に、F に属する単一故障の２つ以上の組合せからなる多重故障を、T に属するいずれのテストパターンによっても検出不可である。

Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

0.4.4 特定の条件を満たす回路でのテストパターン生成

特定の条件を満たすような回路における、多重縮退故障検出について議論する。

図 4 のような、２つの論理 f, g の出力を XOR へ接続した回路を考える。f において縮退故障が発生した場合に f の論理が f1 に変化し、g の場合には g2 に変化するとする。(a1, b1) が f1 に対するテストパターンである時、

$f(a1)! = f1(a1)$

を満たす。この時、w2 の値は変化しない。

同様に、(a2, b2) が g2 に対するテストパターンである時、

$g(a2)! = g2(a2)$

を満たす。この時、w1 の値は変化しない。

さらに、(a3, b3) が f1, g2 での多重故障に対するテストパターンである時、

$f(a3)! = f1(a3) \wedge g(a3) = g2(a3)$

または

$f(a3) = f1(a3) \wedge g(a3)! = g2(a3)$

を満たす。

ここで、パターン (a0, b0) が f, g における２つの単一縮退故障に対するテストパターンであるとする。

この時、

$f(a0)! = f1(a0) \wedge g(a0)! = g2(a0)$

を満たす。

すなわち、二重故障が発生した場合には、w1 側, w2 側の両方の値を反転させるため、XOR の出力値は最終的に反転しない。したがって、この例題においては、単一用のテストパターン圧縮は、多重故障に対して有効でないといえる。

0.4.5 共通の入力をもつ論理の XOR

次に、図**??**のような、共通変数 c を入力にもつ、二つの論理 f,g の XOR を考える。この回路の論理は、

$out = xor(f(a, c), g(b, c))$

となる。

$\exists f1, f2. \forall a, b, c. f(a, c)! = f1(a, c) \Rightarrow f(b, c)! = g2(b, c)$

以上の命題が真である時、図**??**の回路において以下の２通りの状況が起こりえる。

1. f1 と g2 の多重故障が冗長故障となる。

2. 故障 f1 を活性化させる入力パターンは、必ず故障 g2 も活性化させる。

1. の場合には、冗長故障であるため、見かけ上回路は正常な動作をする。

2. の場合には、単一用に圧縮されたテストパターンでは、f1 と g2 の多重故障を検出できない。条件式より、f1 を活性化させる入力パターンは、同時に g2 も活性化させる。また、例題の回路では、f,g における

故障のいずれかのみが活性化した場合、必ず出力まで伝播する。つまり、f1 の単一故障を検出するテストパターンは、g2 の単一故障も検出する。

ここで、単一用にテストパターンの圧縮を考えると、g2 を検出可能なテストパターンは、f1 も検出するテストパターンとそうでないものの２通りある。ここでの議論で考慮する対象の故障は f1, g2 の二つであり、圧縮の結果、f1 も検出するテストパターンが選ばれる。

以上より、2 の場合に分類される状況では、単一用に圧縮されたテストパターンでは検出できない多重故障が存在する。

このような条件を満たす実際の回路の例として、図 2 のような回路を考える。表**??**に、図 2 の回路の真理値表に、回路中の各故障発生時の出力を示す。表**??**の各列は、各入力パターン、出力、各故障を示す。Double F の列は sa1@w1 と sa1@w2 からなる二重故障を示している。各行は回路における入力パターンに対応しており、ある行で横に見た時に、黄色になっている列は、その入力パターンにより検出可能な故障であることを示す。例えば、入力パターン (0,0,0) の行においては、sa1@w1, sa1@w2, sa1@o の列に色が付いており、この色が付いている列に対応する３つの故障を検出する故障を検出する事を表す。図 2 の回路では、想定する故障が sa1@w1 及び sa1@w2 の２つの単一故障のみであり、最少のテストパターンを考えると、(0,0,0) または (0,1,0) のいずれも１つのから成る集合になる。これらのテストパターンの集合はいずれも Double F を検出できない。次に、表中の全ての故障を想定する。表中の全ての単一故障を検出するテストパターンの集合は、sa0@o を検出する４つのパターンのうちの１つを含む。ここで Double F はこの sa0@o と等価な故障であるため、表中の全ての単一故障を検出するテストパターンの集合に属するテストパターンのうち、sa@0 により検出される。今回のケースでは、対象とする多重故障 Double F が、想定する単一故障のうちの１つと等価な故障であった。そのため、この多重故障は、どのように単一用のテストパターンを選択しても、必ず検出される多重故障である。特に、ある２入力ゲートの入力で同時に発生する二重故障は、出力の単一縮退故障と等価な故障として扱えるため、単一用のテストパターンにより検出される事がわかる。

したがって、どのような単一縮退故障とも等価でない事が、検出されない多重故障が存在するための必要条件である。すなわち、想定する故障の集合 F、F に属するすべての故障を検出するテストパターンの集合 T がある時、

$\forall t \in T.f_{multi}(t) = f_{nofault}(t) \rightarrow$

を満たすためには、

$\forall f \in F.f_{multi} \neq f$

を必ず満たす必要がある。

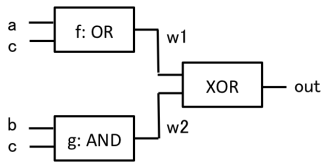

図 2: 条件を満たす回路の例

このような条件を満たす多重故障として、異なるゲートでの入力における多重故障を想定する。

Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

図 3 のような回路における全ての単一故障及び多重故障を考える。表 3 には、図 3 の回路における互いに非等価な全ての故障と、それらの故障が発生した時の出力値を示したものである。各列は、各入力パターン、出力、各故障を示す。表中のセルに付いている色について説明する。色によらず着色してあるセルはその行の入力パターンによって、その列の故障が検出される事を示す。そのうち、青色のセルは、全ての単一故障を検出するテストパターンの集合 $T_1()()()$ のうち、により検出される故障を示す。緑色のセルは、全ての単一故障を検出する別のテストパターンの集合 $T_2()()()$ のうち、により検出される故障を示す。赤色のセルは、全ての単一故障を検出するテストパターン集合 $T_1, T2$ に共通するテストパターンにより検出される故障を示す。つまり、この表においては、列ごとに見た時に、青色または赤色の行がない列は、T_1 により検出されず、緑色または赤色の行がない列は T_2 により検出されない。すなわち、T_1 のどのテストパターンでも Double F1 を検出できず、一方で T_2 のどのテストパターンでも Double F2 を検出できない。この回路では、単一用にテストパターンの圧縮を行うと、選ばれたテストパターン毎に、検出されない多重故障が発生するということになる。一般には、単一用の最少のテストパターン集合 $T_1, ...T_n$ の集合 T_min において、回路中の全ての多重故障 F_multi のうち、検出されない多重故障 f が存在する。

$\forall T \in T_min. \exists f \in F_multi$

表 3: 異なるゲートの入力における多重故障

a	b	c	out	Double F1	Double F2	sa1@w1	sa1@w2	sa0@w1	sa0@w2	sa1@a	sa1@b	sa1@o	sa0@o
0	0	0	0	1	0	1	1	0	0	0	0	1	0
0	0	1	1	1	1	0	1	1	0	1	1	1	0
0	1	0	0	0	1	1	1	0	0	1	0	1	0
0	1	1	1	0	1	0	1	1	0	0	1	1	0
1	0	0	0	1	0	1	1	0	0	0	1	1	0
1	0	1	1	1	0	0	1	1	0	1	0	1	0
1	1	0	1	0	0	1	0	0	1	1	1	1	0
1	1	1	0	0	0	0	0	1	1	0	0	1	0

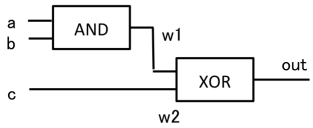

図 3: 異なるゲートの入力における多重故障

図**??**のような回路における全ての単一故障及び多重故障を考える。表 3 には、図 3 の回路における互いに非等価な全ての故障と、それらの故障が発生した時の出力値を示したものである。各列は、各入力パターン、出力、各故障を示す。表中のセルに付いている色について説明する。色によらず着色してあるセルはその行の入力パターンによって、その列の故障が検出される事を示す。そのうち、青色のセルは、全ての単一故障を検出するテストパターンの集合 $T_1()()()$ のうち、により検出される故障を示す。緑色のセルは、全ての単一故障を検出する別のテストパターンの集合 $T_2()()()$ のうち、により検出される故障を示す。赤色のセルは、全ての単一故障を検出するテストパターン集合 $T_1, T2$ に共通するテストパターンにより検出される故障を示す。つまり、この表においては、列ごとに見た時に、青色または赤色の行がない列は、T_1 により検出されず、緑色または赤色の行がない列は T_2 により検出されない。すなわち、T_1 のどのテストパターンでも Double F1 を検出できず、一方で T_2 のどのテストパターンでも Double F2 を検出できない。

Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

この回路では、単一用にテストパターンの圧縮を行うと、選ばれたテストパターン毎に、検出されない多重故障が発生するということになる。一般には、単一用の最少のテストパターン集合 $T_1, ...T_n$ の集合 T_min において、回路中の全ての多重故障 F_multi のうち、検出されない多重故障 f が存在する。

$\forall T \in T_min.\exists f \in F_multi$

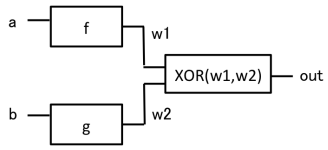

図 4: 条件を満たす回路の例題

表 4: XOR, OR, AND を用いた例題の回路における各故障発生時の真理値表

a	b	c	out	Double F	sa1@w1	sa1@w2	sa0@w1	sa0@w2	sa1@o	sa0@o
0	0	0	0	0	1	1	0	0	1	0
0	0	1	1	0	1	0	0	1	1	0
0	1	0	0	0	1	1	0	0	1	0
0	1	1	0	0	0	0	1	1	1	0
1	0	0	1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	0	1	1	0
1	1	0	1	0	1	0	0	1	1	0
1	1	1	0	0	0	0	1	1	1	0

Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

0.5 N 回検出用を用いた多重故障検出

0.5.1 実験の目的

N 回検出用のテストパターンによる多重故障検出の評価を行なうために、mentor 社が提供する商用 ATPG ツールの fastscan を用いて、N 回検出用の ATPG を行なった。

0.6 実験環境・条件

ISCAS89 の各回路に対してテストパターン生成を行なった。fastscan でのテストパターン生成の際、ABC を用いて、順序回路から組み合わせ回路部分を抽出し、その組み合わせ回路を verilog へ変換したものを、回路情報として与えた。実験は、OS: LInux、CPU Intel Xeon 2.66GHz、メモリ 20GB の計算機上で行なった。

0.6.1 N 回検出の実験結果

mentor 社が提供する商用 ATPG ツール fastscan を用いて、ISCAS89 の各回路における単一縮退故障の N 回検出用のテストパターン生成を行なった結果を示す。表**??**の各列はそれぞれ、各回路名、fastscan により生成された単一縮退故障の 1 回検出用のパターン数 (以降 N1)、2 回検出用のパターン数 (以降 N2)、1 回検出用を元に生成した多重故障用のテストパターン数、2 回検出を元に生成した多重故障用のテストパターン数、1 回検出用を元にした多重用のパターン数-N1、2 回検出用を元にした多重用のパターン数-N2、を示す。追加のテストパターンが必要な回路及びその追加数を比較すると、2 回検出の場合においても、概ね増分が一致するという結果を得られた。この増分は圧縮されたテストパターンを用いてテストパターン生成を行なった時の増分とも近いため、検出回数によるテストパターンの追加は、多重故障検出にはあまり有効でないといえる。

Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

表 5: fastscan で生成されたテストパターンを用いた多重縮退故障用テストパターン生成

circuit	bremen	abc multi with bremen	fastscan N=1 (N1)	fastscan N=2 (N2)	abc multi with N1	abc multi with N2
s27	5	5	7	11	11	12
s208	32	32	36	70	36	70
s298	28	28	35	59	35	59
s344	14	14	22	39	22	39
s349	15	15	23	39	23	39
s382	27	27	38	65	38	65
s386	64	64	72	139	72	139
s400	28	30	33	62	34	62
s420	70	70	75	146	75	146
s444	25	26	35	64	37	65
s510	58	58	67	122	67	122
s526	49	49	69	116	69	116
s526n	51	51	63	112	63	112
s641	25	25	44	80	44	80
s713	24	24	46	81	46	81
s820	99	99	116	211	116	212
s832	101	104	107	208	109	209
s838	142	142	153	295	153	295
s953	80	80	94	174	94	174
s1196	117	117	154	279	154	279
s1238	130	153	160	293	184	306
s1423	25	31	63	93	63	93
s1488	108	108	128	228	128	228
s1494	110	112	128	231	129	231
s5378	102	108	134	241	136	242
s9234	134	297	184	328	315	450
s13207	250	319	563	1088	-	-
s15850	116	141	313	545	-	-
s35932	30	103	158	258	-	-
s38417	120	182	543	879	-	-
s38584	174	197	665	1133	-	-

Vol.2015-SLDM-170 No.12
Vol.2015-EMB-36 No.12
2015/3/6

0.7 おわりに

本稿では単一縮退故障用のテストパターンを用いた多重縮退故障用のテストパターン生成について述べた。圧縮された単一用のテストパターンを用いて、検出対象の故障箇所の増分に対し少数のテストパターンの追加により多重縮退故障用のテストパターン生成を行なった。また、追加のテストパターン回路中にXOR ゲートを含む場合において、単一用に圧縮されたテストパターンでは検出されない多重故障が存在する例を示した。さらに、2 回検出用のテストパターンの集合を用いて多重故障用のテストパターン生成を行なった。ISCAS89 の各回路においては、2 回検出用のパターン数が 1 回検出用と比較しておよそ 2 倍になる点や、多重故障検出に必要な追加のパターン数が、1 回検出用の場合と比較して同程度である点から、2 回検出用を多重故障用に利用する方法が有効でない事を述べた。