検出故障情報を用いたドントケア抽出法

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. はじめに

近年，半導体技術の急速な進歩は，VLSI の大規模化，高速化，微細化を可能にしてきた[2]．しかしながら，論理回路の大規模化はテスト用入力に用いるテストパターン数を増大させる．また，論理回路の微細化，高速化は縮退故障用の故障の仕組みを複雑化にし，従来のテストパターンではそのような複雑な故障を見逃してしまう可能性がある．以上の理由からテスト生成では，縮退故障モデル以外の遷移故障モデル[3]やブリッジ故障モデル[5]などを検出するような，より高品質なテストが要求されている．

一般に生成されたテストパターンの入力値は全て 0 又は 1 に特定される．しかしながら，生成されたテストパターンの中には，外部入力と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する．そのような入力値はドントケア[1]と見なすことができ，ドントケアを抽出する方法[1]が提案されている．ドントケアに値を再割当てすることで，テストパターン圧縮[7]故障以外の故障モデルの検出[4]，テスト時の消費電力の削減[3]などが実現可能である．しかしながら，従来のドントケア抽出では，特定のテストパターンにケアビットが集中し，圧縮などの効果が高くならない場合がある．そのような問題を解決するために，特定のテストパターンにケアビットが集中せず，各テストパターンにケアビットを分散させる必要がある．

本論文では，各テストパターンで検出を保障する故障の情報とテストパターン集合を入力とし，各テストパターン集合で検出が保証された故障の検出に関係しない外部入力をドントケアとして抽出する方法を提案する．また各テストパターンで，検出を保証する故障数が均一になるような検出故障情報を生成するアルゴリズムを提案する．ISACAS’85 ベンチマーク回路を用いて本提案手法の有効性を示す．

2. ドントケア抽出

2.1 問題の定式化

ドントケア抽出は 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
(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 のテストパターン集合 T が与えられた

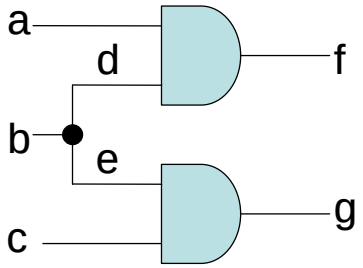

図 1 回路例

表 1 テストパターン集合

T

	abc
t_1	110
t_2	101
t_3	010
t_4	011
t_5	100

T’

	abc
t_1	11x
t_2	101
t_3	010
t_4	x11
t_5	xxx

Don’t Care Identification Method Using target Detected Fault Information
Motohiro WAKAZONO,Toshinori HOSOKAWA and,Masayoshi YOSHIMURA

時，表 1 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 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．このような故障を必須故障[6]と呼ぶ．一方で c/1 は必ずしも t1 によって検出される必要はない．なぜなら，t3 も c/1 を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．同様に t5 における検出故障 b/0,d/0 は両方必須故障ではないので，全ての入力線がドントケアになる．このようにして表１のテストパターン集合 T’を求めることができる．

2.2 ドントケア判定手法

与えられたテストパターン集合 T からドントケアを含むテストパターン集合 T’を得るまでの基本アルゴリズムを図 2 に示す．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，必須故障集合 F を検出するように ti の内部信号値を基に外部入力の値を計算し，ti’の値を決定する．ここで ti’は必須故障以外の故障を検出する可能性があるので，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’では検出されない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 の外部入力値に戻す．Step4 で，T’では未検出だが ti では検出できる故障集合 G を求める．そして Step5 で求めた故障集合 G 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’は未検出故障を検出する可能性があるので，故障シミュレーションを行う．

3. 検出故障情報を用いたドントケア抽出法

3.1 検出故障情報生成

検出故障情報とは，各テストパターンで検出を保証する信号線の縮退故障を示したものである．この検出故障情報を用いることにより，検出を保証した故障のみを検出するようなドントケア抽出が可能になる．本手法では各テストパターンで検出を保証する故障数を均一にするような，検出故障情報生成を行う．

各テストパターンで検出を保証する，検出故障

```
Procedure X-search(C,T)
 Circuit C; Test set T;
{
    T'=T''=φ
    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 X 抽出の基本アルゴリズム

情報生成の基本アルゴリズムを図 3 に示す．まず，Step1 で故障シミュレーションを行い各テストパターンで検出される故障と，各故障の検出回数を算出する．次に Step2 で故障 f が 1 回検出だった場合，故障 f は唯一つのテストパターンでのみしか検出されないので，故障 f と故障 f を検出するテストパターン t のペアを検出故障情報 TDF に追加する．Step3 で，故障 f が 1 回検出でなかった場合，故障 f は複数のテストパターンで検出可能である．そのため，故障 f を検出するテストパターンの中から，一番検出故障数が少ないテストパターン t を選択し，故障 f とテストパターン t のペアを検出故障情報 TDF に追加する．

検出故障情報生成の例を説明する．表 2 は図 1 の回路に対し表１のテストパターン集合 T を印加し，故障シミュレーションを行った結果の故障リストである．ここで t1 の a/0 は 1 回検出故障であるので，故障 a/0 と故障 a/0 を検出するテストパターン t1 のペアを TDF に追加する．この作業を全ての 1 回検出の故障に対して行い，1 回検出の故障全てを TDF に追加する．

表 3 は 1 回検出の故障の TDF である．これは 1 回検出のみの情報なので，まだ未検出な故障に対して処理を行う必要がある．この作業は，検出故障数が少ないテストパターンから選択して処理を行う．表 3 おいて，テストパターン t3 と t5 はまだ検出故障数が 0 である．そのため，t3 と t5 に対して，検出故障を割り当てる．テストパターン t3 において，まだ未検出な故障は a/1 と

c/1 で

```
Make_Target_Detect_Fault_Information(C,T,F)
{
    TDF=φ
    fault_simulation_no_drop(T);                    Step1
    for(n=1;n<MAX;n++){
        for each fault f∈F{
            if(n==1)
            TDF+=(detect test pattern t,f);    Step2
            else{
            t=detect test pattern seach(T);    Step3
            TDF+=(t,f);                        Step4
            }
        }
    }
     return TDF;
}
```

図 3 検出故障情報生成の基本アルゴリズム

ある. この場合どちらの故障を選択してもよいのだが, 今回は故障 a/1 を選択し, TDF に追加する. 次に t5 の検出故障数は 0 なので, t5 で検出可能な故障 d/1 を TDF に追加する. この処理を全ての故障を検出するようになるまで実行し, 最終的に得られる TDF は表 4 のようになる.

3.2 検出故障情報を用いた X 抽出

検出故障情報を用いたドントケア抽出法の基本アルゴリズムを図 4 に示す. まず Step1 で検出故障情報 TDF とテストパターン ti より対象とする故障 F を算出する. 次に Step2 で, 故障 F を検出するような値をテストパターン ti’に対して割り当て処理を行う. Step3 で, 値を割り当てたテストパターン ti’をテストパターン集合 T’に追加する.

検出故障情報を用いた X 抽出の例を図 1 の回路, 表 1 のテストパターン T, 表 4 の検出故障情報を用いて説明する. まずテストパターン t1 について考える. TDF よりテストパターン t1 で検出する故障は a/0, b/0, d/0 である. まず a/0 を TDF より得る. 次に故障 a/0 を検出するような入力線を決定する. a/0 を検出するために入力信号線 c の値は不要なので, 入力信号線 c の値をドントケアとする. さらに, 故障 b/0, b/0 を検出するように, ケアビットを決定する. 最終的にテストパターン t1 の入力信号線 c がドントケアとなる. 次にテストパターン t2 で検出する故障は e/1 と

表 2 故障リスト

t1	a/0,b/0,c/1,d/0
t2	b/1,d/1,e/1
t3	a/1,c/1
t4	a/1,b/0,c/0,d/0,e/0
t5	b/1,d/1

表 3 1 回検出の故障の TDF

t1	a/0
t2	e/1
t3	
t4	c/0,e/0
t5	

表 4 2 回検出の故障の TDF 決定

t1	a/0,b/0,d/0
t2	e/1,b/1
t3	a/1,c/1
t4	c/0,e/0
t5	d/1

```
Don’t_Care_Identification(C,T,TDF)
{
    T’=φ
    for each test pattern ti∈T{
        F=get_target_detected_fault(ti,TDF);    Step1
        ti’=find_value(F);                      Step2
        T’=T’U{ti’};                            Step3
    }
     return T’;
}
```

図 4 検出故障情報を用いたドントケア抽出の基本アルゴリズム

表 5 表４を用いて X 抽出後の
テストパターン集合

	abc
t1	11x
t2	x01
t3	010
t4	x11
t5	10x

b/1 である．これらの故障を検出するのに入力線 a の値は不要なので,入力線 a の値はドントケアとなる．次にテストパターン t3 は，TDF より a/1 と c/1 を検出する必要がある．これら二つの故障を検出するために不要な入力線の値はないので，テストパターン t3 にドントケアは存在しない．次にテストパターン t4 で検出する故障は c/0 と e/0 である．これらの故障を検出するのに入力線 a の値は不要なので入力線 a の値はドントケアとなる.次にテストパターン t5 検出する故障は d/1 である.この故障を検出するのに入力線 c 値は不要なので入力線 c 値はドントケアとなる．表 5 は終的に得られるテストパターン集合 T’である．表 1 のテストパターン集合 T’と比べると，ドントケアの偏りがなくなっているのが分かる．

4. 実験結果

実験環境は OS が WindowsXP,CPU が Inter(R) Core(TM)2 CPU4300 1.8GHz,メモリーが 1.18 GHz の計算機を用いた.テストパターンは TetraMAX により得られた縮退故障用のテストパターンに対し，ドントケア抽出を行った.評価回路は ISCAS’85 ベンチマーク回路を用いた.

検出故障情報を用いたドントケア抽出の予備実験結果を表 6 に示す.今回検出故障情報はドントケアを分散させるような検出故障情報を用いた．結果を見てみると，全体的にドントケアの分散は通常のドントケア抽出より少なくなっているが，ドントケア抽出率が少なくなっている．また c6288 回路ではドントケア抽出率が 0 パーセントとなった.これは回路の入力線が少ないためである.

表 6 予備実験結果

回路例	テストパターン数	X抽出率	X最大数	X最小数	X平均	分散
c499	65	1.7	8	0	0	407
c880	31	27.8	25	12	16	501
c1355	86	5.7	8	0	2	1249
c1908	117	7.8	9	0	2	91
c2670	61	66.8	187	130	155	11897
c3540	125	39.8	29	10	19	2581
c6288	24	0	0	0	0	0
c7552	95	45.2	125	48	93	10247

5. おわりに

本稿，検出故障情報を用いて，ドントケアのが一つのテストパターンに集中しないようなドントケア抽出をおこなった．今後の予定として，ドントケア抽出後のテストパターンを用いたテストパターン圧縮などが挙げられる．

参考文献

[1]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

[2]高松雄三 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

[3]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

[4] 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

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

[6]Jeroen Geuzebroek, Erik Jan Marinissen, Ananta Maihi, Andreas Glowatz, Friedrich Hapke “Embedded Multi-detect ATPG And Its Effect on the Detection of Unmodeled Defects”, INTERNATIONAL TEST CONFERENCE 2007 IEEE

[7]P. Goel, and B. C. Rosales,”Test Generation and Dynamic Compaction of Tests,”Digest of Papers 1979 Test Conf.,pp.189-192,Oct. 1979.