

ドントケア抽出法の低消費電力化への応用に関する研究

日大生産工(学部) ○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.

生成されたテストパターンに対し, 故障検出効率を落とさないまま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用い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.

VLSIのテスト時には, 通常動作時の2倍から3倍の電力を消費することが報告されており[2], テスト時の消費電力を削減することが重要な課題となっている. 本稿ではドントケア抽出手法を用いて, テスト時の消費電力を削減する方法を提案する.

消費電力は, 各信号線の遷移回数に比例して増加するので, 抽出したドントケア部分に遷移回数を削減するように0または1の値を再設定し, 外部入力の遷移回数を削減する. それにより内部信号線の遷移回数の削減を試みる.

第2章でドントケア抽出技術について説明し, 第3章でドントケア抽出技術を用いたテスト時の低消費電力化法を提案する. 第4章で本手法を用いた実験の結果を表記し, 第5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

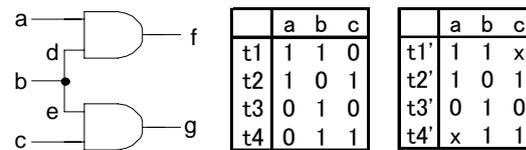


図1: 回路図とそのテストパターン

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. 同様にして $t4$ 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右側のようなテストパターン集合 T' を求めることができる.

Power Reduction Using Don't Care Identification Technique

Kazumitsu HAMASAKI, Toshinori HOSOKAWA

2. 3 ドントケア抽出アルゴリズム

図2にドントケア抽出のアルゴリズムを示す。

```

1: Procedure X-search(C,T)
2:   Circuit C; Test set T;
3: {
4:   for each test pattern ti in T {
5:     F = collect_essential_fault(ti);
6:     ti' = find_value(F);
7:     fault_simulation(ti');
8:   }
9:   for each test pattern ti in T {
10:    G = collect_undetected_fault(ti);
11:    ti' += find_value(G);
12:    fault_simulation(ti');
13:   }
14:   for each test pattern ti in T {
15:    H = collect_undetected_fault(ti);
16:    ti' += extended_find_value(H);
17:    fault_simulation(ti');
18:   }
19:   return T' composed of ti';
20:}

```

図2：ドントケア抽出アルゴリズム

4~8行では必須故障とそれに準ずる故障を検出し、必須故障を検出するために関係ないテストパターン中の値をドントケアに変換する。

9~13行では4~8行で検出することができなかった故障を、検出するようドントケアに値を再割り当てしていく。14~18行では故障シミュレーションにより検出されたが、13行終了時まで検出することが出来なかった故障に対し、拡張含意操作を行う。限定含意操作・限定正当化操作[5]はそれぞれ6・11行で使用される。

3 低消費電力化への応用

3. 1 消費電力

第2章で述べたドントケア抽出技術を用いてテスト時の低消費電力化を考える。テスト実行時の消費電力は以下ようになる。

$$\text{総消費電力} = \sum_j \frac{1}{2} C_L(j) \times V_{pp}^2 \times f_p(j)$$

上式においてCは各信号線の負荷キャパシタンスを示し、Vは電源電圧を示し、fは各テストパターンにおけるスイッチ抵抗を示す。

消費電力は各信号線の遷移回数の総和に比例することが知られており、本手法ではCの各信号線の負荷キャパシタンスの削減、つまり遷移回数の削減を目的とする。

3. 2 低消費電力化

3. 2. 1 外部入力の変移回数削減

今回提案する手法は、ドントケア抽出を用いて外部入力の変移回数を最小化することにより各内部信号線の遷移回数の削減に効果があるかを検討する。対象とする回路は組合せ回路である。

遷移回数の最小にする方法を以下に示す。

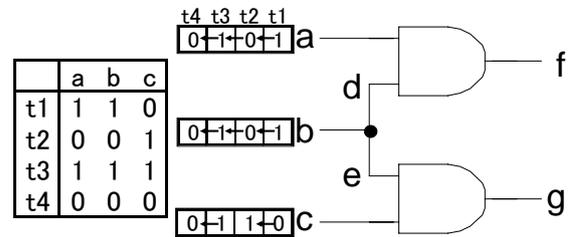


図3：ドントケア抽出前のテストパターン

例として、図3に示されたような回路図とテストパターンが与えられたとする。

この時外部入力の総遷移回数は、外部入力aは3回、bは3回、cは2回の計8回であり、この時内部信号線の総遷移回数は19回である。

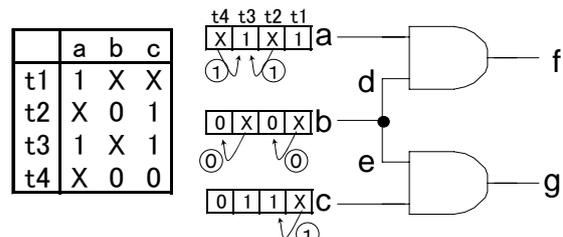


図4：Xの値の再割り当て

図4のようにドントケアの抽出が行われたとき、抽出されたドントケアに対して外部信号線の遷移回数を削減するようにXの値の再割り当てを行っていく。

値の再割り当てを行った後の外部入力の総遷移回数は、外部入力aは0回、bは0回、cは1回の計1回であり、この時内部信号線の総遷移回数は1回である。

このように、外部入力であるテストパターンの遷移回数を削減する事により、その回路の内部信号線の遷移回数を削減する事ができ、これにより消費電力を削減することができる。

3. 2. 2 テストパターンの並べかえ

今回の手法では適用していないが、テストパターンの並べ替えも考慮しながらドントケアの値の再割り当てを行うことを検討する。

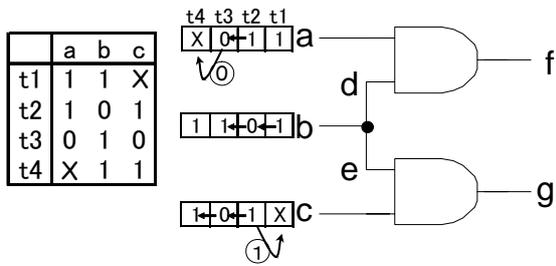


図5：並べ替えを行わない場合

図5のようにドントケア抽出を行ったテストパターンと回路図が与えられた時、テストパターンの並べ替えを行わない場合には外部入力の遷移回数は5回、内部信号線の総遷移回数は12回となる。

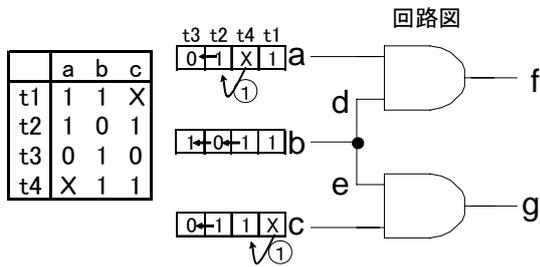


図6：並べ替えを行った場合

次に図6のように遷移回数を削減できるようなテストパターンを並べ替えた場合には、外部入力の遷移回数は4回、内部信号線の総遷移回数は10回となる。これよりXに値の再割り当てを行うだけでなく、テストパターンを並べ替える事によりさらに内部信号線の遷移回数を減らす事ができる。

3. 3 低消費電力化アルゴリズム

図7に低消費電力化のアルゴリズムを示す。

```

1: Procedure X-reduction(T)
2: Test pat T;
3: {
4:   for each test pattern t[i] in T{
5:     if( t[i] = find(X))
6:       {
7:         if (t[i+a] != find(X))
8:           {
9:             t[i]'=t[i];
10:            find_x(t[i]') = find_no_x(t[i+a]);
11:           }else{
12:             t[i]'= t[i];
13:             find_x(t[i]') =find_no_x(t[i-a]);
14:           }
15:         ALL_x(t[i]);
16:         t[i]' = "0"or"1";
17:       }
18:   return T' composed of ti';
19: }

```

図7：低消費電力化アルゴリズム

取り出した一つのテストパターン中にドントケアが存在していた場合、その次のテストパターンの値をXに割り当てる。またドントケアでない値はそのまま維持する。

次の値もXであったならばXではない値を検出するまでテストパターンの探索を行い、X以外を検出しならその値を取り込む。最後までX以外の値が出てこなければ、1つ手前のテストパターンの値をXに割り当てる。

すべてのテストパターンにおいて値がXであったならば“0”か“1”のどちらかを割り当てる。今回は“0”を割り当てる事とする。

4 実験結果

本稿の手法を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。ATPGによって作成されたテストパターンは圧縮されたものを用いている。

表1はその実行結果である。

表1において『Normal』は本稿の手法を用いずにATPGによって作成されたテストパターンをそのまま用いた時のテスト実行時間(CPUtime)や各内部信号線の総遷移回数(Transition)を示し、『After_X_reduction』は本稿の手法を用いてドントケアを抽出し、低消費電力化を施したテストパターンで実行した物を示す。

『Transition』は本稿の手法を用いたときと用いていないときでどれほど内部信号線の遷移回数を削減出来ているかを示し、

『reduction_time』はドントケアを含むテストパターンに対して“0”または“1”を再割り当てするのに要した時間を示している。

本手法により、各信号線の総遷移回数は最大でc1908の50%強という非常に良い結果を出すことができた。

しかし一方でc499やc880、c6288に対しては効果が低い。これはドントケア抽出の段階であまりドントケアを抽出することが出来なかったためであると考えられる。また、c499においては回路中にEXORが含まれているためにドントケアに値を設定してもあまり遷移回数に変化が無かったのではないかと考えられる。

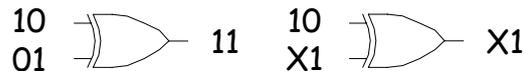


図8：与えられた状態

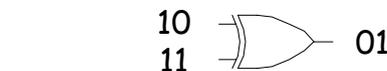


図9：X抽出後



図10：値の再割り当て後

表1 : ISCAS'85ベンチマーク回路実行結果

CircuitName	Fault coverage(%)	Normal			Arter X reduction		Transition	reduction_time CPUtime(sec)
		X extraction(%)	CPUtime(sec)	Transition	CPUtime(sec)	Transition	(N-X)/N(%)	
c432	99.24	43.17	0.11	7460	0.14	5447	26.98	0.01
c499	98.94	0.63	0.17	7585	0.16	7563	0.29	0.03
c880	100.00	49.12	0.34	11586	0.30	8668	25.19	0.03
c1355	99.49	0.23	0.97	24582	0.95	24576	0.02	0.03
c1908	99.47	61.11	1.92	93338	2.00	42281	54.70	0.03
c2670	96.03	76.39	3.52	76579	3.44	49763	35.02	0.05
c3540	96.25	56.95	7.03	167233	6.91	141083	15.64	0.05
c5315	98.90	72.26	12.80	226232	12.39	157882	30.21	0.06
c6288	99.56	13.49	18.31	93269	18.08	93310	-0.04	0.03
c7552	98.38	66.33	28.36	351015	29.25	283283	19.30	0.06

例えば図8のようにテストパターンが与えられていて、ドントケア抽出を行うと図9のようにドントケアを抽出できたとする。ここに本手法を用いて値の再割り当てを行うと図10のようになる。しかし、図10は外部入力の遷移回数こそ減少しているが、その出力の遷移回数は増加している。このような状態がc499の回路中でも起こっていると考えられる。

また、本手法を実行する際にドントケアの値の再割り当てを行うが、それにかかる時間は表1より殆どなく、これによるコストの増大はほぼないと言える。

5 まとめ

本稿ではドントケア抽出の応用の一つとして、ドントケア抽出の低消費電力化への応用を提案した。外部入力の遷移回数を削減すると、ほとんどの回路において内部信号線の遷移回数を削減できた。今後はテストパターンの並び替えを適用し、さらに消費電力を削減する予定である。

「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 publication of the engineering sciences society,2001 Nov.
- [2]SeijiKajiwara,KojiIshida,KoheiMiyase Aberage Power Reduction in Scan Testing by Test Vector Modification. 2002 Oct.
- [3]J.-S. Chang, and C.S. Lin, "Test Set Compaction for Combinational Circuits," IEEE Trans. on CAD, pp.1270-1378, 1995,Nov.

[4] 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."

[5]"論理回路のテストパターン中のドントケア判定とその応用に関する研究" 宮瀬 紘平.