

テスト不能故障抽出のための一手法

A technique for the undetectable fault recognition

辺保 久、牛久保 政憲、菊地原 秀行、村上 道郎

Henbo Hisashi, Ushikubo Masanori, Kikuchihara Hideyuki, Murakami Michio

沖電気工業株式会社

OKI Electric Industry Company, Ltd.

あらまし 故障シミュレーションの処理時間を大きく左右しているものの一つに、検出不可な故障(テスト不能故障)がある。しかし、これらの故障は、回路構成に深く関係しているため、その限定が困難であった。今回、筆者らは、これらの故障を、合理的に抽出する手法(Hアルゴリズム)を考案した。本手法は、正常回路および故障回路における、各信号ラインの変化の可能性を解析することにより、テスト不能故障を抽出するものである。本手法を適用した結果、ある回路では、全故障の約18%にのぼる故障(テスト不能)の抽出が行われ、故障シミュレーションの処理時間が約64%減少した。

Abstract The performance of fault simulation is much influenced by undetectable faults. These faults are resulted from the circuit structure. Thus, the recognition of these faults is very difficult. We thought out a technique (H-algorithm) to recognize these faults theoretically. This technique analyzes the variability of every signal value in a good circuit and faulty circuits, and drops the undetectable faults.

By using of this technique, 18% in all faults was dropped and 64% in fault simulation CPU time was cut in a circuit.

1 はじめに

近年のLSIの大規模化に伴い、筆者らは、コンカレント法による故障シミュレータを核とした、故障解析プロセッサ(FLAP)^{(1)・(2)}を開発し、その適用を図っている。

通常、故障シミュレーションに費やすCPU時間は、論理シミュレーションの5倍から10倍以内であるが、中には、10倍を超えるものもある。そこで筆者らは、詳細な調査分析を行い、故障シミュレーションのパフォーマンスを大幅に劣化させているごく一部の故障(重負荷故障)の存在を明らかにした⁽³⁾。また、これらの故障の性質を利用した高速化のための

一手法を先に提案した⁽⁴⁾。

今回の発表では、それら重負荷故障に故障シミュレーションでは検出できない故障(以後テスト不能故障と呼ぶ)が、数多く含まれていることに着眼した。これらの故障は、故障シミュレーションの対象故障としては意味がなく、また、対象故障となった場合、故障シミュレーションのパフォーマンスを劣化させるものである。しかし、これらの故障の対処法については、今まで、あまり明確に報告されていない。

そこで、これらの故障をシミュレーションの対象外とすべく、テスト不能故障抽出の一手法(Hアルゴリズム)を考案した。

本稿では、本手法について詳細に述べるとともに、その試行結果について報告する。

2 テスト不能故障

テスト不能故障とは、故障シミュレーションで検出できない故障を意味し、以下に、その故障について、図1の回路を用いて述べる。

なお、本論文における故障とは、全て、単一縮退故障を意味する。

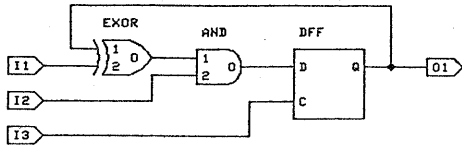


図1

図1は、EXOR、AND、D-フリップフロップ(DFF)の3素子で構成された回路である。本図において、DFFのC端子(以後、素子名、端子名の形式で記す)の0、1両故障は、その故障の影響で、クロックエッジを形成できず本フリップフロップの出力値は、確定しない。よって出力Q端子は、常に信号値X(不定値)の出力となり本両故障は、テスト不能故障となる。

また、AND、2の1故障による故障回路では、ANDの出力は、AND、1の値により決定されるが、本回路では、そのフィードバックラインにより、本信号値は、本素子の出力に依存する。このことから、ANDの出力は、初期状態値Xのままとなり、本故障もまた、テスト不能故障となる。

しかし、本回路が、ある回路の部分回路であった場合、これらの故障と同じ効果を生じさせる故障であっても、テスト不能故障でない場合がある。このように、テスト不能故障は、その回路構成に依存するため、その限定は非常に困難である。よって従来は、故障として定義すべきものでないにもかかわらず、除外することができなかった。

また、これらテスト不能故障は、その性質により、故障シミュレーション中において、常に解析の対象となり、かつ、それらの多くは、状態値Xをもつ故障リストを多量に発生させるため、故障シミュレーションのパフォーマンスを

著しく劣化させる^{[3]・[4]}。

これらのテスト不能故障を、抽出するのが、今から述べるHアルゴリズムである。

3 Hアルゴリズム

本手法は、まず、正常回路および故障回路における各端子(信号ライン)の取りうる値を求め、その結果に基づいて、テスト不能故障群を抽出するものである。本手法は、処理及び機能面から大きく以下の二つに分かれている。

・制御性解析部

正常回路と全故障回路の各信号ラインの信号値の変化の可能性を求める。

・テスト性解析部

正常回路と全故障回路の各信号ラインの信号値の変化の可能性から、テスト不能故障を抽出する。

以下に、それぞれについて詳細に述べる。

なお、本章では、簡単化のため、信号値として0、1、Xの3値を扱う故障シミュレータを前提としている。

3.1 制御性解析部

本解析部は、以下に述べる制御値、および制御演算表をもとに故障シミュレーション手法を用いて、正常回路および故障回路の各信号ラインの制御性を求めている。

3.1.1 制御値

制御値とは、回路中の各信号ラインの信号値の、変化の可能性を表現するものであり、その種類と定義について、表1に示す。

制御値	定義
0	信号値0にしか制御できないことを意味する。
1	信号値1にしか制御できないことを意味する。
X	信号値0、1のどちらにも制御できないことを意味する。
V	信号値0、1のどちらにも制御可能であることを意味する。

表1 制御値

たとえば、ある信号ラインが、制御値0の場合、本ラインは、信号値0をとるよう制御できる（初期化できる）が、信号値1には、制御できないことを意味する。制御値1の場合も同様である。ただし、制御値Vの時は、当ラインは、信号値0、1のどちらにも制御できる可能性があることのみを意味する。よって、実際には、どちらか一方にしか制御できない場合がある。これは、制御値Vが、信号値0と1の両方の特質を兼ね備えた値であることと、Vの反転をVとして扱っていること等に起因している。

3.1.2 制御演算表

本制御演算表は、素子の入力ラインの制御性から出力ラインの制御性を得るために用いられるものであり、先に述べた制御値の定義、および当該素子の論理表から、ユニークにかつ簡単に作成できる。

具体例として、以下2入力ANDを例に取りその作成方法について述べる。

2入力ANDの論理表と制御演算表を表2に示す。

論理表(AND)			制御演算表(AND)		
1	2	0	1	2	0
0	0	0	0	0	0
0	1	X	0	1	X
0	1	0	0	1	0
1	1	1	1	1	1
1	1	X	1	1	X
1	X	0	1	X	0
1	X	1	1	X	1
1	X	X	1	X	X
1	X	X	1	X	X



表2 ANDの論理表と制御演算表

表2の制御演算表において、制御値Vが、入力に存在しない部分は、論理表と全く同じ（記号の意味の置き換えのみ）となる。入力としてVが存在する場合、たとえば制御値(V、0)の入力では、入力信号値としては、(0、0)、(1、0)の2通りの可能性があり、本入力の組み合わせの場合、出力信号値はともに0であることから、出力制御値は、0となる。また、入力(V、X)では、同様に行うことにより出力信号値としては、0とXの場合があるが、ライ

ンの制御性の観点から、その出力制御値は、0となる。

このように、入力(フリップフロップの内部状態値も含む)にVが存在する場合、その出力制御値は、

「Vを信号値0と1にそれぞれ置き換え、得られる全組み合わせの入力に対する、全出力値を考慮すること」

により求められる。

このほか、信号の変化(クロックエッジ)が影響し、また、演算により内部状態値を変化させるフリップフロップ等では、本制御値の定義から、以下の2点に留意すればよい。

- (1) クロック端子にVが入力する場合は、信号値0、1の他にクロックエッジが入力する可能性もあること。
- (2) 内部状態値の変化が生じる場合、その変化後の内部状態値をもとにした入力の組み合わせも考慮すること。

次に、本解析部での処理フローを述べる。

3.1.3 処理フロー

本解析部の処理フローを、図2に示す。

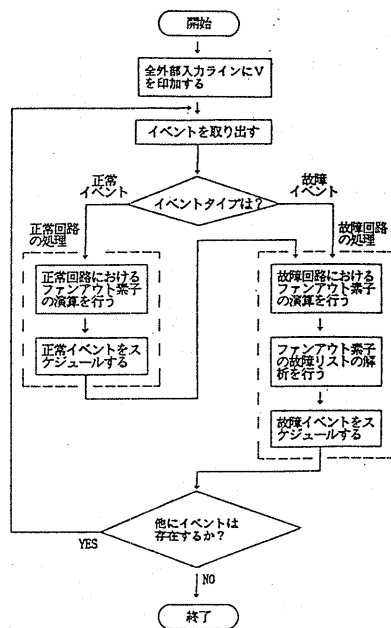


図2 制御性解析部処理フロー

本図から本フローは、故障シミュレーションフローと、ほとんど同じであることが理解される。本フローでの特筆すべき点は、次の二点である。

一点目は、全外部入力ラインに制御値Vを印加していることである。これは、当入力ラインは、テストパターンにより、信号値0にも1にも制御できるからである。また、本パターンは、全ての入力パターンを、圧縮したパターンであると言える。

二点目は、本フローの正常回路の処理の部分で行われる故障挿入にある。故障挿入においては、その制御値の意味および故障の性質により、正常回路のある信号ラインが、制御値1のときは、当ラインの0故障が、0のときは、1故障が挿入の対象となる。また、Vのときは、0、1両方の故障が、挿入の対象となり、それぞれの故障回路では、0故障の場合、制御値0が、1故障の場合、制御値1が挿入される。

次に、簡単な回路を用いて、その処理例を説明する。

3.1.4 処理例

ここでは、第2章のテスト不能故障で、例示した図1の回路を用い、故障としては、テスト不能故障であるAND.2の1故障のみを仮定して以下説明する。また、EXORおよびDFFの制御演算表については、本説明で必要な部分のみを、表3に示す。

制御演算表(EXOR)			制御演算表(DFF)				
1	2	A	D	C	Q	Q	
X	0	V	X	0	V	X	0
0	V	V	V	V	0	0	V
V	V	V	V	V	0	0	V

表3 EXORとDFFの制御演算表

まず、外部入力ライン全てに対し、制御値Vを印加する。本状態を、図3に示す。ただし、各ラインの初期状態値は、制御されていない状態を仮定し、全て制御値Xとする。

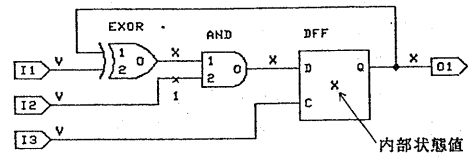


図3

このことにより、EXOR.2とDFF.Cの両端子では、正常回路および故障回路とも初期状態値Xから制御値Vに変化する。しかし、AND.2は、正常回路では制御値Vを入力するが、故障回路では、故障の挿入が行われ、制御値1の入力となる。

次に、それぞれの制御演算表(表2と3を参照)を用いて3素子の演算が行われ、それぞれの出力値が決定される。ここでのイベントの発生は、AND.0の端子のみで、正常回路では、その入力(X, V)とANDの制御演算表(表2参照)から、制御値0となるが、故障回路では、(X, 1)の入力であるため、その出力制御値は、Xのままである。この状態を示したのが、図4である。

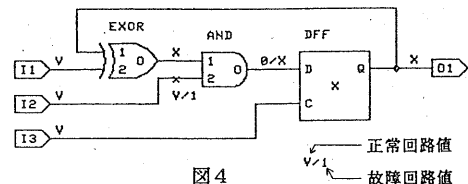


図4

さらに、本イベントを受けDFFの出力およびEXORの出力が変化することにより図5の状態となる。

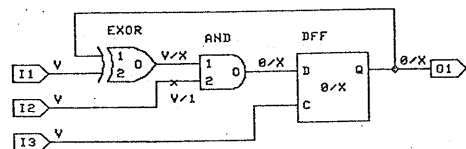


図5

その後、最終的には、全てのイベントが処理され、図6の状態となる。本解析により、正常および故障回路で、どのラインが、信号値0あるいは、1に制御できないかが明確になる。

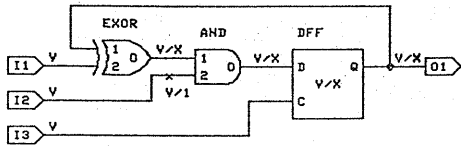


図6

このようにして求めた、正常回路および故障回路における任意のラインの信号値の変化の可能性をもとに、次に述べるテスト性解析部でテスト不能な故障の抽出を行っている。

3.2 テスト性解析部

本解析部は、先に述べた制御性解析部で得た制御に関する、正常回路および故障回路の全ラインの状態値をもとに、テスト不能故障の抽出を行っている。まず、以後の説明の明確化のため、本解析部で、キーとなる故障隠滅条件なるものの定義および命題を以下に示す。

故障隠滅条件：正常回路と故障回路の状態制御値が0もしくは1で全く同じか、少なくともどちらか一方がXである。

命題：ある故障回路で、当故障オリジンから外部出力ラインに至る全ての経路（故障オリジンは含まない）において、故障隠滅条件を満たすポイントが存在するならば、本故障は、テスト不能故障である。

本命題は、外部出力ラインにおいて、正常回路と故障回路で、信号値0と1の差異が生じないことを証明すればよく、これは、以下の帰納法的証明により明らかである。

- (1) 上記故障隠滅条件となったポイントでは、正常回路と故障回路の取り得る信号値が、0と1（順不同）となる組み合わせはない。

また、単一縮退故障を前提としているため、当該経路の任意のネットと素子は、正常回路と故障回路で同一機能を有する。よって、以下のことが言える。

- (2) ネットは、正常回路と故障回路で、その

信号入力側で、信号値0と1での差異がなければ、その出力側に、信号値0と1での差異を生じない。

- (3) AND、インバータ素子は、正常回路と故障回路の任意の入力ラインで、信号値0と1での差異がなければ、出力に0と1での差異を生じない。また、全ての論理回路（ワイアードネットを含む）は、ANDとインバータ素子およびネットの組み合わせで構成できる。

本解析部では、以上述べたテスト不能故障の性質に着目し故障の抽出を行っている。

3.2.1 処理概要

ここでは、今回の適用で行った処理概要について説明する。

本処理では、まず外部テストピンからのバックトレースにより、上記故障隠滅条件を満たす故障を、テスト不能故障の可能性のある故障として洗いだし、その後、当該故障のそれぞれについてフォワードトレースを行い上記命題を満たす故障を、テスト不能故障として抽出した。

たとえば、前節で説明した図6の場合、AND.2の1故障は、AND.0の端子で本命題を満たし、テスト不能故障として抽出される。

4 適用

本アルゴリズムを、実際に弊社のFLAPに適用するに当たり、トライステート素子（ハイインピーダンス状態(Z))も扱えるよう以下の点で拡張を行った。

- (1) ハイインピーダンス状態に関する制御性も考慮するため、制御値としては、4値から8値に拡張したものを使用した。
- (2) テスト性解析部における、条件の拡張および、トライステート素子を含めての証明を行った。

これらの詳細については、紙面の都合上、省略する。

5 評価結果

評価に用いたサンプル回路を、表4に示す。本表における故障数は、全素子の全入力及び出

力端子に定義した故障から、以下の、故障削除を経て定義された故障である^[1]。

- (1) 等価故障解析による故障削除。
- (2) 外部クロックピンからのみ影響を受けるライン上の故障の削除。(ベースクロックライン上の故障は、実際の検査時に、容易に検出されると考えられることによる)
- (3) 固定ライン上の同一論理値の故障。

なお、(2)の一部(テスト不能故障)および(3)の故障は、本手法により抽出可能であるが、今回の評価では、対象外とした。また、回路1と回路2における故障検出率は、それぞれ、77.2%と87.3%であった。

	ゲート数	テストベクター数	故障数
回路1	3827	3351	5004
回路2	3233	20122	3366

表4 サンプル回路

以下、それぞれのサンプル回路について、故障数、CPU時間と故障リスト量を評価項目にして試行結果のデータを提示する。

5.1 故障数

本手法適用により、抽出されたテスト不能故障数およびそれが全故障数に占めていた割合を表5に示す。これら抽出された故障のほとんどは、故障回路において、初期化されない部分を持つもので、内部のクロックラインに故障効果を与えるものであった。特に、回路1では全素子におけるフリップフロップの占める割合が、回路2より高く、そのことにより、抽出故障の割合も高くなった。

	抽出故障数	抽出故障の割合
回路1	897	17.9%
回路2	220	6.5%

表5 抽出故障数

5.2 故障シミュレーションのcpu時間と故障リスト

回路1、2それぞれにおいて、故障シミュレーションにおけるcpu時間と故障リストに関するデータを表6に示す。

	cpu比	領域比
回路1	35.9	34.6
回路2	78.7	86.1

表6 故障シミュレーション評価結果

本表は、本手法適用前のcpu時間と最大故障リスト量(領域)を、それぞれ100とした場合の比率で示してある。ただし、cpu時間に関しては、抽出に要した時間も含めて算出したデータを示した。

以上のことから回路1では、約18%の合理的な故障削除により、約64%のcpu時間の削減が得られたことがわかる。

6 おわりに

故障シミュレーションのパフォーマンスを多分に劣化させるテスト不能故障を抽出する一手法を提示した。

本手法は、テスト不能故障を完全に抽出する方法ではないが、抽出された故障が全てテスト不能故障であることについては保証され、その結果、故障シミュレーションの故障の検出の精度に全く影響を与えない。また、現在評価の過程であるが、今回得られた試行結果のデータにより、故障シミュレーションの高速化に非常に有効であると考えられる。

なお、本手法は、既存の故障シミュレーション手法を用いているため、その適用は、容易であった。

今後、さらに評価を深めその有効性を図っていく予定である。

参考文献

- [1] 辺保、牛久保、木幡、菊地原、村上、"FLAP: 故障解析プロセッサ 一機能概要一"、第33回情処全大7R-5、P2297
- [2] 牛久保、辺保、菊地原、村上、"FLAP: 故障解析プロセッサ 一シミュレーション部評価一"、第33回情処全大、7R-6、p2299
- [3] 牛久保他、"故障シミュレーションのパフォーマンス低下を招く故障について"、設計自動化夏期シンポジウム(1987.8.7)
- [4] 牛久保、辺保、菊地原、村上、"故障シミュレーション高速化のための動的故障削除手法"、電子情報通信学会技術研究報告、vol.87、no.11、VLD87-3、p17(1987.4.22)