

誤り経路追跡の診断分解能向上のためのテスト生成法

日大生産工 (学部) ○千田 祐弥 日大生産工 細川 利典
明治大学 山崎 浩二

1. はじめに

半導体微細化技術の進歩に伴い、超大規模集積回路 (Very Large Scale Integrated circuits : VLSI) において、異常動作の物理的な原因を特定する故障解析[1]は、歩留まりの向上のために重要である。故障解析では、電子顕微鏡などを用いて故障 VLSI 内部の観測を行うため、多大なコストを要する。そのため、故障 VLSI に存在する可能性のある故障 (被疑故障) の数を事前にできる限り絞り込んでおく故障診断[2]が、故障解析コストの低減のために重要となる。故障診断では、故障 VLSI の異常な外部出力応答を裏付けることのできる故障箇所を推定する。

組合せ回路やスキャン設計された順序回路における単一縮退故障モデルの故障診断手法は様々なものが提案されており、被疑故障数も大きく削減できることが報告されている[2]。

被疑故障は、故障 VLSI に対してテスト集合を入力として与え、誤り経路追跡を実行することによって推定される。VLSI の設計現場では、設計された順序回路に対して高い故障検出率を達成するテスト集合を得るために、フルスキャン設計と組合せ回路のテストパターン自動生成ツール (Automatic Test Pattern Generator : ATPG) を使用したテスト設計方法が広く用いられている。順序回路にスキャン設計を施すことにより、疑似的に組合せ回路として扱うことが可能になり、組合せ回路のテスト生成技術を利用したテスト生成が可能となる。組合せ回路に関しては、効率的なテスト生成アルゴリズムが提案されており[3-7]、大規模な回路であっても、現実的な時間で高い故障検出率を達成するテスト集合の生成が可能となっている。しかしながら、高い故障検出率を達成するテスト集合が必ずしも高い故障診断分解能を達成するとは限らない。テストベクトル数を増加させれば、故障診断分解能が向上し、被疑故障数の削減が可能だが、テストコストが増加するという問題がある。テストコストを増加させることなく、被疑故障数を削減するためには高い故障検出率と故障診断分解能を両立したテスト集合の生成が重要である。

現在、一般のテスト生成のアルゴリズムでは D フロントティア[8]の選択時において、可観測費の安いものを優先的に選択する。そのため、可観測費が安い故障伝播経路が活性化されるテストパターンが多く生成され、可観測費が高い故障伝播経路を活性化するテストパターンが生成されにくいという問題がある。それゆえ、故障が検出される外部出力や活性化される故障伝播経路に偏りが生じ、故障診断分解能が高いテスト集合が生成されるとはいえない。

本論文では、可観測費に代わる選択基準として各ファンアウトブランチに故障検出回数を記憶させ、用いる。故障シミュレーション時にファンアウトブランチで対象故障が検出されるたびに、そのファンアウトブランチの故障検出回数を1つ増加させる。テスト生成時には、故障検出回数の値が小さいファンアウトブランチに対応する D フロントティアを優先的に選択することで、検出される外部出力や故障伝播経路の偏りを削減する。生成されたテスト集合に対して誤り経路追跡を行い、被疑故障数を算出し、その有用性を検証する。第2章では故障診断における用語の定義を行い、第3章では、誤り経路追跡を説明し、第4章ではファンアウトブランチに対応する D フロントティアの選択方法とそのテスト生成法を提案する。第5章では、実験結果について示し、最後に第6章で今後の課題について述べる。

2. 諸定義

本章では、故障診断の基本的な用語の定義を行う。

(定義1：故障診断分解能)

故障診断分解能は、故障診断で存在すると推定された被疑故障数であり、故障診断の性能を示す指標として用いられる。被疑故障数が少ないほど、故障診断分解能が高い。

(定義2：可観測費)

信号線の内部状態を回路外部から観測する困難さを表す値を可観測費と呼ぶ。可観測費が高いほど回路の外部出力などの観測点でその信号線の値を観測することが困難になる。観測点から遠い信号線ほど観測が困難になり、可観測費が高くなる。

(定義3：ファンアウト)

ゲートの出力信号線が複数のゲートに接続されている場合、分枝元の信号線をファンアウトシステム、分枝先の信号線をファンアウトブランチと呼ぶ。

(定義4：パスセグメント活性化)

パスセグメント[8]は外部入力またはファンアウトブランチから外部出力またはファンアウトシステムまでの部分経路のことである。外部入力またはファンアウトブランチに伝搬している故障影響を出力先のファンアウトシステムまたは外部出力まで伝搬させるために、部分経路上のその故障影響が到達しない信号線に非制御値を割当てる操作をパスセグメント活性化と呼ぶ。

A Test Generator to Improve Diagnostics Resolution of Errorneous Paths Tracing

Yuya CHIDA, Toshinori HOSOKAWA and Kouzi YAMAZAKI

(定義 5：フェイル外部出力，パス外部出力)

被検査回路に対してテストパターンを印加しテストを実行したとき，期待値と異なる応答値を観測した外部出力をフェイル外部出力と呼ぶ．一方，期待値と同一の応答値を観測した外部出力をパス外部出力と呼ぶ．

(定義 6：フェイルパターン，パスパターン)

被検査回路に対してテストパターンを印加しテストを実行したとき，少なくとも 1 つの外部出力がフェイル外部出力であったとき，そのテストパターンをフェイルパターンと呼ぶ．一方，すべての外部出力がパス外部出力であったとき，そのテストパターンをパスパターンと呼ぶ．

3. 誤り経路追跡

本章では，誤り経路追跡について説明する．

誤り経路追跡はフェイルパターンを印加して論理シミュレーションを実行し，フェイル外部出力から外部入力に向かって行われる．各ゲートにおいて入力信号線が非制御値である場合，すべての入力信号線を選択し後方追跡を行う．一方，各ゲートにおいて入力信号線に制御値が存在する場合，制御値を持つ入力信号線を任意に 1 つ選択し後方追跡を行う．後方追跡により選択された信号線が被疑故障信号線となる．

図 1 にフェイルパターン(A,B,C)=(1,0,1)を印加した時の誤り経路追跡の回路例を示す．信号線 F に 1 縮退故障が発生した場合を想定すると，外部出力 H で異常値が観測されるため，外部出力 H の後方追跡が行われ，入力信号線 F が選択される．信号線 F の入力信号線は信号線 A と信号線 D であるが，AND ゲートの制御値 0 を持つ信号線は信号線 D であるため，信号線 D が選択される．その後，信号線 D の後方追跡を行い，信号線 B が選択される．信号線 D は外部入力であるため，(A,B,C)=(1,0,1)での誤り経路追跡は終了である．後方追跡によって選択された信号線 {B,D,F,H} が (A,B,C)=(1,0,1)のときの被疑故障信号線である．図 2 にフェイルパターン(A,B,C)=(1,0,0)を印加した時の誤り経路追跡の回路例を示す．外部出力 H と外部出力 J で異常値が観測されているため，外部出力 H と外部出力 J について後方追跡を行う．まず，外部出力 H の後方追跡を考える．図 1 と同様の後方追跡が行われ，信号線 {B,D,F,H} が選択される．次に，外部出力 J の後方追跡を行うと，信号線 {B,C,D,E,F,G,I,J} が選択される．次に，外部出力 H の後方追跡で求めた被疑故障信号線の集合 {B,D,F,H} と外部出力 J の後方追跡で求めた被疑故障信号線の集合 {B,C,D,E,F,G,I,J} の積集合を求める．得られた信号線 {B,D,F} が (A,B,C)=(1,0,0)を印加した時の被疑故障信号線である．

最後に，(A,B,C)=(1,0,1)で得られた被疑故障信号線の集合 {B,D,F,H} と (1,0,0)で得られた被疑故障信号線の集合 {B,D,F} の積集合を求める．求められた積集合 {B,D,F} が最終被疑故障信号線となる．

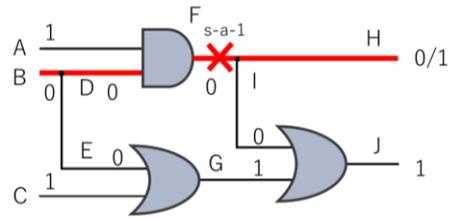


図 1. (A,B,C)=(1,0,1)を印加した回路例

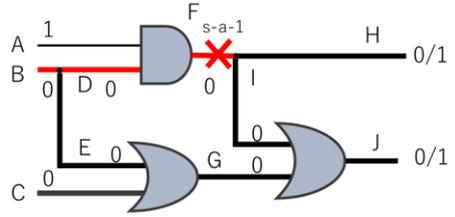


図 2. (A,B,C)=(1,0,0)を印加した回路例

```

01 Diagnostic_ATPG(f,c){
    //故障励起
02 IMP=fault_excitation(f,c);
    //故障の影響が外部出力に伝搬するまで
    //ループ
03 while(故障の影響が外部出力まで伝搬)
    {
    //含意操作
04 (Dfront,Unjust)=implication(c,IMP);
    //D フロントティアの選択
05 df=Dfrontier_selection(c);
    //パスセグメント活性化
06 IMP=path_sensitization(c,df);
    }
    //未正当化信号線が 0 になるまでループ
07 while(Unjust!=φ)
    {
    //正当化可能判定
08 IMP=justification(c,Unjust);
09 Unjust=implication(c,IMP);
    }
10 return t;
}

```

図 3. テスト生成の疑似ソースコード

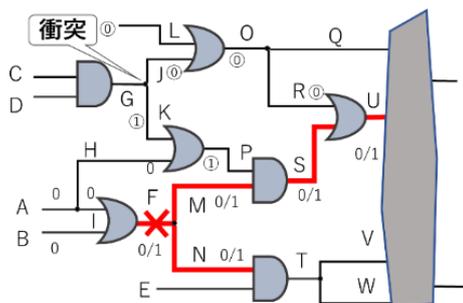


図 4. S を選択した場合の動作例

4. 診断分解能向上のためのテスト生成

本章では、ファンアウトブランチに対応する D フロントティアの選択方法とそのテスト生成法を提案する。

4.1. 全体フロー

図 3 に診断分解能向上のためのテスト生成の疑似コードを示す。まず、目標故障 f を挿入し故障励起を行う。次に、含意操作、D フロントティアの選択、パスセグメント活性化を目標故障 f の影響が外部出力に伝搬するまで繰り返す。最後に未正当化信号線の正当化処理を実行し、すべての未正当化信号線の正当化に成功した場合、テスト生成が行われる。

4.2. D フロントティアの選択

本節では図 3 の 5 行目のテスト生成時の D フロントティアの選択方法を示す。テスト生成は外部出力まで故障を伝搬させるために、パスセグメント活性化を行い、未正当化信号線に対して正当化処理を行う。ファンアウトブランチ S を通って外部出力で検出された故障数を $fd(S)$ と表記する。

図 4 の回路例を用いて動作説明を行う。信号線 F に 1 縮退故障を仮定する。故障励起により、信号線 F の正常値に 0、故障値に 1 を割当てる。含意操作により信号線 A, B, H, I は 0 に含意される。同様に信号線 M, N の正常値が 0 に含意される。故障の影響が外部出力まで伝搬していないため、D フロントティアの選択を行う。選択可能な D フロントティアを判定木に追加する。図 5 に D フロントティアの判定木を示す。故障シミュレーション時の故障検出回数が $fd(M) < fd(N)$ であると仮定すると、故障検出回数が少ない信号線 M に対応する D フロントティア S が選択され、パスセグメント活性化が行われる。故障検出回数の算出については第 3 節で説明する。パスセグメント活性化により、 $P=1, R=0$ を割当てる。含意操作により、図 4 の丸で囲まれた数値が割当てられる。このとき、信号線 G において、値の衝突が発生しているため、バックトラックを行う。

図 6 にバックトラックの例を示す。D フロントティアの選択により割当てられたすべての値を元に戻し、バックトラックにより図 5 の判定木から D フロントティア T を選択する。パスセグメント活性化により、信号線 E に 1 を割当てる。値の衝突が発生しないため、パスセグメント活性化は成功である。次に、故障の影響が外部出力まで伝搬しない場合、再度 D フロントティアの選択を行う。この処理を信号線 F の故障の影響が外部出力に伝搬するまで繰り返す。外部出力へ伝搬後、未正当化信号線の正当化処理が行われる。D フロントティアの選択を全通り行っても、対象故障が外部出力まで伝搬しない場合、その故障は冗長故障と判定される。

4.3. 故障シミュレーション時のファンアウトブランチの故障検出回数の算出

本節では、1 個のテストパターンを生成した後、そのテストパターンで故障シミュレーションを実行する際にファンアウトブランチに記憶される故障検出回数の算出について示す。

4.2 で生成されたテストパターンを印加し、対象故障を挿入して故障シミュレーションを実行する。外部出力に対象故障が伝搬された場合、未検出故障集合から対象故障を削除する。その後、ファンアウトブランチを通して外部出力に故障が伝搬した場合、そのファンアウトブランチの故障検出回数を 1 つ増加させる。

図 7 の回路を用いて動作例を説明する。対象故障が信号線 F の 0 縮退故障である場合に、 $\{A,B,C,D,E\}=\{1,1,0,1,1\}$ を印加し、故障シミュレーションを実行すると、外部出力 M に故障が伝搬する。信号線 F の 0 縮退故障が伝搬するファンアウトブランチは信号線 H, G, K, L であるが、 H, G, L に伝搬した故障の影響は外部出力まで伝搬しているのに対して、 K に伝搬した故障の影響は外部出力まで伝搬していない。したがって、 H, G, L の故障検出回数のみ 1 つ増加させる。



図 5. D フロントティアの判定木

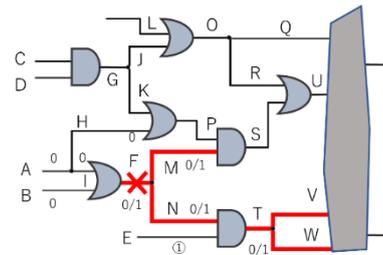


図 6. バックトラックで T を選択した場合の動作例

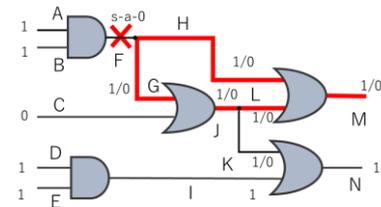


図 7. 故障シミュレーションの動作例

表 1. 従来手法と提案手法の結果

回路名	従来手法						提案手法					
	被疑故障数					テスト数	被疑故障数					テスト数
平均	標準偏差	最大	最小	最頻	平均		標準偏差	最大	最小	最頻		
b10	16.0168	13.3096	56	1	3	52	16.1708	13.3905	56	1	3	52
b11	26.0028	20.8015	133	1	10	72	25.5868	20.6874	133	1	10	73
b12	23.3269	27.2218	170	1	4	282	23.1453	26.9751	170	1	4	284
b13	10.1256	8.8063	55	1	4	74	10.1858	8.8866	55	1	4	69
b14	155.7845	176.9408	1730	1	32	1207	155.1665	174.1705	1730	1	32	1213
s13207	53.4575	72.2519	324	1	7	1027	53.279	71.8021	323	1	7	994
s15850	53.3395	85.9003	830	1	7	765	51.0805	79.0867	830	1	7	713



図 8. s15850 を対象とした従来手法により各外部出力で検出された故障



図 9. s15850 を対象とした提案手法により各外部出力で検出された故障

5. 実験結果

本章では、第 4 章で述べた可観測費に代わる選択基準として故障シミュレーション時に計算される故障検出回数を使用した場合の有用性を誤り経路追跡で算出された被疑故障数をもとに検証する。対象回路は ITC99 ベンチマーク回路の b10,b11,b12,b13,b14, ISCAS89 ベンチマーク回路の s13207,s15850 である。冗長故障を除くすべての縮退故障を対象として、テスト生成を行った。b10,b11,b12,b13 には冗長故障を除くすべての縮退故障を単一故障として 1 つの回路に 1 つずつ挿入し、b14,s13207,s15850 には冗長故障を除くすべての縮退故障からランダムに 2000 個の故障を単一故障として 1 つの回路に 1 つずつ挿入し、誤り経路追跡を行った。ファンアウトブランチの故障検出回数はテストパターンを 10 回生成するごとに更新した。

表 1 に選択基準に可観測費を用いた場合の結果と提案手法を用いた場合の結果の比較を示す。テスト数はテスト生成により生成されたテストパターン数である。表 1 より、b11,b12,b14, s13207,s15850 の回路において被疑故障数の削減に成功したことが示される。b10,b13 の回路は被疑故障数の削減に至らなかった。区別しやすい故障ばかりが外部出力に伝搬するテスト集合となってしまった可能性が考えられる。図 8 に s15850 を対象回路とした従来手法を用いた場合の各外部出力で検出される故障数のグラフ、図 9 に提案手法を用いた場合の各外部出力で検出される故障数のグラフを示す。縦軸が検出された故障数、横軸が外部出力名である。丸で囲まれた部分に着目すると提案手法により外部出力で検出された故障数が増加し、偏りが削減されたことが示される。

6. おわりに

本論文では、テスト生成時の D フロントティアの選択において、可観測費に代わる選択基準として故障シミュレーション時に算出される故障検出回数を使用した際の有用性を検証した。実験結果より、各外部出力で検出される故障の偏りの削減、b11,b12,b14, s13207,s15850 の回路において被疑故障数の削減に成功した。今後の課題として、故障検出回数の更新頻度

の変更や他回路での実験、故障検出回数以外の可観測費に代わる選択基準の提案などが挙げられる。

参考文献

- [1] H.Y.Chang, E.Manning and G.Metze: "Fault Diagnosis of Digital Systems", John Wiley & Sons, Inc.1970
- [2] E.Manning H.Y.Chang and G.Metze. Fault Diagnosis of Digital Systems. John Wiley & Sons, Inc., 1970.
- [3] M. Schulz, E. Trischler, and T. Serfert, "SOCRATES: A Highly Efficient Automatic Test Pattern Generation System," IEEE trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 7, No. 1, pp. 126-137, Jan. 1988.
- [4]W. Kunz, and D. Pradhan, "Recursive Learning: An Attractive Alternative to the Decision Tree for Test Generation in Digital Circuits," Proc. IEEE International Test Conference on Discover the New World of Test and Design, pp. 816-825, 1992.
- [5]M. Henftling, H.C.Wittmann, and K.J.Antreich, "A Single-Path-Oriented Fault Effect Propagation in Digital Circuits Considering Multiple-Path Sensitization," Proc. 1995 IEEE/ACM International Conference on Computer-Aided Design, pp. 304-309, 1995.
- [6]C. Wang, S. Reddy, I. Pomeranz, X. Lin, and J. Rajski, "Conflict driven techniques for improving deterministic test pattern generation," Proc. IEEE International Conference on Computer-Aided Design, pp. 87-93, 2002.
- [7]E. Gizdarski, and H. Fujiwara, "SPIRIT: A Highly Robust Combinational Test Generation Algorithm" IEEE trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 21, No. 12, pp. 1446-1558, Dec. 2002.
- [8] 藤原 秀雄, "デジタルシステムの設計とテスト", 工学図書株式会社, 2004. p163