RC-004

パス遅延故障の過剰テストを削減するためのテストパタン生成法

A Scheme of Test Pattern Generation to Reduce Over-testing of Path Delay Faults

古本 圭† 吉川 祐樹†
Kei Furumoto Yuki Yoshikawa

1．まえがき

近年，VLSI の動作速度や性能はますます向上する中で，回路のタイミング不良の原因となる遅延故障のテストは重要な課題となっている．しかし，順序回路において遅延故障を検出するためのテストパタン生成は一般に難しく，膨大な時間を要しても高い故障検出率は得られない．そこでテストパタン生成を容易にするために，FF の値を外部から制御，観測できるスキャン設計が用いられる．スキャン設計された回路に対する遅延テストのアプローチとして，ブロードサイドテスト[1]が提案されている．

ブロードサイドテストは，検出困難な故障に対するテストを容易にすることで，高い故障検出率を達成することができる．しかし，スキャン動作によって通常動作では起こり得ない回路状態にも設定できてしまうため，回路の機能に影響しない故障(以下，機能的冗長故障と呼ぶ)を検出してしまう可能性がある．テストにおいて回路機能に影響しない機能的冗長故障を検出することは，検出すべきでない故障まで過剰に検出してしまうという意味で過剰テスト[2]とよばれており，歩留まりの低下を招く．

本研究では，ブロードサイドテストにおけるパス遅延故障の過剰テスト削減を目的としたテストパタン生成法を提案する．パス遅延故障に対するテストは，回路の微小遅延を検出できる高品質なテストである．ただし，パス遅延故障には機能的冗長故障が多く存在する．そのため過剰テストの削減は重要な課題である．

2．パス遅延故障のテスト

2.1 パス遅延故障

回路におけるパスとは，記憶素子であるフリップフロップ（FF）を始点とし AND，OR などの論理ゲートを通り FF を終点とする経路のことを言う．パス遅延故障とはパス上の累積遅延がある遅延値(一般にはクロック周期)を超える故障である．

パス遅延故障のテストパタンは，回路を初期化するための初期化パタン v1 と，パスの始点に信号遷移（0→1 もしくは 1→0 の変化）を起こしパスの終点までその信号遷移を伝搬する活性化パタン v2 からなる．例えば，図 1 の回路のパス(FF2，g6，g7，FF1)について，始点での立ち上がり遷移（0→1）が遅れるパス遅延故障 f が存在する場合を考える．ここでの記号 X はドントケアであり，その信号線の値は 0，1 どちらでも良いことを表現している．ある時刻 t に(PI1，PI2，PI3，PI4，FF1，FF2)の値がそれぞれ(X，X，X，X，X，0)で，時刻 t+1 で(X，X，0，1，0，1)とすると，時刻 t+1 で FF2 の出力に 0 から 1 の立ち上がりの遷移が起こり，その遷移はパスの終点の FF1 に取り込まれる．

†呉工業高等専門学校, Kure National College of Technology

図 2 に示すように，故障がない正常時には遷移が遅れることなく伝搬し，FF1 は 1 を取り込むが，故障時には遷移が遅れて伝搬し，FF1 に 0 が取り込まれる．

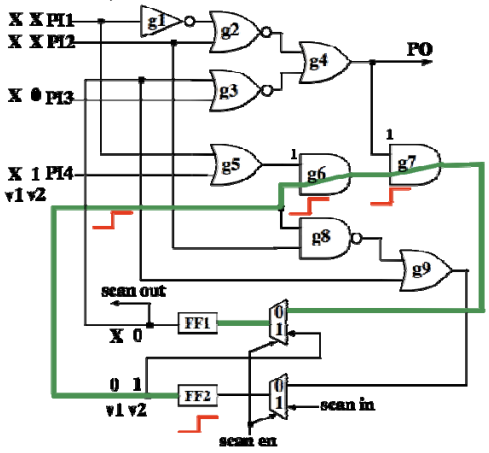

図 1　スキャン設計した順序回路

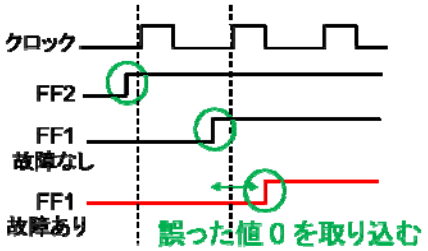

図 2　故障ありと故障なしの信号伝搬波形

2.2 ブロードサイドテスト[1]

順序回路は，記憶素子である FF の値を外部から直接制御することや観測することはできない．そこで回路のテストを容易にするための 1 つの方法として，図 1 に示すようなスキャン設計がある．スキャン設計を行った回路では，スキャン信号（scanenable）を 1 にすることでスキャンインから各 FF へ値を設定することができる．また，スキャンアウトでは各 FF の値を観測することができる．このようなスキャン設計された回路を利用した遅延故障のテスト方法としてブロードサイドテストが提案されている．先のパス(FF2，g6，g7，FF1)について，始点の遷移が立ち上がりのパス遅延故障 f を例にブロードサイドテストの説明を行う．図 3 は図 1 の順序回路を 2 時刻展開した回路を示している．故障 f に対するブロードサイドテストでは，時刻 t の(FF1，FF2)に対する初期化パタン(X，0)をスキャンインからスキャンシフトによって設定する．

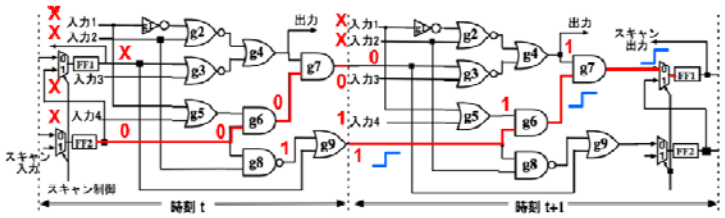

図 3　ブロードサイドテストの例

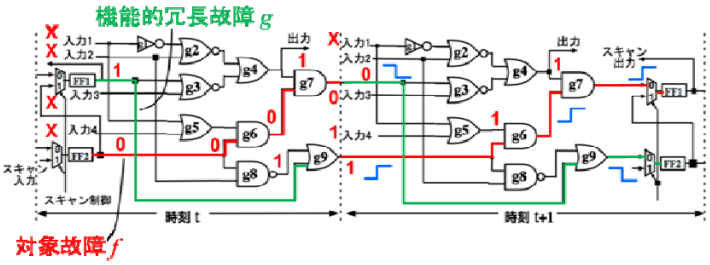

図 4　過剰テストの例

この例では，初期化パタンは(X，X，X，X，X，0)となる．時刻 t+1 の FF1，FF2 の活性化パタンは，初期化パタンの出力（時刻 t のゲート g7 と g9 の出力 0 と 1）を利用する．この例では，活性化パタンは(X，X，0，1，0，1)となる．そして，時刻 t+1 で FF1 と FF2 に取り込まれた値をスキャンシフトすることで外部へ出力する．これにより故障 f の影響を観測することができ，故障を検出することができる．この初期化パタンと活性パタンのペア(v1，v2)を故障 f のテストパタンという．

3．過剰テスト

回路にはたとえ存在したとしてもその回路動作に影響しない故障が存在する．このような故障を機能的冗長故障という．機能的冗長故障は通常動作の範囲では回路の出力に影響しないため，故障として検出されることはない．しかし，スキャン設計などのテスト容易化設計を行った回路では，通常動作では起こらない回路状態に設定できるため，回路動作に影響しない機能的冗長故障を検出してしまう可能性がある．これを過剰テストという．

例えば，図1におけるパス(FF1，g9，FF2)上のパス遅延故障gを機能的冗長故障とする．故障fを見つけるためのテストパタンv1=(X，X，X，X，1，0)，v2=(X，X，0，1，0，1)でブロードサイドテストを行うとき，同時に故障gも検出される．つまり，この回路において故障gは存在しても正常に動作するにも関わらず，gを検出することによりこのチップは不良品として判定される．

4．過剰テスト削減のためのテストパタン生成

4.1　テストパタンと故障検出の関係

ある故障を検出するためのテストパタンは，その故障だけではなく他の故障も検出する可能性がある．また，故障を検出するテストパタンは 1 つではなく複数存在する．

例えば表 1 に示すように，各テストパタン t1,t2,..,t6 とそれぞれの故障が検出する故障の関係があるとする．この表ではテスト対象故障集合を Ft={f1，f2，f3，f4，f5，f6}とし，機能的冗長故障の集合を Fu={f7，f8，f9，f10} とする．ただしこれ以降，機能的冗長故障は非テスト対象故障と表現する．ここで，テスト対象故障集合 Ft に属する全ての故障を検出するテストパタン集合 T1={t1，t3，t6}と T2={t1，t4，t5} を考える．それぞれのパタン集合 T1 と T2 が検出する非テスト対象故障 Fu は以下のようになる．集合 T1 が検出する非テスト対象故障は f7，f8，f9，f10 の 4 つであるのに対し，集合 T2 は f7，f8 の 2 つのみを検出する．

このように，故障集合の作り方によって非テスト対象故障の検出数が異なることから，個々のテストパタンの検出故障だけを考えるのではなく，テスト集合全体でどの非テスト対象故障を検出するかを考慮する必要がある．

表 1　各パタンの故障検出の関係

	テスト対象故障Ft						非テスト対象故障Fu			
	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10
t1	○	○					○	○		
t2		○		○		○			○	
t3			○	○				○		○
t4			○	○		○		○		
t5					○		○	○		
t6					○	○			○	○

4.2　過剰テスト削減のためのテストパタン生成

以下に，対象とする問題を示す．

問題

入力：対象回路 C，回路 C におけるテスト対故障集合 Ft，非テスト対象故障集合 Fu．

出力：対象故障集合 Ft の中で検出可能な故障を全て検出するテストパタン集合 T．

最適化目標：非テスト対象故障集合の検出故障数最小化．

この問題に対するテストパタン生成のアプローチとして，大きく 2 つの方法が考えられる．1 つは，各故障に対してパタン生成を行う際，必須割当てをどのように入力側へトレースするかに着目し，ドントケアを含むパタンを生成するときに，非テスト対象故障の検出数が少ないパタンを生成するためのアプローチが考えられる．2 つ目は，既に生成されたドントケアを含むパタン集合について，非テスト対象故障の検出数を最小にするためのドントケアへの 0,1 の割当て方法である．この論文ではまず後者のドントケア割当てに着目し，非テスト対象故障の検出数を最小にする手法を提案する．

以下に，生成したテストパタン集合のドントケアの埋め方について，非テスト対象故障の検出数を最小化するための戦略を述べる．

①各パタンはできるだけ非対象故障の検出数|Fud|が少なく，かつ対象故障の検出数|Ftd|が多くなるようにドントケアに0，1を割り当てる．

②あるパタンで検出された故障f∈{Ft∪Fu}は，それ以降のパタン生成時にドントケア割り当ての尺度計算から除くようにする．

1つ目の戦略は，各パタンが検出する非対象故障Fuの検出数削減だけではなく，全体のテストパタン数を減らすために各パタンのFtの検出数を多くすることに着目している．各パタンが検出する対象故障数が多くなれば，テストパタ

ン集合Tのパタン数が少なくなり，結果として非対象故障の検出数を減らすことができると考えられる．

一方，2つ目の戦略は，あるパタンで１度でも検出された故障は，それ以外のパタンによる検出／非検出にかかわらず全体のパタン集合Tで検出されることに着目している．つまり，その時点でまた未検出のFuの故障を非検出にすること，未検出のFtの故障を検出することを重視した戦略である．

提案するテストパタン生成アルゴリズムの概要を以下に示す．

ステップ1. Ft=φならば手続きは終了し，そうでない場合，テスト対象故障集合Ftから故障fを1つ選択する．

ステップ2. 選択した故障fに対してドントケアを埋めていないテストパタンtを生成する．

ステップ3. パタンtのドントケアへの割り当てを評価する尺度にしたがって，ドントケアを埋めたパタンt ’ を生成する．

ステップ4.

生成したパタンt’をTに追加し，そのパタンが検出するテスト対象故障と非テスト対象故障を故障シミュレーションで求め，検出される故障をFtとFuから取り除き，ステップ1.へ移る．

ステップ1では，テスト対象故障集合Ftからテストパタン生成の対象故障fを１つ選択する．このとき，Ftがφであれば全てのテスト対象故障に対してパタン生成が終了したことを意味するため手続きを終了する．次にステップ2では，選択した故障fに対してドントケアを埋めていない状態のテストパタンtを生成する．ステップ3では，生成したパタンtのドントケアへの割り当てを決めるために，できるだけ非テスト対象故障の検出数|Fud|が少なく，かつテスト対象故障の検出数|Ftd|が多くなるドントケア割り当てのための尺度を計算し，尺度にしたがってtのドントケアに0または1を割り当てたt ’ を生成する．この尺度は，上記で述べた戦略の１つ目に対応する．具体的な尺度の説明は後ほど述べる．ステップ４は，上述の戦略の２つ目に対応する．

ステップ3で用いる，できるだけ非テスト対象故障の検出数|Fud|が少なく，かつテスト対象故障の検出数|Ftd|が多くなるようにドントケアに0，1を割り当てるための尺度の１つとして，ここでは，生成したパタンt’の非テスト対象故障の検出数|Fud(t’)|とテスト対象故障の検出数|Ftd(t’)|のバランスをパラメータαで調節する評価式を提案する．この評価式をステップ3に埋め込んだ時の具体的なヒューリスティックアルゴリズムを図5に示す．

この手続きではまずテスト対象故障集合Ftからある故障fを選択し，fに対してテストパタンtを生成する．それから，生成したパタンtのドントケアをランダムに埋めたN個のパタンt’1，t’2，．．．，t’Nを生成し，以下の評価式で評価値を計算する．

$$e(t'i) = \alpha \times |Fud(t'i)| - (1-\alpha) \times |Ftd(t'i)|$$

この評価式はαを係数とし，ある対象故障fに対して生成したパタンtのドントケアをランダムに埋めたパタンt’i(i=0，1，．．．，N)で検出される非テスト対象故障数|Fud(t’i)|と，検出されるテスト対象故障数|Ftd(t’i)|から計算する．検出される非テスト対象故障数|Fud(t’i)|が少なく，また検出されるテスト対象故障数|Ftd(t’i)|が多いほど評価値 e は小さくなる．また，係数 α は 0≤ α≤1であり，α が大きいときはパタン t’i で検出される非テスト対象故障数|Fud|の重みが大きくなり，|Fud|が少ない方が評価値 e はより小さくなる．つまり，|Fud|の検出数が少ないパタンを優先して選択する．一方，αが小さいときは，パタン t’i で検出されるテスト対象故障数|Ftd|の重みが大きくなり，|Ftd|が多い方が評価値 e はより小さくなる．つまり，|Ftd|の検出数が多いパタンを優先して選択する．このように，αによって|Fud|の検出数を減らすことを優先するのか，|Ftd|の検出数を増やすことを優先するのかを調整する．ここではα=0.5としている．

```
# Define N 3
# Define α 0.5
# Define evel(Fa, Fb) (α*|Fa|-(1-α)*|Fb|)
tpg(Ft,Fu) {
// Ft is a set of faults to be tested
// Fu is a set of faults to be undetected

  T = φ;
  While(|Ft| = φ){
   f = target_from(Ft); // Ft = Ft - {f}
   t = test_generation_for(f);
   // if f is redundant, t will be nil
   if(t != nil){
     tb = nil; e = |Fu|;
     Ftb = Fub = φ;
     for (i=0; i<N; i++ ){
           tt = randomfill(t)
        // Xs in t are specified randomly
           (Ftd, Fud) = fault_sim_by_for(tt, Ft, Fu);
        // Ftd and Fud are sets of faults detected
       // by tt in Ft and Fu, respectively.
           v = eval(Fud, Ftd);
           if(v < e) {
              tb = tt;  e = v;
              Ftb = Ftd; Fub = Fud;
           }
     }
     T = T ∪ {tb};
     Ft = Ft - Ftb; Fu = Fu - Fub;
     }else{
        Ft = Ft - {f};
     }
  }
  return T;
}
```

図５　提案するヒューリスティックアルゴリズム

続いて，ドントケアをランダムに埋めて生成したN個のパタンの中からe(t’i)が最小となるものを選択し，それを対象故障fのテストパタンとして記憶し，故障シミュレーションにより各故障集合Ft，Fuから検出されるものを取り除く．これにより，今後はすでに検出されたものを取り除いたFtとFuに対して，パタンt’で検出される非テスト対象故障数|Fud(t’)|と，検出されるテスト対象故障数

|Ftd(t')|を故障シミュレーションで求めて評価値eを求めるため，未検出の非テスト対象故障をより非検出に，未検出のテスト対象故障をより検出するようになる．|Fud(t')|が少ないということは最適化目標であるFuの検出数最小に直接関係する．また|Ftd(t')|が多いことは生成するパタン数|T|を減らすことになり，結果として非対象故障の検出数を少なくできる．

上記のアルゴリズムの実行例を示す．図3を用いて説明する．まず，テスト対象故障集合Ftと，非テスト対象故障集合Fuが与えられたとき，Ftの中からある故障fを選択する．故障fに対するテストパタンtを生成する．生成されたパタンtは初期化ベクトルが(PI1，PI2，PI3，PI4，FF1，FF2)=(X，X，X，X，X，0)，活性化ベクトルが(PI1，PI2，PI3，PI4，FF1，FF2)=(X，X，0，1，0，1)である．このパタンtのドントケアをランダムに埋めたパタンをN個生成する．ここでは，Nは3とする．3つのパタンt1，t2，t3はそれぞれ，t1の初期化ベクトルが(PI1，PI2，PI3，PI4，FF1，FF2)=(0，1，1，0，1，0)，活性化ベクトルは(PI1，PI2，PI3，PI4，FF1，FF2)=(0，0，0，1，0，1)，t2の初期化ベクトルが(PI1，PI2，PI3，PI4，FF1，FF2)=(0，1，1，0，1，0)，活性化ベクトルは(PI1，PI2，PI3，PI4，FF1，FF2)=(0，1，0，1，0，1)，t3の初期化ベクトルが(PI1，PI2，PI3，PI4，FF1，FF2)=(0，0，351，0，1，0)，活性化ベクトルは(PI1，PI2，PI3，PI4，FF1，FF2)=(0，0，0，1，0，1)である．これらのパタンが検出するテスト対象故障数と非テスト対象故障数を故障シミュレーションで求める．パタンごとの検出するテスト対象故障数と非テスト対象故障数は，t1の|Ftd(t1)|は4，|Fud(t1)|は1，t2の|Ftd(t1)|は4，|Fud(t1)|は2，t3の|Ftd(t1)|は4，|Fud(t1)|は0となる．それぞれのパタンの検出する故障から評価式を用いて，それぞれのパタンの評価値eを求める．ここでは，αは0.5とする．3つのパタンt1，t2，t3について評価値eを求める．まずt1の評価値eは，以下のように-1.5となる．

$$e(t1)=\alpha\times|Fud(t1)|-(1-\alpha)\times|Ftd(t1)|$$
$$=0.5\times1-(1-\alpha)\times4$$
$$=0.5-2=-1.5$$

同様にして，t2の評価値eは−1，t3の評価値eは−2となる．この３つの中で評価値が最も小さいものを選択するため，この例の場合，t3が選択される．パタンt3によって検出される故障をテスト対象故障集合と非テスト対象故障集合から取り除き，これをテスト対象故障集合Ftがφになるまで繰り返す．

4．実験結果

この評価実験では，ベンチマーク回路としてPaulin，Lwf，Tseng，Jwf の4つを対象とし，どのようなαのときに非テスト対象故障の検出数が少なくなるのかを調べるために，α=0，0.25，0.5，0.75，1 のそれぞれについて実験を行った．また，与えられる故障集合によって，同じ回路でも検出するFuの故障数|Fud|が最小となるα が同じなのかを調べるために，各回路についてそれぞれ3種類の故障集合を用いて実験を行った．

実験結果の表2 より，各パタンについて，非テスト対象故障の検出数を最小（α=1）にするだけでは，全体の非対象故障の検出数|Fud| は最小にならない．また，各パタンについて，対象故障の検出数を最大（α=0）にするだけでも，やはり全体の非対象故障の検出数|Fud|は最小にならない．これは，非対象故障の検出数を最小（α=1）にすると1パタン当たりの非対象故障の検出数は減少するが，テスト対象故障に対するテスト集合が増加するため，テスト集合全体では非対象故障の検出数が増加してしまったものと考えられる．一方，対象故障の検出数を最大（α=0）にする場合は，テスト集合のパタン数自体は減少することで，非対象故障の検出数を抑えようとしているが，非対象故障の検出数のことを考えていないため，非対象故障の検出数が増加してしまったものと考えられる．

表2　実験結果

回路	対象故障 |Ft|	非対象故障 |Fu|	非対象故障の検出数 |Fud|				
			α=0	α=0.25	α=0.5	α=0.75	α=1
Paulin	532	1000	187	183	177	167	173
	532	532	167	165	108	108	135
	1000	1998	936	918	888	889	906
Lwf	1458	2918	1821	1721	1756	1777	1793
	1458	1458	605	597	588	598	605
	2918	1458	1403	1359	1353	1341	1366
Tseng	900	1786	239	185	184	183	188
	1343	1343	492	451	450	458	469
	1786	900	343	345	345	339	347
Jwf	1266	2500	440	439	439	438	438
	1883	1883	1292	1273	1296	1304	1314
	2500	1266	897	859	867	867	871

以上のことより,各パタンの非対象故障の検出数と対象故障の検出数のバランス(α)を考えることで，全体の非対象故障の検出数を減らせることが分かる．今回の実験から，α=0.5～0.75 で Fu の検出数が最小になることが多いことが分かる．このことから，非対象故障の検出数を減らすことに重みを置きながら，対象故障の検出数を増やすことにも配慮する必要がある．また回路の特性などから前もって最適なαをある程度予測する手法が必要であり，これは今後の課題とする．

5．まとめ

本研究では，ブロードサイドテストにおけるパス遅延故障の過剰テスト削減を指向したテストパタン生成法の１つとして，すでに生成されたテストパタンのドントケアの埋め方に着目し，ドントケアをランダムに埋めて，テスト集合全体で非対象故障の検出数Fudが最小となるようなパタンを選択するための評価式を提案し，実験を行った．実験により，同じ回路でも故障集合が異なると非対象故障の検出数が最小となるαがばらつくことが分かった．

今後の課題としては,より大規模な回路での実験や,回路ごとに最適な係数αを推測する尺度の提案などが挙げられる.また，パタン生成時に必須割当てをどのように入力側へトレースするかという問題にも今後取り組む.

参考文献

[1] J.Savirand S.Patel “Broad-side delay test, ”IEEE Trans. on CAD, Vol.13，No.8, pp.1057-1064, Aug.,1994.
[2] J.Rearick “Too much delay fault coverage is a bad thing, ”Proc. IEEE International Test Conf., pp.624-632, 2001.
[3] I.PomeranzandS. M.Reddy, “On generationing tests that avoid the detection of redundant faults in synchronous sequential circuits with fulls can, ”IEEE Trans. on Comp, Vol.55, No.4, pp.491-495, April, 2006.
[4]藤原秀雄，ディジタルシステムの設計とテスト，工学図書株式会社，ISBN4-7692-0459-0，2004.