検出外部出力を考慮した
動的テストパターン圧縮の効率化

日大生産工（院）〇秋山 祐介　　日大生産工　細川 利典
明大　　山崎 浩二

1. はじめに

近年，LSI の大規模化に伴い，それにかかるテストコストの増大が問題となっている．テストコストとテストパターン数は比例関係にあるためテストパターン数を削減することにより，テストコストの削減が期待できる．

小規模の回路では，ほぼ最小解のテスト集合を得るアルゴリズム 1)が提案されている．しかし大規模回路では計算量が大きく現実的な計算時間での適用は困難である．そこで大規模回路に適応しうる圧縮手法として，ドントケア故障シミュレーションを用いる動的圧縮手法 2)を提案した．しかし従来法では，圧縮効率が不十分であるという課題を持つ．

本稿では従来法に対して，目的故障を検出する外部出力を考慮し，圧縮バッファ内の既存のテストパターンと圧縮容易なテスト生成を行うことで，更なる総テストパターン数の削減を試みる．

2. 圧縮バッファを用いた動的圧縮の問題点

圧縮バッファを用いた動的圧縮では，テスト生成の過程の中でテストパターンが１つ生成されるたびに，圧縮バッファ内の既存のテストパターンと圧縮できるかどうかを調べ，圧縮が可能なら圧縮し，圧縮が不可能ならそのテストパターンは圧縮バッファに格納される．

最終的に生成されたテスト集合は，より圧縮された回数の多いテスト集合の方がテストパターン数の少ない優良なテスト集合であることが予想される．しかしながら，一般に目標故障の伝搬経路はテスト生成のアルゴリズムに依存される．ほとんどのテスト生成アルゴリズムは可観測費 3)が安い伝搬経路を選択する．そのため，同じ外部入力に論理値の割当要求が集中し，目的故障の検出が可能であり，かつ圧縮バッファ内のテストパターンと圧縮が可能であるテストパターンを生成できる可能性がありながら，圧縮不可能なテストパターンを生成してしまうことが多々起こりうる．そこで，動的圧縮を開始した時点で圧縮バッファ内のテストパターンとの圧縮を意識したテスト生成を行うことで，より圧縮率の向上が期待できる．

図１に一般のテスト生成の値の割り当ての例を示す．また，図 2 にそのときの圧縮バッファの状態を仮定して示す（バッファサイズ：5）．

図 1 では信号線bの 1 縮退故障を外部出力kに伝搬している．これにより得られるテストパターンt_Aは(0,0,1,X,X)である．

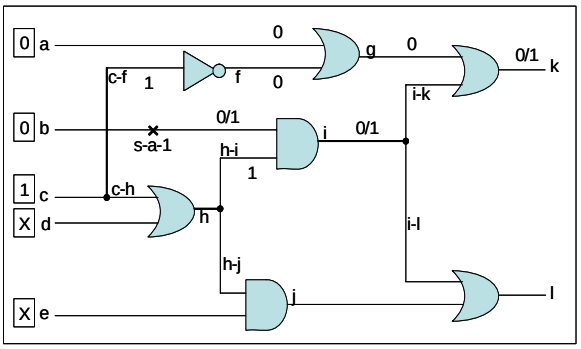

図 1 テスト生成例（t_A）

t_1	10X00
t_2	01X11
t_3	1X010
t_4	X0011
t_5	11101

圧縮バッファ
サイズ:5

図 2 圧縮バッファ例

Efficient Dynamic Test Compaction
that Considered the Detection Primary Outputs

Yusuke AKIYAMA, Toshinori HOSOKAWA, and Koji YAMAZAKI

しかし，テストパターンt_Aは圧縮バッファ内の既存のテストパターンとは圧縮が不可能である．

ここで図 3 に信号線 b の 1 縮退故障を外部出力 l に伝搬した場合を示す．

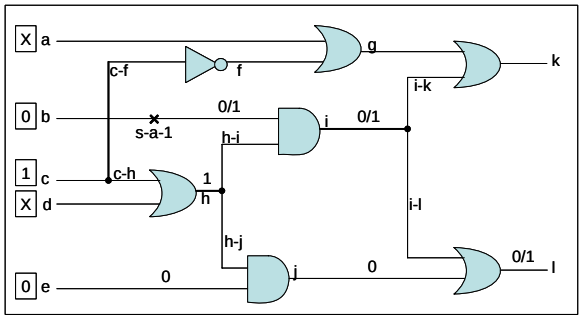

図 3 テスト生成例（t_B）

これにより得られるテストパターンt_Bは(X,0,1,X,0)である．t_Bは圧縮バッファ内のテストパターンt_1と圧縮が可能である．

以上の例より，生成されるテストパターンの値の割り当ては検出する外部出力に依存することがわかる．

3. 多重後方追跡を用いた仮想テストパターン生成

1 つの故障に対して検出する外部出力を変化させて，各々のテストをすべて生成することは多大な時間を要する．よって，提案手法では各外部入力の割当要求に着目することで仮想テストパターン を生成して，それを圧縮可能性の評価尺度として用いることにする．

3.1 多重後方追跡による値の割り当て

多重後方追跡 4）は，回路中の特定の内部信号線に特定の値（1 または 0）を割り当てるという目標を達成するためにどの外部入力に，どの程度の割当要求が来ているか求めるアルゴリズムである．

ここで多重後方追跡時に分岐元で値が衝突した際の処理について説明する．分岐元の場合は分岐先から異なる要求が来ることがありうる．この場合は，分岐元の割当要求回数の総和が大きい値を分岐元の値として目標に設定する．

図 4 に割り当て値の衝突の例を示す．分岐元 A には分岐先 B から 0 の割当要求が，分岐先 C からは 1 の割当要求が来ている．

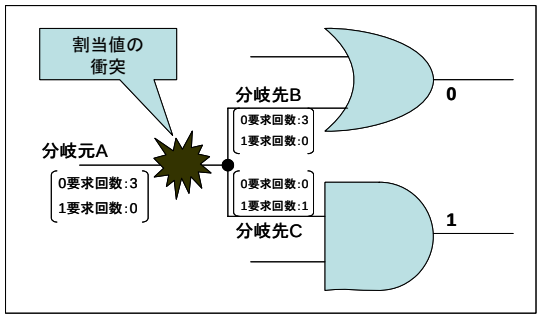

図 4 分岐元での割当値の衝突

このとき，B からの 0 割当要求は 3 回，C からの 1 割当要求は 1 回伝搬されてきているので割当要求回数の総和の大きい 0 の値が分岐元 A の目標値として設定される．

3.2 仮想テストパターン生成アルゴリズム

仮想テストパターン 5)とは目標に対する多重後方追跡の結果，外部入力で要求された論理値である．提案手法では 1 つの目標故障に対して到達可能な外部出力ごとに伝搬経路を設定して仮想テストパターンを生成する．

図 5 に仮想テストパターン生成のアルゴリズムを示す．前処理として各信号線には到達可能な外部出力の情報を持たせておく．

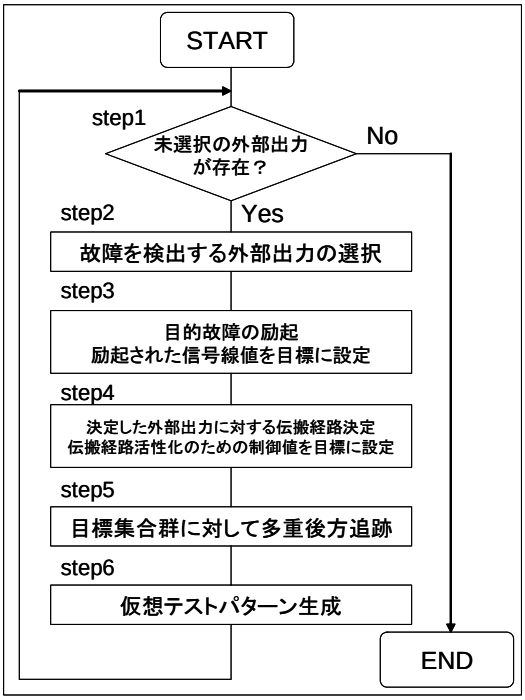

図 5 仮想テストパターン生成アルゴリズム

(step1)
未選択の外部出力の存在を確認．存在しないならば，処理を終了する．
(step2)
故障を検出する外部出力を決定．
(step3)
目標故障を励起して，その信号線の値を目標集合群に追加する．
(step4)
指定した外部出力までの故障伝搬経路を可観測費を基に決定し，その経路を活性化するための値を目標集合群に追加する．
(step5)
step2 と step3 で設定した目標集合群に対して多重後方追跡を行い，各信号線の割当要求値を計算する．
(step6)
外部入力における目標の結果から仮想テストパターンを生成する．

4. 動的圧縮テスト生成アルゴリズム

4.1 故障テーブルソート

提案手法では，テスト生成の前処理としてランダムパターンを用いて故障テーブルのソートを行う．図 6 に故障テーブルソートのアルゴリズムを示す．

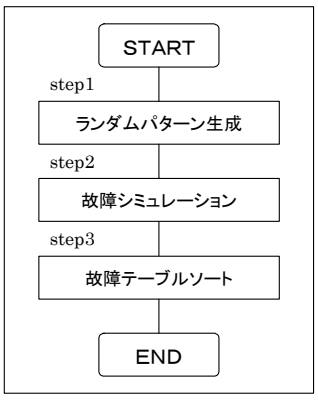

図 6 故障テーブルのソートアルゴリズム

(step1)
N 個のランダムパターンを生成する．(N の値はパラメータで与えられる．)
(step2)
ランダムパターンに対して故障シミュレーションを行うことで各故障が何回ずつ検出されたかをカウントする．
(step3)
step2 の結果から検出回数の低い故障を優先的にテスト生成の対象となるように故障テーブルをソートする．

検出回数の総数が低い故障は，擬似的に検出困難な故障とみなすことができる．検出容易な故障は目標故障ではないテストパターンでも検出されることが高確率でありうるため，検出困難な故障に対して優先的にテスト生成を行うことで，テストパターン数と CPU 時間の削減が期待できる．

4.2 テスト生成アルゴリズム

図 7 に検出外部出力を考慮した動的圧縮テスト生成アルゴリズムを示す．

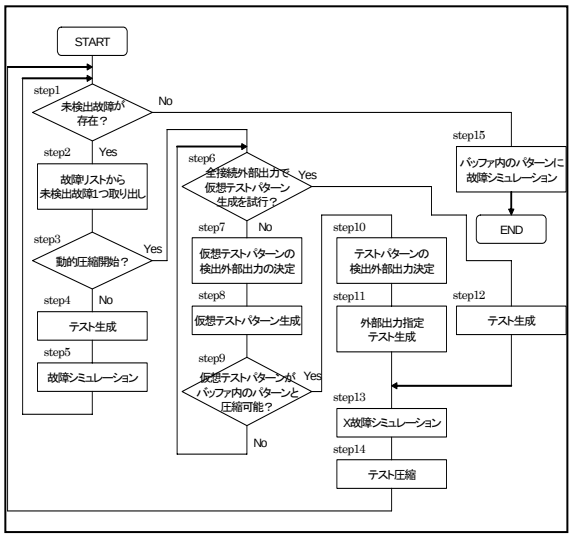

図 7 テスト生成アルゴリズム

(step1)
テスト生成は故障テーブル内の未検出故障がなくなるか，全ての故障に対してテスト生成を試みた時点で終了する．
(step2)
故障テーブルからテスト生成の対象となる未検出故障を 1 つ取り出す．
(step3)
故障検出効率を考慮し，一定の閾値を設けることで動的圧縮の開始を判断する．
(step4)
テスト生成し，パターンのドントケアはランダムで 0 または 1 の値が割当てる．
(step5)
生成されたパターンに故障シミュレーションを実行して故障テーブルを更新する．

(step6)
目標故障の信号線から到達可能なすべての外部出力に対して，仮想テストパターンを生成を試みたかを判断する．
(step7)
実選択な接続関係にある外部出力を１つ選択する．
(step8)
仮想テストパターンを生成する．
(step9)
生成した仮想テストパターンと圧縮バッファ内のパターンが圧縮可能か解析する．
(step10)
step9 の結果からテスト生成時に故障を検出する外部出力を決定する．
(step11)
故障を検出する外部出力を限定してテスト生成を行う．
(step12)
検出外部出力を限定しない一般のテスト生成を行う．
(step13)
ドントケアを含んだテストパターンに対して，ドントケア故障シミュレーションを行う．
(step14)
圧縮バッファに生成したテストパターンを格納する．もし圧縮が可能ならば圧縮し，新しいパターンにドントケア故障シミュレーションを実行する．圧縮バッファ内をパターンのドントケア数で降順にソートし，次の目標故障に対するテスト生成に移行する．また，圧縮が不可能な場合は生成されたパターンのドントケア故障シミュレーションとバッファ内をドントケア数で降順にソートした後に，圧縮バッファ内のパターン数がバッファサイズを超過していないか調べる．パターンがバッファから溢れた場合は最もドントケア数の少ないパターンを１つ取り出して，ドントケアがある場合は 1 または 0 の値をランダムで割当てて故障シミュレーションを実行する．このパターンを圧縮バッファから除去して，次の目標故障に対するテスト生成に移行する．
(step15)
圧縮バッファ内に残ったパターンに故障シミュレーションを行い，未検出故障を検出できるパターンはテスト集合に追加する．

5. 予備実験結果

今回は予備実験として，従来法で ITC'99 ベンチマーク回路に対して圧縮可能なテストパターンをどの程度生成しているのかを調査した．バッファサイズは 200，圧縮閾値 2) は 4 である．“circuit”は回路名，“#TP”はテスト集合のサイズ，“#ATPG_comp”は動的圧縮開始後のテスト生成回数，“#comp_pat”は圧縮可能だったテストパターン数を示す．また，“comp_rate”は圧縮率を示し，

comp_rate = #comp_pat / #ATPG_comp

で計算した．

表 1 圧縮不可能なテスト生成数

curcuit	#TP	#ATPG_comp	#comp_pat	comp_rate(%)
b14_C	1167	3058	1690	55.3
b15_C	827	2030	998	49.2
b17_C	1423	5763	4315	78.9
b20_C	1723	4062	2080	51.2
b21_C	1791	3953	1944	49.2
b22_C	1640	5345	3555	66.5

6. おわりに

本稿では目的故障の伝搬経路および，圧縮バッファ内の既存のテストパターンに着目したテスト生成を行うことでより圧縮効率を向上させる手法を提案した．

今後の課題として，本手法の実装と ITC'99 ベンチマーク回路による手法の評価が挙げられる．

参考文献

1)Y.Matsunaga, “MINT-An exact algorithm for finding minimum test sets,” *IEICE Trans. Fundamentals* vol. E76-A, pp1652-1658 (1993)

2)秋山祐介, “ドントケア故障シミュレーションを用いた動的テスト圧縮の効率化”, 第 39 回日本大学生産工学部学術講演会数理情報部会公演概要，pp.89-92 (2006)

3)R.Lisanke , F.Brglez , A.J.Degeus and D.Gregory. “Testability-driven random test-pattern generation”,*IEEE Transactions on Computer-Aided Designed* ,CAD-6,:1082 (1987)

4)H.Fujiwara and T.shimono,”on the Acceleration of Test Generation Algorithms,” *IEEF trans* , on Computers,Vol.C-32 No.12 , pp.1137-1144 (1983)

5)斉藤善洋, “テスト圧縮効率化のためのテストポイント挿入尺度”, 第 39 回日本大学生産工学部学術講演会数理情報部会公演概要，pp.89-92 (2006)