7-12

テスト圧縮指向ドントケア抽出法

日大生産工(院)　○若園　大洋
日大生産工　細川　利典
九大　吉村　正義

1 まえがき

近年VLSI(Very Large Scale Integrated circuits)の大規模化，複雑化に伴い，テストパターン数の増大、故障モデルの多様化が問題となっている．テストパターン数の増大を解決する手法としてテスト圧縮[1]が存在する．テストパターン中のドントケアを利用することによりテストパターン数を削減することが可能となる[2]．また，故障モデルの多様化により，縮退故障以外の故障モデルであるブリッジ故障[3]や遷移故障[4][6]の検出が重要となっている．これらの故障を検出するテスト生成は縮退故障テスト生成と比べ困難であり，また，それらの故障を検出するためにテストパターン数が増えてしまいテストコストの増大につながってしまう．そこで，テストパターン中に含まれるドントケアに値を再割り当てすることにより，テストパターン数を増やすことなく縮退故障以外の故障モデルの検出を向上させることが可能となる[7].

一般に生成されたテストパターンの入力値は全て0, 1に設定される．しかしながら，生成されたテストパターンの中には，逆の論理値に変更しても故障検出率が低下しない入力値が存在し，それらの論理値を抽出するドントケア抽出技術[2]が提案されている．

しかしながら, 文献[2]で提案されたドントケア抽出技術では特定のテストパターンにドントケアビットが偏っている傾向があり，特定の適用分野によってはドントケアの再割り当てによる効果が小さい場合がある[8]．特定のテストパターンにドントケアケアビットが偏る問題を解決する手法として，各テストパターンで検出する故障を均一化する手法[8]が提案されている．しかしながら，この手法ではテストパターンに対してはドントケアビットの分散を考慮しているが，外部入力に対してはケアビットの分散を考慮していない．例えば，適用分野としてテスト圧縮を考えた場合，特定の外部入力にケアビットが集中するとテスト圧縮効率を下げる可能性がある．圧縮率を向上させるためには外部入力に対してケアビットを分散させる必要がある．

本論文では，入力されたテスト集合に対しテストパターン及び，外部入力に対してケアビットの制御，また各テストパターンで検出する故障を制御することにより，テスト圧縮に適したドントケア抽出法を提案する. 第2章で従来のドントケア抽出技術について説明し, 第3章でテスト圧縮指向ドントケア抽出法を提案し, 第4章でISACAS'89, ITC'99ベンチマーク回路を用いてドントケア抽出率とテスト圧縮率を評価し, 第5章で結論と今後の課題について述べる.

2. ドントケア抽出技術

2. 1. ドントケア

ドントケアとは，テストパターンにおいて対象となる故障を検出するときに必要のない論理値のことである．その値は"0"もしくは"1"のどちらでもよい．ドントケアはテストパターン中に"X"または"x"と表記される.

2. 2. ドントケア抽出の定式化

本論文では，ATPG(Automatic Test Pattern Generator)などで生成されたテストパターン集合を扱う．テストパターンの外部入力値が全て0，1に特定されたテストパターン集合Tが与えられたとき，次の特性をもったドントケアを含むテストパターン集合T'を導出する.

(1)TはT'を被覆する（0から1，1から0の変換がない)
(2)TとT'の縮退故障検出率は等しい
(3)T'はできるだけ多くのドントケアを含む

以下に例を示す. 図1の回路に対して生成された表1(a)のテストパターン集合Tが与えられた時, 表1(b)のテストパターン集合T'は解の一つである. テストパターンt1は故障a/0,b/0,c/1を検出する．ここでs/vは信号線sのv縮退故障を意味する．a/0は，t1以外で検出できないためt1によって必ず検出されなければならない.一方でc/1は

A Test Compaction Oriented Don't Care Identification Method

Motohiro WAKAZONO, Toshinori HOSOKAWA, Masayoshi YOSHIMURA

必ずしもt1によって検出される必要はない．なぜなら，t3もc/1検出するからである．このためt1の外部入力cの信号値0はドントケアにできる．同様にt4における外部入力aの信号値もドントケアになる．このようにして表1(b)のテストパターン集合T'を求めることができる．

2．3．ドントケア抽出の基本アルゴリズム

与えられたテスト集合Tからテスト集合T'を得るまでの基本処理手順を図2に示す．ここでCは回路であり，Tはテスト集合である．まず，Step1で，テストパターンtiに対し，tiの必須故障集合を求める．次にStep2で，それらの必須故障を検出するようにtiでのシミュレーション結果を基に，外部入力の値を計算し，ti'値を決定する．t'は必須故障以外の故障を検出するかもしれないので，Step3でti'に対する故障シミュレーションを行う．Step1からStep3の処理により，初期のテストパターン集合T'が得られる．T'は必須故障を全て検出するテスト集合である．T'では未検出故障があるので，全ての故障が検出されるように，ti'のドントケアのいくつかを元のテストパターンtiの値に戻す．Step4で，T'では未検出だがtiでは検出できる故障を算出する．そしてStep5で算出された故障を検出するための論理値を計算し，ti"を最終的に決定する．Step6でti"に対し故障シミュレーションを行い，故障リストを更新する．

3．テスト圧縮指向ドントケア抽出

2章では従来のドントケア抽出法について説明した．本章では，本論文の手法であるテスト圧縮指向ドントケア抽出法について説明する．

3．1．全体のアルゴリズム

図3に本手法の全体のアルゴリズムを記す．

```
Procedure X-search(C,T)
Circuit C, Test Set T;
{
  for each test pattern ti in T{
    F=collect_essential_fault(ti);          step1
    ti'=find_value(F);                      step2
    fault_simulation(ti');                  step3
  }
  for each test pattern ti' in T'{
    G=collect_undetected_fault(ti');        step4
    ti''+=find_value(G);                    step5
    fault_simulation(ti");                  step6
  }
  return T'' composed of ti''
}
```

図2．従来手法処理手順

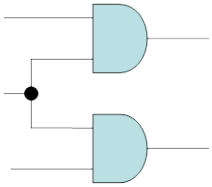
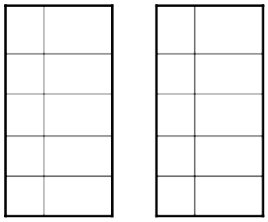


図1．回路例

表1．テストパターン集合



(Step1)

T に対して故障シミュレーションを実行することにより必須故障[1]を算出し，必須故障の検出のみを対象としドントケア抽出を行い，T'を得る．ここで必須故障とは一つのテストパターンでしか検出できない故障である．必須故障の検出に関して，必須故障を検出するテストパターン選択の余地がないので，はじめに必須故障のみを対象としドントケア抽出を行う．

(Step2)

T'に対し故障シミュレーションを行う．Step1で必須故障のみを対象とし，ドントケア抽出を行ったが，必須故障を検出するケアビットで偶発的に検出される故障が存在する．そのため，故障シミュレーションを行い T'で検出可能な故障を未検出故障リストから除外する．全てのテストパターン ti に対し故障シミュレーションを行うことにより，各故障の検出回数を算出する．

(Step3)

外部入力に対するケアビットを制御したドントケア抽出を行う．対象とする故障は未検出故障 UD のみである．故障をどのテストパターンで検出するか評価関数を用いて選択し，外部入力に対するケアビットを制御したテストパターンを算出する．詳細については 3.3 節で説明する．

3．2．外部入力に対するケアビットを制御したドントケア抽出

3．2．1．評価関数

外部入力に対するケアビットを制御したドントケア抽出で用いる評価関数について説明する．故障fiをどのテストパターンで検出するかを

```
Procedure distribution X-Identification(C,T)
Circuit C, Test Set T;
{
  T'=essential_Xidentification(T);                    Step1
  UD=fault_simulation(T');                            Step2
  T''=distribution X_identification for input(T',UD);  Step3
  return T'';
```

図3．全体のアルゴリズム

選択するときに用いられる．表 2 は，ドントケアを含むテストパターン集合 t”に対して各外部入力に含まれるケアビット数を表している．ここで x はドントケアを表し，c がケアビットを表しテストパターンは t1~t5, 外部入力は p1~p7 まで存在する．

本手法では以下の計算式を用いて各外部入力に対する評価関数を計算する．各外部入力に対する評価値は，式(1)のように求めることができる．ここで外部入力数は$N(1<j<N)$テストパターン数は$M(1<i<M)$である．また$b(t_i,p_j)$はテストパターンt_iの入力ピンp_jのビットがドントケアビットだった場合0，ケアビットだった場合0となる．

表 2 の外部入力 t_1 に対してコストを計算すると，外部入力 p_1 はケアビットが 2 つ存在するので $w(p_1)=2$ となる．これらの計算を全ての外部入力に対して計算を行う．

$$W(p_j)=\sum_{j=1}^{M} b(t_i,p_j) \quad \cdots\cdots (1)$$

3. 2. 2. 検出パターン選択

3.2.1 節では各ビットに対する評価値の計算について説明した，3.2.2 節では 3.2.1 節で算出した評価値を用いて，検出パターンの選択について説明する．必須故障ではない故障 f に対しドントケア抽出を行う場合，故障 f を検出するために必要なケアビットを，故障 f を検出する全てのパターン
に対して算出する．しかしながら，故障 f は一つのパターンで検出すれば十分であるため，どのテストパターンで検出するかを選択する必要がある．今回の手法では，各外部におけるケアビットを分散させることにより，テスト圧縮に適したドントケアを含むテストパターン集合を生成する．そこで 3.3 節で算出したコストを用いて故障 f を検出するテストパターンを選択する．故障 f はテストパターン t1,t2,t3 で検出できたと仮定する．そこで各テストパターンで故障 f を検出するために必要なケアビットは表 3 のようになる．ここで各テストパターンにおいて故障 f を検出するために必要な評価値を算出する．例えば t_1 で故障 f を検出する場合外部入力 p_1,p_2,p_3,p_6,p_7 をケアビットにする必要がある．ここで表 3 の各外部入力に対する評価値を用いて，t1 で故障 f を検出するときの評価値を求めると
t1=(p1)+(p2)+(p3)+(p6)+(p7)=2+0+4+2+1=9
となり，t1で故障fを検出するときにかかる評価値は9となる．同様に故障fをt3,t5で検出するときの評価値を求めると表4のようになり，それぞれ評価値は10,4となり，故障fはt5で検出するときが評価値が最小になるので，故障fはテストパターンt5で検出するようケアビットを決定する．

3. 2. 3. 外部入力に対するケアビットを制御したドントケア抽出のアルゴリズム

外部入力に対するケアビットを制御したドントケアスト集合を用いた．

表 2. 各パターンに含まれるケアビット

	p1	p2	p3	p4	p5	p6	p7
t1	x	x	c	c	x	c	x
t2	c	x	x	c	c	x	c
t3	x	x	c	c	x	x	x
t4	c	x	c	c	c	x	x
t5	x	x	c	c	x	c	x
	2	0	4	5	2	2	1

表 3. 故障 f を検出するために必要なケアビット

	p1	p2	p3	p4	p5	p6	p7
t1	c	c	c	x	x	c	c
t3	x	c	c	c	x	x	c
t5	c	c	x	x	x	c	x

表4. 各パターンで故障fを検出するために必要なコスト

	p1	p2	p3	p4	p5	p6	p7	
t1	c	c	c	x	x	c	c	9
t3	x	c	c	c	x	x	c	10
t5	c	c	x	x	x	c	x	4
	2	0	4	5	2	2	1	

抽出について説明する．図 4 にケアビット分布を制御したドントケア抽出のアルゴリズムを記す．

(step1)
未検出故障リス F より未検出故障 fi を選択する．fi は検出回数が最も少ない故障である．

(step2)
未検出故障 fi を検出するテストパターン集合 DT を T に対し故障シミュレーションを行い算出する．

(step3)
fi を検出するテストパターン集合 DT に対し，故障 fi を検出するために必要なケアビットを算出する．

(step4)
fi を検出するテストパターン集合 DT に対し評価値を計算する．

(step5)
fi を検出するテストパターン集合 DT の中で，step4 で算出した評価値が最小となるテストパターンを算出し，ケアビットを決定する．

(step6)
ti は未検出故障 fi に対しケアビットを決定したが，偶発的に検出できる故障が存在する場合があるので，ti に対し故障シミュレーションを行い，未検出故障リスト ND’を更新する．

4. 実験結果

本章では提案手法の性能を評価するために，提案手法を実装し，ドントケア抽出率を評価する。またドントケア抽出後のテスト集合に対しテスト圧縮[9][10]を実行することにより圧縮率を評価する．対象回路は ITC’99 ベンチマーク回路と ISCAS’89 ベンチマーク回路に対して実装し，初期テスト集合は Synopsys 社の TetraMAX

によって生成された縮退故障用のテスト集合を用いた．表 5 は各テストパターンにおけるドントケア抽出率を表している．提案手法の比較対象として，従来の x 抽出[4]を用いた．提案手法は従来手法と比較して，全ての回路においてドントケア抽出率が高くなる結果となった．本手法の，入力ピンのケアビットを制御したドントケア抽出は，入力ピンの評価値を元にドントケア抽出を行う．そのためケアビットが少ないテストパターンを選択する場合が多くなるので，ドントケア抽出率が高くなったと考えられる．

表6はTetraMaxで生成したテストパターン集合に対し，2つのドントケア抽出を適用したテストパターン集合に対し，テスト圧縮を適用したときのテストパターン数を表している．従来手法と提案手法を比較してみると，s35932を除く全ての回路において，提案手法が従来手法と比較してテストパターン数を削減することができた．s35932回路のみ，テスト圧縮率が低かった原因として，s35932回路の初期テストパターン数が元々少なく，検出パターンの選択において，あまり効果が得られなかったことが考えられる．

5. おわりに

本稿では，テスト圧縮指向ドントケア抽出法を提案，および評価実験を行った．従来手法と比較して，提案手法が一つの回路を除いた全ての回路においてテスト圧縮率が高いことがわかった．今後の予定として，さらなる圧縮率を指向のドントケア抽出の提案，圧縮以外のアプローチに対するドントケア抽出などがあげられる．

表5. 各手法におけるドントケア抽出率

回路名	×抽出率	
	従来の×抽出	提案手法
s13207	94.3	96.0
s15850	94.0	94.7
s35932	86.1	88.7
s38417	97.2	97.8
s38584	96.3	96.7
b14	82.1	82.7
b15	90.1	91.9
b20	87.0	87.6
b21	87.9	88.3
b22	91.2	91.8

表6. 各手法におけるテスト圧縮率

		圧縮後のテストパターン数	
回路名	初期テストパターン数	従来の×抽出	提案手法
s13207	587	281	266
s15850	566	167	145
s35932	81	38	39
s38417	1178	191	136
s38584	878	206	168
b14	1303	908	876
b15	895	524	492
b20	2155	1186	1101
b21	2316	1275	1202
b22	2759	1068	972

```
Procedure distribution XIdentification(C,T,T',F)
Circuit C,First Test Set T, Essential Fault Test
Set T', Un Detect Fault List F;
{
  for each fault in F{
    fi=collect_no_detect_fault(F);          step1
    DT=collect_detect_test_pattern(fi);  step2
    find_value(DT);                         step3
    caluclation_cost(DT);                   step4
    ti=sellection_low_cost_pattern(DT);  step5
    fault_simulation(ti);                   step6
  }
}
```

図4. ケアビット分布を制御したドントケア抽出

参考文献

[1]Niraj jha and Sadeep Gupta, “Testing of Digital Systems”, Cambridge university Press, 2002.

[2]K. Miyase, S. Kajihara ,“XID: Don’t Care Identification of Test Patterns for Combinational Circuits,” IEEE Trans.
Comuter-Aided Design of Integrated Circuits and Systems, Vol. 23, No. 2, pp. 321-326,Fed. 2004.

[3]松雄三 塩坂知子 山田輝彦 山崎浩二“CMOS 回路における短絡故障の一モデルとそのテスト生成方” 電子情報通信学会論文誌 D-I vol. J81-D-I No.6 pp.872-879 1998 年 6 月

[4]Sying-Jyan Wang and Yan-Ting Chen ,“Low Capture Power Test Generation for Launch-off-Capture Transition Test Based on Don’t-Care Filling”, Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on, 27-30 May 2007 ,pp3683-3686.

[5] Jeremy Lee, Mohammad Tehranipoor, "LS-TDF: Low-Switching Transition Delay Fault Pattern Generation," vts,pp.227-232, 26th IEEE VLSI Test Symposium (vts 2008), 2008.

[6]A.Krstic, and K.-T.Cheng, “Delay Fault Testing for VLSI Circuits,” Kluwer Academic Publishers, 1998.

[7]Kohei Miyase, Kenta Terashima, Seiji Kajihara, Xiaoqing Wen and Subhakar M. Reddy”On Improving Defect Coverage of Stuck-at Fault Tests” Proceedings of the 14th Asian Test Symposium(ATS ’05) PP.216~223.

[8]Masayuki Arai Satoshi Fukumoto and Kazuhiko Iwasaki and Takahisa Hiraide,”Test Data Compression Using TPG Reconstruction for BIST-Aided Scan Test,”IEEE 6th Workshop on RTL and High Level Testing, Harbin,China,July 20-21,200

[8]Kohei Miyase, Kenji Noda, Hideaki Ito, Kazumi Hatayama, Takashi Aikyo, Yuta Yamato, Hiroshi Furukawa, Xiaoqing Wen, Seiji Kajihara”Effective IR-Drop Reduction in At-Speed Scan Testing Using Distribution-Controlling X-Identification” 2008 IEEE/ACM International Conference on Computer-Aided Design pp52-58.

[9]D.Brelaz,”New methods to clor the vertices of a graph”, Communications of the ACM, 22, pp-251-256,1979

[10]Seiji Kajihara, Irith Pomeranz, Kozo Kinoshita and Sudhakar M.Reddy “Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits”, 30th ACMIEEE Design Automation Conference, pp102-106,1993