

ドントケア抽出を用いたスキュードロード方式による 遷移故障検出率の最大化

日大生産工(院) ○濱崎 和光
日大生産工 細川 利典

1 まえがき

近年, VLSIの大規模化に伴いそれに対するテスト生成時間やそのテストコストが増大している[1]. その増大するテストコストを抑えるために効率のよいテストパターンの生成や回路のテスト容易化設計が行われてきた. また, VLSIの縮退故障検出だけでは不十分だとされており, 縮退故障の検出に加えて信号線の遅延を検出する遷移故障や, 複数の信号線上に対して発生するブリッジ故障の検出が必要となってきている.

生成されたテストパターンに対し, 故障検出率を落とさないでテストパターン中にドントケアを発生させるドントケア抽出技術が提案されている[5]. ドントケア抽出技術を適用することで生成されたドントケアを含んだテストパターンに対して, テストパターン数の削減や単一縮退故障以外の故障モデルの検出, 消費電力の削減などの新たな特性を持たせることが可能になると考えられる.

本稿では, 縮退故障用テストパターンを用いてスキュードロード方式で遷移故障の検出を行い, さらにドントケア抽出を行って抽出されたドントケアを再割り当てすることにより遅延故障検出率の最大化する方法を提案する.

第2章でドントケア抽出技術について説明し, 第3章でスキュードロード方式による遷移故障検出技術について説明する. 第4章で遷移故障の検出を指向したXの再割り当て法を提案し, 第5章で縮退故障用テストパターンによる遷移故障検出率の評価を行う. 最後に第6章で今後の課題について述べる.

2 ドントケア抽出

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 ドントケア抽出問題の定式化

本稿では, 与えられたテストパターンから, 潜在的に存在するドントケアを抽出する問題を以下のように定式化する.

与えられたテストパターン T に対して次のような特性を持ったドントケアを含むテストパターン 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

以下にそのドントケア抽出の例を示す

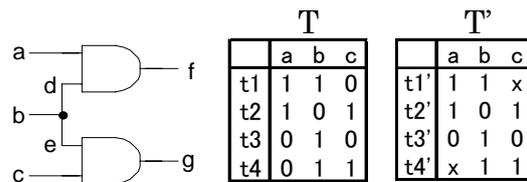


図1 : 回路図とそのテストパターン

図1の回路に対してこのようなテストパターン集合 T が与えられたとき, テストパターン集合 T' はドントケア抽出問題の解の1つである. テストパターン $t1$ は, 故障 $a/0$, $b/0$, そして $c/1$ 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 $t1$ 以外で検出することができないため $t1$ は必須故障[3]である. しかし, $c/1$ は $t3$ でも検出可能であるため $t1$ で必ず検出される必要はない. そのために, $t1$ の入力 c の値 0 はドントケアにすることができる. 同様に $t4$ における外部入力 a の値もドントケアにすることができる. このようにして図1に示すテストパターン集合 T' を求めることができる.

Maximization of transition breakdown detection rate by skewed-load method to use Don't Care extraction.

Kazumitsu HAMASAKI, Toshinori HOSOKAWA

3 遷移故障検出

3.1 スキュードロード方式

遷移故障は2パターンで検出できる。第1パターンを初期パターン、第2パターンを遷移パターンと呼ぶ。本稿では遷移故障検出の方法としてスキュードロード方式[4][5]を用いている。スキュードロード方式とは、シフト動作により初期パターンと遷移パターンをSFFに設定し、故障の影響をキャプチャ動作でSFFに取り込んで検出するテスト方式である。以下にその例を示す。

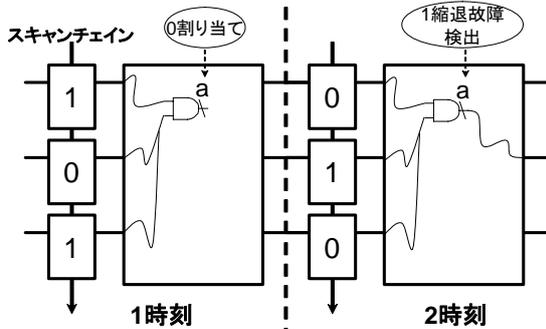


図3：スキュードロード方式

図3はスキュードロード方式の例である。信号線aの立ち上がり遷移故障の検出を行う。

まずシフト動作によりスキャンチェーンに1時刻目のテストパターンである初期パターンを印加する。この場合ではスキャンチェーンに(1,0,1)の順で入力する。この状態で1ビットのみシフトを行う。2時刻目には1ビットシフト後の値が各SFFに印加されておりこの状態では新たに最もスキャンチェーンに近いSFFに印加される値を"0"とすると(0,1,0)と表すことができる。

1時刻目で信号線aの値が"0"になり、かつ2時刻目に信号線aの値が"1"に遷移すると仮定する。1時刻目で信号線aの値が"0"となり2時刻目で信号線aの0縮退故障が検出できれば、信号線aの立ち上がり遷移故障が検出できる。

3.2 スキュードロード方式におけるドントケア抽出

遷移故障は2パターンテストで検出されるためATPG(Automatic Test Pattern Generation)で縮退故障検出用に生成されたテストパターンを2パターンテスト用に変換する必要がある。

ATPGにより与えられた縮退故障検出用テストパターン集合を $T=(t1,t2,t3,t4)$ とすると、以下に2パターンテストを行うテストパターンに変換する例を示す。

表1：与えられたテストパターン例

	p1	p2	p3	s1	s2	s3	s4	s5
t1	0	1	0	1	1	1	0	1
t2	0	0	0	1	0	0	0	0
t3	1	1	0	1	1	1	1	1
t4	0	1	1	0	0	0	1	1

例として、フルスキャンされた順序回路における縮退故障用テストパターンを変換する。

ATPGより表1のようなテストパターンが与えられたものとする。左3ビット(p1~p3)が外部入力から入力される値、右5ビット(s1~s5)がシフト動作で設定される擬似外部入力(SFF)への入力である。

ここでt1~t4までのテストパターンを1ビット逆シフトしたものをt1'~t4'すると、スキュードロード方式のテストパターンに変換すると表2のようになる。逆シフトとは、仮想的にスキャンパス方向の反対に値をシフトすることである。t1'~t4'は初期テストパターンでありt1~t4は遷移パターンである。

表2：スキュードロード方式による2パターンテスト用集合T

	p1	p2	p3	s1	s2	s3	s4	s5
t1'	X	X	X	1	1	0	1	X
t1	0	1	0	1	1	1	0	1
t2'	X	X	X	0	1	1	0	X
t2	0	0	0	1	0	1	1	0
t3'	X	X	X	0	0	1	1	X
t3	1	1	0	1	0	0	1	1
t4'	X	X	X	0	0	1	1	X
t4	0	1	1	0	0	0	1	1

縮退故障に関しては各t1~t4で検出されることがATPGによって保証されているので、SFFに関係ないt1'~t4'の外部入力値は0か1どちらでもよい。よってドントケアとして表記することができる。

擬似外部入力の値はスキャンチェーンをとおり各SFFへと伝搬されていく。よって、スキュードロード方式を適用させるために1ビットのみ逆シフトを行う。ここでシフトを行った後に発生するスキャンアウトに最も近いSFFの値は"0"か"1"どちらでも良いため、この部分もドントケアとして表記することができる。

表1の与えられたテストパターン集合に対してドントケア抽出を行い、その後逆シフトを行い表2のようなテストパターン集合Tを生成する。

4. 遷移故障検出を指向したドントケアの値の再割り当て

ドントケア抽出を行った表2のテストパターン集合Tに対し、遷移故障検出率が最大になるようにドントケアの値の再割り当てを行っていく。テストパターンtiで信号線lの縮退故障(s-a-v;v∈{0,1})を検出していて、逆シフトしたテストパターンtiで信号線lの値がXになっているものに着目する。

以下にそのアルゴリズムを示す

```

1: Procedure X-delay(T)
2: Test pat T;
3: {
4: for each test pattern ti' in T{
5:   if(ti = line-l(s-a-v) && (ti' = line-l(X))
6:   {
7:     while(ti == EOF)
8:     {
9:       "X" in ti' = assign 0 or 1 in ti;
10:      fault_simulation(T');
11:      if( fault_coverage(T') > max_fc)
12:      {
13:        white 0 or 1 in ti';
14:      }
15:    }
16:  }
17:  return T' composed of no_x(T);
18: }

```

図4：ドントケア再割り当てアルゴリズム

テストパターン t_i で信号線 l の縮退故障を検出し、なおかつ t_i' で信号線 l の値が X になっていると仮定する。その時 t_i' のドントケアの値を X 抽出前に設定して、ドントケアの値を1ビットずつ反転させながら検出できる遷移故障数が最大になるものを見つける。その時、遷移故障数が最大になるものを見つけ、ドントケアをその値へと再割り当てする。以下にその例を示す。

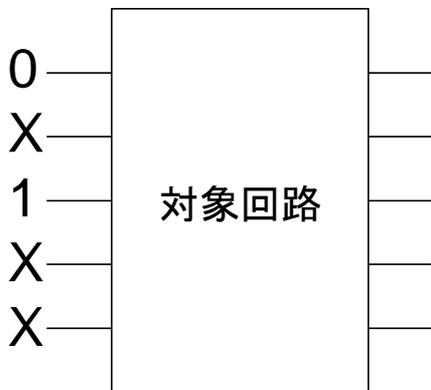


図5：X再割り当ての回路例

ドントケア抽出されたテストパターン集合 T に含まれるテストパターン t_i が、図5のように入力されていると仮定する。

このドントケアに対し、 X の値を1ビットずつ反転させながら検出される故障数を計測する。

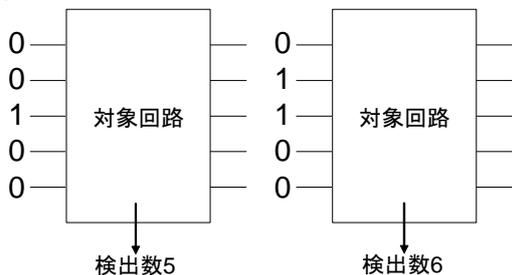


図6：X再割り当ての例1

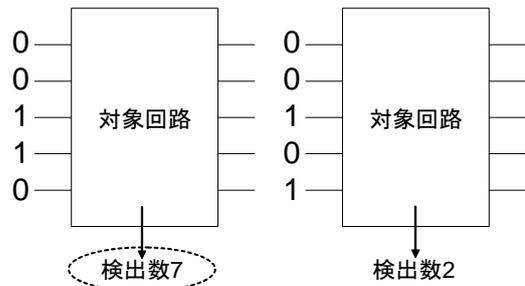


図7：X再割り当ての例2

図6と7のように、抽出したドントケアの値に0か1を割り当て、1ビットずつその値を変えていく。その中で最も検出数が多いものを選択してその時の値をドントケアに再割り当てを行う。この場合だと3番目の(0,0,1,1,0)が検出数7で最大となるため、このテストパターンにはこの値を再割り当てする。

5. 予備実験結果

Celeron(R) CPU 2.40GHz,メモリ 512MBの計算機上でC言語を用いて実装しISCAS'85ベンチマーク回路に対して実験を行った。テストパターンは縮退故障用ATPGによって生成され、圧縮されたものを用いている。

今回は、入力値はすべて擬似外部入力からの入力であるものとする。つまり、全ての入力にSFFが接続されていると仮定する。また今回はドントケア抽出は行わず、1ビット逆シフトのみを行ったテストパターンを用いて実験を行った。

表4：ISCAS'85実験結果

回路名	縮退故障			遅延故障		
	総故障数	検出数	検出率(%)	総故障数	検出数	検出率(%)
c17	22	22	100.00	36	18	50.00
c432	524	518	98.85	864	479	55.44
c499	758	740	97.63	998	301	30.16
c880	942	942	100.00	1760	1164	66.14
c1355	1574	1566	99.49	2710	468	17.27
c1908	1879	1870	99.52	3816	1483	38.86
c2670	2747	2678	97.49	5492	3668	67.15
c3540	3428	3307	97.47	7080	5015	70.83
c5315	5350	5291	98.90	10630	7681	72.26
c6288	7744	7710	99.56	12576	9148	72.74
c7552	7550	7497	99.30	15106	11832	78.33
平均検出率			98.92			56.28

表4はISCAS'85ベンチマーク回路における実験結果である。縮退故障に関しては問題なくほぼ100%の検出率を得ている。しかし、遅延故障の検出に関して多くても70%、c499などの回路に関しては30%ほどしか検出されていない。これは、スキュードロード方式による遷移故障の検出が、逆シフトして得た初期テストパターンに依存するためであると考えられる。

6. まとめ

本稿ではドントケア抽出の応用の1つとして、スキュードロード方式における遅延故障の検出率最大化を提案した。

今回は逆シフトのみを行った縮退故障用テストパターンで遷移故障の検出能力を評価した。

今後は、逆シフトして得られ初期パターンに対してドントケア抽出と再割り当てを行い、遷移故障の検出数を評価する。

「参考文献」

- [1]ToshinoriHosokawa,Masayoshi Yoshikawa, and MitsuyasuOhta. Novel DFT Strategies Using Full/Partual Scan Designs and Test Point Insartion to Reduce Test Application Time. IEICE A publocation of the engineering sciences society,2001 Nov.
- [2] Yoshinobu Higami, Shin-ya Kobayashi and Yuzo Takamatsu Dept. of Computer Science, Ehime University “A Method to Find Don't Care Values in Test Sequences for Sequential Circuits.”
- [3] Kohei Miyase and Seiji Kajihara, “XID: Don't Care Identification of Test Patterns for Combinational Circuits,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Val.23, No2,pp.321-326, Feb.2004.
- [4] J.Savir. Skewd-Load Transition Test: Part 1:, Calculus. Proceedings of IEEE International Test Conference, pages 705-713 October 1992.
- [5] J.Savir. Skewd-Load Transition Test: Part 2:, Calculus. Proceedings of IEEE International Test Conference, pages 714-722 October 1992.