

故障シミュレータを用いた ゲート／機能ブロック内部の故障絞り込み

勘田 芳正、木村 敬、買手 巧、桑原 教雄、下野 武志*

NECソフトウェア北陸 〒920-21 石川県石川郡鶴来町安養寺1番地

* 日本電気㈱ 〒183 東京都府中市日新町1-10

LSIの高密度化・微細化が進むにつれて、電子ビーム(EB)プローブによる故障診断が困難になって来ている。故障診断の作業を減らすために、事前に出来るだけ狭い範囲にまで故障箇所を絞り込んでおく必要がある。

これまでにも、テスターでのエラー状況と故障シミュレータなどを用いた故障の絞り込み手法が報告されているが、ゲートや機能ブロック内部に故障があった場合、故障による振る舞いが一致しないため正しい絞り込みが出来なかった。

今回提案する手法は、各ゲートや機能ブロックの出力ピンに縮退故障を定義して故障シミュレーションすることで、ゲートや機能ブロック内部の故障を絞り込むものである。本手法を紹介し、本手法を組み込んだシステムでの試行結果から有効性を示す。

A Logical Fault Diagnosis in a Gate or Function Block using Fault Simulation

Yoshimasa Kanda, Takashi Kimura, Takumi Kaite, Norio Kuwahara,
Takeshi Shimono*

NEC Software Hokuriku, Ltd.

1, Anyoji, Tsurugimachi, Ishikawa-gun, Ishikawa-ken 920-21, Japan

* NEC Corporation 1-10, Nisshincho, Fuchu, Tokyo 183, Japan

For VLSIs using a deep submicron process, a fault diagnosis using electron-beam(EB) probing becomes very difficult. It is strongly desired to localize the suspected faulty area as small as possible, before observing the faulty point by EB probing. Several methods for diagnosing logical faults are proposed for combinational circuits. However, they are useless for other type of faults such as internal faults of primitive gates.

This paper describes a efficient method to diagnose faults in a gate or function block. The effectiveness of the method is demonstrated by experimental results.

1. はじめに

LSIが故障を起こした場合、その故障箇所を検出し故障原因を調べる必要がある。故障箇所を調べるための手法として、電子ビーム（EB）テスターを用いて、LSI内部の信号を直接観測し故障箇所を診断する手法がいくつか報告されている[3, 4]。しかし、LSIの高密度化、配線

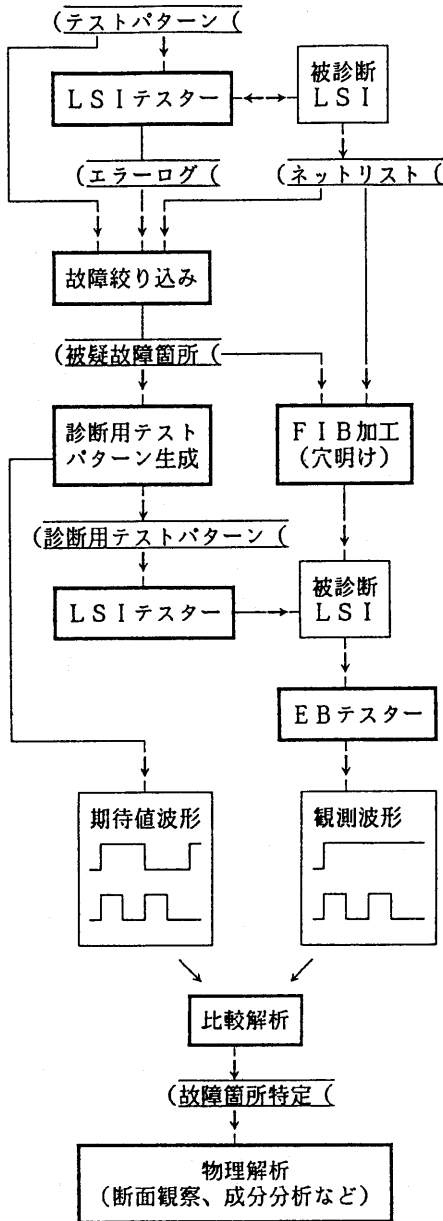


図1. 故障診断フロー

の多層化、微細化がますます進み、EBテスターによる故障診断が極めて困難になって来ている。

図1にEBテスターを用いた故障診断のフローを示す。LSIテスターでエラーが検出された場合、エラーの発生状態と回路論理から観測したい箇所を絞り込む。観測箇所が決まったならば、電源配線や他の信号線が通らない箇所など物理的に観測可能な箇所を選択し、その箇所に収束イオンビーム（FIB）を使って観測用の穴を明ける。EBプローブによる観測方法には電位コントラスト像を観測する方法と電位波形を観測する方法がある。電位コントラスト像は微細な配線パターンやFIBで明けた微小な穴では、非常に精度が悪くなり電位のHigh/Lowを判別するのが難しい。一方、電位波形を観測するには観測箇所を短周期でHigh/Lowに振るような入力パターンを作成する必要がある。

このように、一ヶ所を観測するだけでも非常に手間がかかる。従って、事前にできる限り故障箇所を絞り込んでおく必要がある。

従来も単一縮退故障用の故障シミュレータを用いて、LSIテスターでのエラー状況とゲートの入出力に定義した故障の振る舞いとが一致するものを絞り込み結果とする故障診断システムを開発したが、絞り込みで失敗することが多かった。絞り込むことができる故障の範囲を広げるには、Trレベルの回路モデルを用いたり、スタックオープン故障や短絡故障などいろいろな故障を扱える故障シミュレータを用いることが考えられる。しかし、扱う故障の数が飛躍的に増加し、LSI自体の回路規模を考えると実用的な方法とは言えない。

本論文では、扱う故障の種類を増大させることなく、より広い範囲の故障について絞り込みを可能とする方法を紹介する。まずTrレベルでの故障の振る舞いを示し、従来からの故障シミュレーションを用いた方法とその問題点について説明する。次に、今回の絞り込み手法を紹介し、本手法を取り込んだシステムを説明する。最後にISCA回路と実LSIでの試行結果を紹介する。

2. Trレベルの故障の振る舞い

ゲート内部の故障がゲートの論理動作に対してどのような影響を及ぼすかについて、例を挙げて説明する。

図2はXORゲートの内部回路を示したものである。Xの箇所が0縮退故障になるとA=1のとき下側のトランスマッションゲートがonになり、出力Yは1にしかならない。従って、論理はEORからA+BのOR論理に変わる。

このようにTrレベルでの縮退故障、オープン故障、短絡故障などは故障の位置によってゲートの論理が変わり、ゲートの入出力の縮退故障とは等価でないものが多い。また故障によっては出力値が不定となる場合もある。これらの故障の振る舞いを正しくシミュレーションしようとするTrレベルの回路モデルと故障モデルが必要となり、処理すべきデータ量は膨大になる。従って、

ごく小さな回路を除いてTrレベルで詳細な故障解析を行うのは困難である。

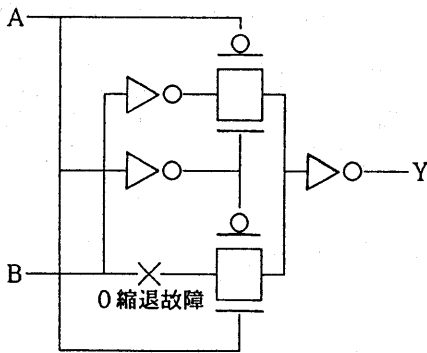


図2. XORゲート内部の故障例

3. 故障シミュレーションによる故障絞り込み

3.1 故障モデルに基づく故障絞り込み

縮退故障であれ短絡故障であれ、その故障モデルに対する故障シミュレータにより、それぞれの故障が存在する時の回路動作を求めることができる。この故障シミュレータにより得られた被診断回路の出力とLSIテスターで得られた回路出力とを比較することにより、動作が一致する故障を被疑故障として絞り込むことができる。この方法では故障シミュレーションの対象となる故障が実際の故障と一致する場合に、少ないテストパターンで効率良く正確な絞り込みを行うことができる。(図3)

- (1) 故障モデルに基づき全故障を定義する。
 - (2) LSI選別で使用したテストパターンを用いて1パターン故障シミュレーションを行う。
 - (3) LSIテスターにおいて、エラーになった出力ピンで検出されない故障を全て削除する。
 - (4) LSIテスターにおいて、期待値が一致した出力ピンで検出された故障を削除する。
- (2)~(4)を全パターンについて繰り返す。

図4(a)の回路を用いてこの手法による絞り込み例を説明する。図4(b)は検査入力とそれに対する出力応答で、「期/故」はそれぞれ期待値出力/故障回路出力を表す。この例では、1パターン目の出力Eでエラーになっている。故障としてゲートg1~g3の入出力の0/1縮退故障を対象とする。まず1パターン目で故障シミュレーションを行うと、ゲートg1の出力の0故障が出力E、F両方で検出され、ゲートg2の第2入

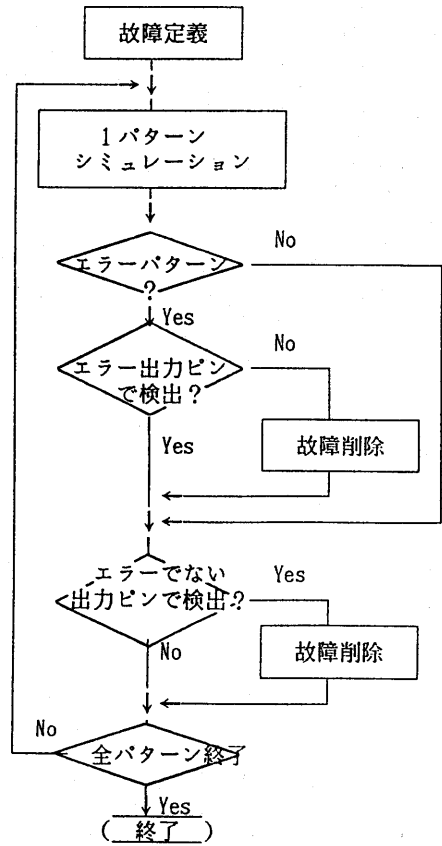
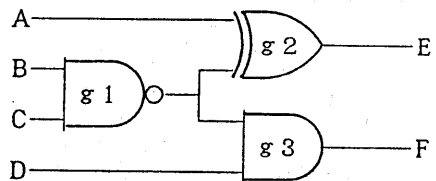


図3. 故障モデルに基づく故障絞り込みフロー



(a)

No	A	B	C	D	E		F	
					期/故	期/故		
1	0	0	0	1	1/0	1/1		
2	1	1	1	0	1/1	0/0		

(b)

図4. 被診断回路とテスト結果の例

力の0故障とg2の出力の0故障が出力Eだけで検出され、ゲートg3の第1、第2入力0故障とg3の出力の0故障が出力Fだけで検出される。出力Eでエラーになっていることからこの出力で検出される故障だけ残し、検出されない故障を全て削除する。更に、出力Fでエラーになっていないことから出力Fで検出される故障を削除する。この結果、ゲートg2の第2入力0故障とg2の出力0故障の2個だけが残る。次に残った2つの故障について2パターン目で故障シミュレーションを行う。その結果、ゲートg2の出力の0故障は出力Eで検出されるが、g2の第2入力0故障は検出されない。2パターン目で出力Eが正常であることから検出される故障が削除され、ゲートg2の第2入力0故障ただ1つだけが残る。このようにわずか2パターンで故障の絞り込みが完了する。

しかし、この方法では対象とした故障以外の故障があると絞り込みに失敗する。例えば、図2に示す故障が発生しg2がXOR論理からOR論理に変化した場合、ゲートの入力が共に1の時だけエラーとなる。つまり、パターン1のようにg2の入力が(0, 1)だった場合、出力Eは期待値と故障値が一致してしまうので、g2は被疑故障から削除されてしまう。

3.2 機能故障モデル

2節で述べたような各種の故障を一括して扱えるようにするために機能故障モデルを導入する。機能故障とはゲート内部の故障によりゲートの論理動作が変わる故障全てを指す。故障の単位は1つのゲートもしくは、複合ゲート、あるいは論理設計の最小単位としてセレクトや加算器などの機能ブロックが用意されていれば1つの機能ブロックを故障単位とする。ゲートや機能ブロック内部の実際の故障がTレベルの縮退故障、オープン故障、短絡故障あるいは他のどのような故障であろうと1つの機能故障として扱う。

(以下、ゲート/複合ゲート/機能ブロックなどの故障単位をゲートと記す。)

これにより多くの故障を1つの故障モデルで表すことが可能となり、かつTレベルの回路モデルを用いなくても済むので故障絞り込みの問題は非常に単純化される。しかし、任意の論理動作の違いを含む機能故障に対しては、3.1節で述べた故障絞り込み手法は残念ながら使うことは出来ない。

まず、基本的な条件を順に示す。

① 故障ゲートの出力は任意の値を取り得る

このような条件でも「ゲートの出力が何であってもエラーパターンと一致しない」というゲートを候補から除くことにより故障箇所の絞り込みに使用可能である。

② 各パターンで故障ゲートの出力は0又は1のどちらかであり、かつ故障ゲートの入力に変化しなければ故障ゲートの出力も変化しない

この条件は故障によりゲート内部で発振でも生じない限り適用できる。

③ 故障ゲートの出力はエラーパターンでエラー出力まで活性化されている(組合せ回路のみ)

順序回路では故障ゲートの出力に異常が現れるパターンと、外部出力にエラーが出るパターンとの間に数クロックのずれが生じる場合がある。しかし、組合せ回路ではパターンがずれることはないのでこの条件を使用できる。この条件はフリップフロップ回路に対しても適用可能である。

④ 故障ゲートの同一入力値に対する出力値はパターンによらず同じ(メモリ機能を持つ機能故障は除く)

この条件は順序回路/組合せ回路のどちらにも適用可能である。もともと組合せ論理であったゲートがメモリ機能を持つようになる故障を除去すると、この条件を与えることができる。この条件により被疑故障の候補を更に小さく絞り込める可能性があるが、実際の故障がメモリ機能をもつ故障だった場合、実際の故障箇所が被疑故障の候補に残らず、故障の絞り込みに失敗する危険性がある。よって適切な診断を行うには、最初は④の条件を加えずに絞り込みを行い、その結果と④を加えた結果を比較して判断する必要がある。

3.3 組合せ回路での機能故障の絞り込み

次に、前節で述べた組合せ回路に限定した条件③「故障ゲートの出力はエラーパターンでエラー出力まで活性化されている」について、詳細に説明する。

これは0/1縮退故障用の故障シミュレータを利用して実現出来る。また、被疑故障ゲートの出力値を不定値Xとしたシミュレーションも、通常縮退故障で、値0または1を故障として定義する代わりに故障値Xを定義するだけで、縮退故障の場合と同じ故障伝搬が良い。つまり、ゲートの演算の結果、正マシンと故障マシンとで出力値が違っていればその故障が伝搬する。回路出力まで伝搬すれば、その故障が存在する時回路出力が不定値Xを取り得ることを意味する。

故障絞り込み手順は以下のようなになる。

- (1) 全ゲートに対して、1出力のものへは出力に0/1縮退故障を、多出力のものへは全出力にX値故障を定義する。
- (2) LSIテスターでエラーとなったパターンを1パターン故障シミュレーションする。
- (3) エラー出力ピンで検出されない故障を削除する。
- (4) 1出力のゲートの故障で、エラーでない出力ピンで検出される故障を削除する。

(5) 故障が残ったゲートに、1出力のものへは出力に0/1縮退故障を、多出力のものへは全出力にX値故障を定義する。

(2)~(5)を全てのエラーパターンについて繰り返す。

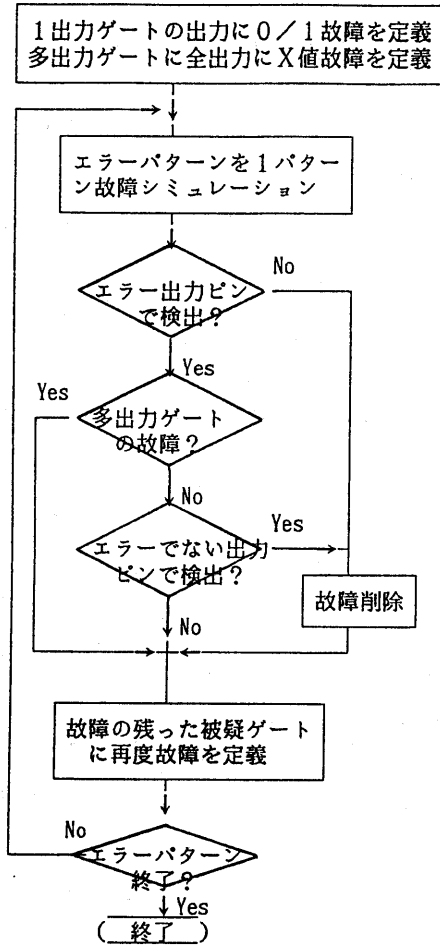


図5. 組合せ回路における機能故障絞り込みフロー

このフローは、図3の故障モデルに基づく故障の絞り込み方法にいくつかの違いがある。

まず、エラーパターンしか使わない点で異なる。この理由は、エラーでないパターンでは故障ゲートが正常値と同じ出力値を持っている可能性があるため故障が検出されても矛盾とならないからである。これにより3. 1節で説明した問題を防止できる。

次に、多出力ゲートの場合にX値の故障を定義している点が違う。多出力ゲートの内部に故障が存在した場合、故障の影響がゲートの全出力に影響

する事もある、ゲートの1出力にしか影響しないこともある。また、0/1の縮退故障を定義しても、再収斂などでゲートの出力信号同志が影響しあって故障の影響が正しくシミュレーションされない。従って、ゲートの全出力にX値を定義し伝搬させることで故障が影響する回路出力ピンを判定できる。

最後に、残ったゲートに再度故障を定義する点が違う。あるエラーパターンで0故障が検出され、あるエラーパターンでは1故障が検出されるということがある。この場合、検出されない反対の値の故障がそれぞれ削除されてしまい被疑ゲートが消えてしまう。故障を再定義するのはこれを防止するためである。

4. 開発システム概要

本手法を用いたシステムをEWS4800上に構築した。設計される回路はフルスキャン回路がほとんどなので、現状は、組合せ回路に限定したシステムとなっている。

開発システムは次のプログラムから構成される

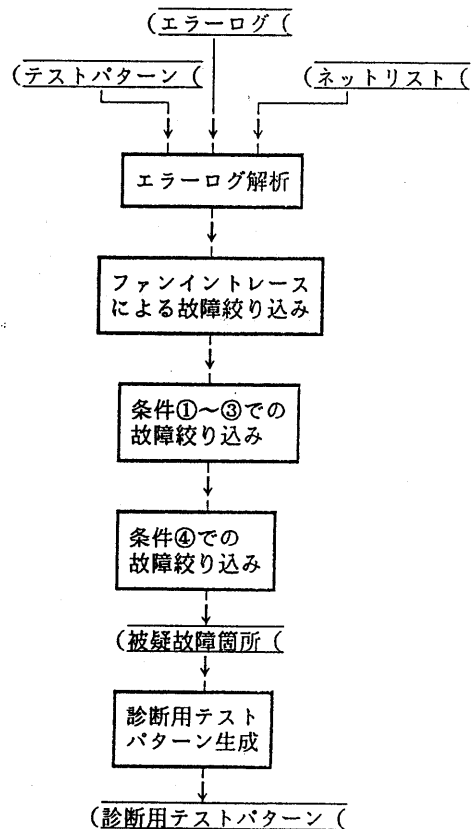


図6. 開発システム概要

- (1) エラーログ解析
 テスターで出力されたエラー情報を読む。
- (2) ファンイントレースによる絞り込み
 被試験回路のネットリストで、エラー検出箇所からファンイントレースして被疑故障ゲートを抽出する。
- (3) 条件①～③での絞り込み
 故障シミュレータを用いて、3. 2節で述べた条件①～③を用いて絞り込みを実施する。図5によるフローを実施している。
- (4) 条件④での絞り込み
 3. 2節の条件④を用いて、さらに絞り込みを行う。テストパターン全体に対して被疑故障ゲートの入力値と出力値を求め、矛盾があった場合その故障を削除している。
- (5) 診断用テストパターン作成
 絞り込まれた故障に対して、ATPG(Auto Test Pattern Generator)を用いて故障箇所を活性化するパターン作成し、EBテスター用のテストパターンに編集する。

5. 試行結果

ISCASベンチマーク回路と実LSIにおいて機能故障を仮定し、本手法を適用してみた。機能故障は回路中からランダムに選択したゲートの論理を適当に変えることによって故障を設定した。ISCASベンチマーク回路は全て基本ゲートで構成されており、2章で述べたようなT_rレベルで意味のある機能故障を定義することが難しい。そこで、ANDゲートがOR論理に変わるような実際にはほとんど起こらないような機能故障も含まれている。機能故障は各回路に5ヶ所準備し、それぞれの機能故障について絞り込みを実施した。LSIテスターで出力されるエラーログの代わりに論理シミュレーションにより期待値比較の結果を用いている。

表1に評価に使用した回路の特徴を示す。回路Aと回路BはNECで設計されたCMOSゲートアレイである。表2は、評価に使用したテストパターン数を示す。テストパターンはいつでもATPGで作成した。

表3に絞り込み結果を示す。「エラー数」は各機能故障が存在する時にテストパターンの内何パターンでエラーとなるかを示している。「被疑故障数1」は回路のファンイントレースで残る被疑故障ゲートの数、「被疑故障数2」は3. 2節の条件①～③での絞り込みで残る被疑故障ゲート数、「被疑故障数3」は3. 2節の条件④を追加して残る被疑故障ゲート数を示している。

ファンイントレースによる絞り込みでは、ゲート数が比較的小さいものでは絞り込み効果にバラツキがあるが、規模の大きい回路Bでは元のゲート数の80%以上を削除できている。この絞り込みは、回路をファンイントレースするだけなので

非常に短時間で処理でき、次処理の故障シミュレーションの負荷を軽減できている。

絞り込まれた被疑故障には、設定した故障ゲートが含まれており、いずれも絞り込みは成功している。また、絞り込んだ結果被疑故障ゲートが1となった場合も多く見られ、本手法で故障箇所を的中できることを表している。「被疑故障数2」及び「被疑故障数3」を見ると回路規模に対して少ない数になっただけでなく、回路規模によらず絶対量としても少ない数を示している。

「被疑故障数3」が1でないものを調べてみると、故障が等価な関係であった。例えば、回路Bの故障番号4では5つの故障ゲートが残るが、EBテスターで信号値を見なければ故障ゲートを判定出来ないものである。(図7)

本評価では、メモリ機能を持つ故障を入れてないので条件④まで故障ゲートが残っている。しかし、実LSIの故障での解析事例が少ないので、条件⑤まで残ったが条件④で削除された被疑故障ゲートがどの程度確からしいかは不明である。ただ現状は、条件④まで残った被疑故障ゲートを優先して解析している。

このように、被疑故障箇所を数個にまで絞り込みできれば、比較解析や物理解析も実用的な範囲と言える。

表1. 評価用回路データ

回路名	ゲート数	入力数	出力数	双方向数	キヤン長
C880	383	60	26	0	0
C1355	546	41	32	0	0
C3540	1669	50	22	0	0
C7552	3512	206	107	0	0
A	8414	46	5	160	1023
B	52342	57	234	35	4193

表2. 使用テストパターン数

回路名	テストパターン数
C880	81
C1355	131
C3540	236
C7552	294
A	2083
B	3345

表3. 機能故障絞り込み結果

回路名	No.	エラー数	被疑故障数1	被疑故障数2	被疑故障数3
C880	1	58	34	11	2
	2	19	193	1	1
	3	10	159	2	1
	4	6	63	7	1
	5	81	5	3	3
	平均		34.8	90.8	4.8
C1355	1	93	546	8	2
	2	60	540	7	3
	3	67	522	10	4
	4	128	340	1	1
	5	11	355	3	2
	平均		71.8	460.6	5.8
C3540	1	9	1495	5	2
	2	24	1513	1	1
	3	19	1505	7	2
	4	111	1517	1	1
	5	69	1510	7	3
	平均		46.4	1508.0	4.2
C7552	1	37	1456	1	1
	2	25	619	4	4
	3	67	598	23	2
	4	74	395	8	3
	5	92	474	25	4
	平均		59.0	708.4	12.2
A	1	754	3088	2	1
	2	33	1067	11	1
	3	30	397	5	1
	4	234	286	2	2
	5	641	860	2	1
	平均		338.4	1139.6	4.4
B	1	130	6432	3	1
	2	45	6455	7	1
	3	33	6445	10	1
	4	3	7180	11	5
	5	55	10253	5	2
	平均		53.2	7353.0	7.2

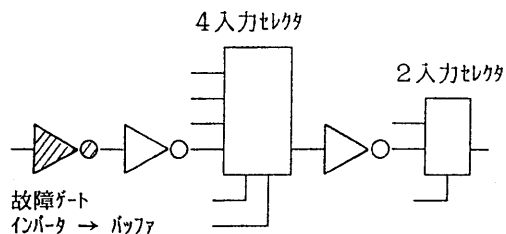


図7. 回路B故障番号4の絞り込み結果

6. 結び

本論文ではゲート/機能ブロックの内部をブラックボックスとして扱い、T_rレベルの回路モデルを用いることなく従来の故障シミュレーション手法を用いて故障を絞り込む手法について述べた。さらに本手法を用いたシステムを開発し、実LSIでの有効性を確かめた。現在このシステムはNEC第一コンピュータ事業本部・コンピュータ事業部で運用を始めており、いくつかの事が新たに分かって来ている。例えば、試行結果にあるような多くのエラーボタンが無くても、実質的に絞り込みが可能であることも分かった。

現在は、現状システムの運用性の改善や多分岐ネットのオープン故障や配線短絡故障、スキャンパス上の故障やバーシャルスキャン回路への対応を検討しており、さらに良いシステムにして行く予定である。

[参考文献]

- (1) T. Shimono : "An Efficient Logical Diagnosis for Combinational Circuits using Stuck-at Fault Simulation", Proc. ATS, pp. 76-81, 1994.
- (2) 下野 : 「故障シミュレーションを用いたゲート/機能ブロック内部の故障絞り込みに関する一手法」第31回FTC研究会資料 1994.
- (3) K. Yamazaki and T. Yamada : "Method of Diagnosing Logical Faults in Combinational Circuits", Proc. ATS, pp. 170-175, 1992.
- (4) N. Itazaki, T. Sumioka, S. Kajihara and K. Kinoshita : "Automatic Fault Location Using E-beam and LSI Testers", Proc. ATS, pp. 255-260, 1993.
- (5) 加藤、松尾、村瀬、大谷、加藤、下野、河合 : 「故障シミュレーションを用いたLSIの故障解析事例」日本學術振興会第132委員会第109回研究会資料 pp. 72-76, 1989.