

順序回路の k サイクルテスト集合に対する ドントケア判定法に関する研究

日大生産工 ○荒木 優 日大生産工 細川 利典

京産大 吉村 正義

1. はじめに

近年，超大规模集積回路 (Very Large Scale Integrated circuit : VLSI) が社会の様々なシステムの中で利用されるようになり，医療機器，交通システム，自動車制御などの高い信頼性が要求されるものに多く用いられている．一方で，半導体技術の発展により回路の複雑化，微細化が進み，製造ばらつきや回路の経年劣化への対応が問題となっている[1]．特に，システム稼働中の回路の経年劣化については NBTI（負バイアス温度不安定性），HCI（ホットキャリア注入効果），TDDB（酸化膜経時破壊），エレクトロマイグレーションやストレスマイグレーションの現象がよく知られており，微細化に伴って劣化が与える影響への懸念が高まっている[2]．現在でもライフタイムの予測や出荷前の信頼性試験および寿命試験が行われているが，個々の回路により実際の使用状況や使用環境は異なっており，劣化の進み具合も変化するため，それらを事前に把握することは困難である．例えば，高温，高電圧の使用環境では，NBTI によるトランジスタの遅延増加量が大きくなる事が知られている[1]．劣化による障害を回避する手段として，現状では動作マージン設計が行われているが，動作マージンは回路の製造ばらつき，動作環境，使用年数などの最悪の場合を考慮しながら決定するため，過大な動作マージンとなって VLSI の性能を犠牲にする可能性がある[1]．

障害を回避するもう一つの手段として，通常動作時に回路の出力や内部信号線の値を監視するオンラインテストもよく用いられている[1]．監視の方法とし

てはパリティチェックや信号の安定性検知機能付きの専用 FF[3] を用いてソフトエラー等の遅延障害に対応する方法が知られている．自動車に搭載されている VLSI に関しては，エンジンを起動したと同時に短時間でテストを行う必要があるため，VLSI にテストパターンを生成する回路を挿入し，オンラインでテストを行う方法 (BIST) [4] や，あらかじめ生成したテストパターンをメモリに格納し，短時間で少しずつテストを行う手法が用いられている．しかし BIST では十分な故障検出率の誇るテストパターンが逐一生成されるとは限らないしメモリを使用すると，オンラインテスト用メモリは通常のメモリに比べ 3 倍強の面積になる[3] などオーバーヘッドが大きいので，実用上は回路内の一部の FF にしか適用できず，回路全体の信頼性を保証できない．さらに自動車に搭載される VLSI に関しては製造段階での面積制約などが存在するため，面積オーバーヘッドは大きな問題であり，かつ人命に関わる機能を扱うため信頼性も重要であるため，信頼性を保持しつつ面積オーバーヘッドを削減する必要がある．それゆえ，テストデータを符号化して効率よく圧縮する技術の一つとして，テストパターン中のドントケアビットを判定する技術が提案されている[3]．また，オンラインテストでスキャンテストを用いると非スキャンテストよりも高い故障検出率は得られるが，テスト実行時間が長く，テスト時の消費電力が膨大なため効率的ではないゆえに，非スキャンテストを用いる．上記の問題に対して，高い故障検出率を得るための非スキャン

For The k-cycle Test Set of Sequential Circuits
Study on The Do Not Care Judging Method

Yutaka ARAKI, Toshinori HOSOKAWA, and Masayoshi YOSHIMURA

テストを対象としたテスト生成法として文献[6]が提案されている。このテスト生成法において、時間展開数が k ($k>0$) に限定される。したがって長さ k のテスト系列のテスト集合が生成される。

また、テスト系列のテスト集合が生成される非スキャンテストを対象とした順序回路のテスト系列のドントケア判定法が提案されている[1]。しかしながら、文献[1]の方法は、順序回路全体のテスト系列に対するドントケア判定であることから k サイクルテストのドントケア判定にそのまま適用することができない。また判定されるドントケア数が十分に大きいとは言えないことが課題である。

本論文では、オンラインテストで使用されるために高い故障検出率が得られる文献[6]でテスト生成した長さ k のテスト系列のテスト集合を対象としたドントケア判定法を提案する。第2章では、従来のオンラインテスト手法について説明を行い、第3章から第4章にかけて提案手法について説明を行う。第5章では実験結果について説明し、第6章でまとめ、今後の課題について述べる。

2. 従来手法

オンラインテストは、VLSIにテストを行うモジュールを挿入し、そのモジュールがシステムの起動時に自動で実行される。図1にオンラインテストモジュールを挿入したVLSIの簡単な構造図を示す。

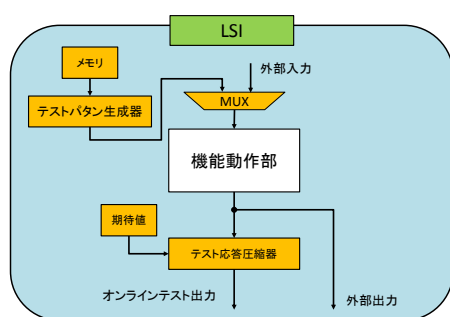


図1.オンラインテスト対応LSI構造図の例

図1のVLSIでは、まずテストパターン生成器からテストパターンが非検査回路に印加され動作し、出力応答がテスト応答圧縮器に伝搬する。その後、記憶しておいた期待値と出力応答を比較し、期待値通りの出力でない場合は危険信号をテスト用外部出力へ送り、機能の緊

急停止や危険警告をユーザに対して行う。

テストパターン生成器は一般的にランダムパターン生成器と決定的パターン生成器に分類され、ランダムパターン生成器では何らかのランダムパターン生成アルゴリズムに従ってパターン生成を行い出力する。ランダムパターン生成器を用いる手法ではBISTが多く知られている[4]。決定的パターン生成器は、あらかじめ製造段階でテストパターンをATPGなどでテスト生成し、メモリに格納しておいたものを出力する。現実的にランダムパターン生成器を用いて生成するテストパターンは高い故障検出率はあまり期待できないが、面積のオーバーヘッドが小さい。一方、決定的パターン生成器は故障検出率の高い性能を実現できるが、面積のオーバーヘッドが大きくなる傾向にある。また、従来手法ではLSIにフルスキャン設計やパーシャルスキャン設計を施しテストを容易化しているが、スキャンFFのシフト動作によるテスト実行時間の増大が問題として挙げられる。

本論文では、決定的パターン生成器に焦点を置き、格納するテストパターンをドントケア判定と符号化技術を組合せることにより削減し、テストデータ量を削減することを目的としている。自動車の起動時に短時間で行うオンラインテストを想定しているため、テスト実行時間や消費電力の増加を防ぐために、ノンスキャン設計でのオンラインテストを前提としている。

3. オンラインテストのためのテストデータ圧縮

以下の図2に本論文で提案するオンラインテスト手法についての大まかなフローチャートを示す。

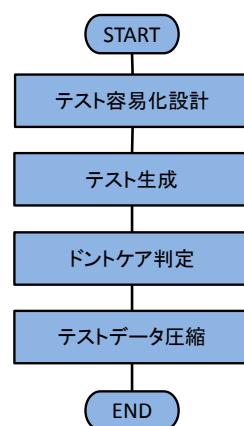


図2.提案手法フローチャート

本手法では、まず非スキャンでの高い故障検出率を

達成するためコントローラ拡大によるテスト容易化設計手法[5, 6]を用いて設計した回路を対象としている. 上記の手法によって設計された回路に対してテスト容易化機能的時間展開モデルを生成し, そのモデルに対してテスト生成を実行し, 長さ k のテスト系列のテスト集合を生成する. このテスト系列はテスト実行時間が短くオンラインテストに適している. また, おおむね機能動作を行うため, テスト時の消費電力も低くなる.

ドントケア判定はテスト生成により生成したテスト系列で故障シミュレーションを実行することで検出故障情報を抽出し, その情報に従い後方追跡や含意操作, 正当化操作を行うことでテスト系列中のドントケアになりうる部分を抽出し, 生成したテスト系列に対して, ドントケアを含んだ形に変換する. これによりテスト系列中のドントケアを利用し効率の良い符号化を行えるようにしている.

符号化に関してはテスト系列のビット幅をなるべく短くすることでメモリのビット数を削減することを目的としているため, 可逆圧縮である繰返しを使用することで圧縮を行う連長圧縮アルゴリズム[7]など簡単な符号化アルゴリズムを用いてなるべく回路面積のオーバヘッドの小さい符号化回路を使用しパターン長を削減する. 本論文では, コントローラ拡大によるテスト容易化設計手法[5,6]を適用した回路について故障シミュレーションとドントケア判定を行うため, 故障シミュレーションではテスト系列に対して時間展開数 k を指定し実行する (図3).

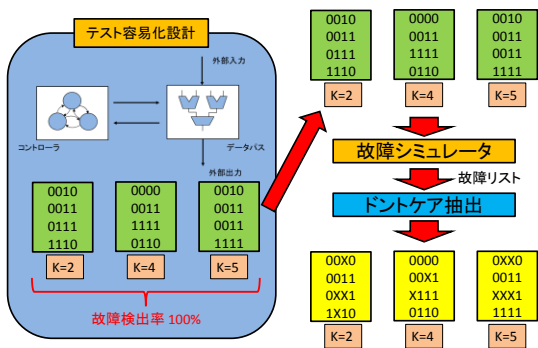


図3.Kサイクル分割の図

4. k サイクルテスト集合に対するドントケア判定

以下の図4に本手法で行う k サイクルテスト集合に

対するフローチャートを示す. また, その例を表1を用いて説明する.

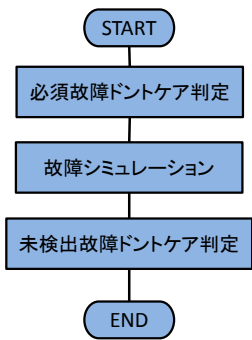


図4. k サイクルテスト集合ドントケア判定

表1.ドントケア判定の例

テスト系列	検出故障
t_1	f_1, f_2, f_4
t_2	f_2, f_3, f_8
t_3	f_4, f_5
t_4	f_5, f_6, f_7
t_5	f_5, f_6

表1はある k サイクルテスト集合中のテスト系列が検出する故障が示されている. 初期のテスト集合を T とすると, まず表1から f_1, f_3, f_7 はただ一つのテスト系列でしか検出できないため必須故障に分類され, それに対してドントケア判定を行い, 得られたテスト集合を T' とする. T' は必須故障の検出は保証されているが, T で検出できていた他の故障については考慮していない. そのため, T' に対して故障シミュレーションを実行し未検出故障を洗い出す. 例えば故障シミュレーションの結果, 表1の t_1 で f_4 , t_4 で f_6 が検出されたとする. このとき未検出故障は f_2, f_5, f_8 となり, この未検出故障に対して検出するテスト系列を選択し, ドントケア判定をもう一度行うことで未検出故障も検出可能となる. テスト系列の選択は符号化の際, 効率よく符号化を行うためにすべてのテスト系列のドントケア数なるべく均等になるように選択する.

参考文献

- 1) 梶原誠司, ” 組込み自己テストによるフィード高信頼化について”, DC2012-31, 2012年11月, pp39-42
- 2) W.Wang, et al., ”Compact Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology,” IEEE Trans. on Device and Material Reliability, Vol.7, No.4, pp.509-517, 2007.
- 3) 宮瀬紘平, 梶原誠司, ”論理回路のテストパターンに含まれるドントケア判定法について”, 情報処理学会研究報告, システムLSI設計技術研究会, 2001-SLDM-103, 2002年11月 pp111-116
- 4) Edward J. McCluskey, Stanford University, ” Built-In Self-Test Techniques”, IEEE DT1985, 1985年4月, pp21-28
- 5) Jun Nishimaki, Toshinori Hosokawa, Hideo Fujiwara, ” A Scheduling Method for Hierarchical Testability Based on Test Environment Generation Results”, Test Symposium (ETS), 2016 21th IEEE European, May 2016, pp23-27
- 6) Tetsuya, MASUDA, Jun NISHIMAKI, Toshinori HOSOKAWA, Hideo FUJIWARA, ” A Test Generation Method for Data Paths Using Easily Testable Functional Time Expansion Models and Controller Augmentation”, Test Symposium (ATS), 2015 IEEE 24th Asian, 2015, pp 22-25 Nov. 2015
- 7) S.W. Golomb, ”Run-length encodings”, IEEE Transactions on Information Theory, IT-12(3):pp399-401.
- 8) Thomas M. Niermann, Student Member, IEEE, Wu-Tung Cheng, Senior Member, IEEE, Janak H. Patel, Fellow, IEEE, ”PROOFS : A Fast, Memory-Efficient Sequential Circuit Fault Simulator”, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, VOL. 11, NO. 2, FEBRUARY 1992, pp198-207

5. 実験結果

本論文で構築した故障シミュレータを用いてドントケアを含まないテスト系列とドントケアを含むテスト系列との比較を行った。実験では、ISCAS ‘89 ベンチマーク回路を対象とし、テスト生成ツールのSTAGYを用いてテスト系列を生成した。また、生成したテスト系列に対してランダムに全体の20%, 40% 60%, 80%の割合でドントケアを割当て故障検出率と故障シミュレーションの実行時間を比較した。表の2に実験結果を示す。

表1のcircuitは回路名, exptimeは時間展開数, fault coverageは故障検出率, sim timeは実行時間, Xrateはランダムに割当てたドントケアの比率を示している。

表1のfault coverageからドントケアの比率が増加するごとに故障検出率は低下している。

この実験よりシミュレーション時間はほぼ同等であるが、ランダムにドントケアを割当てると故障検出率が著しく低下し、アルゴリズム的なドントケア判定が必要不可欠であることがわかる。

6. おわりに

本論文では、VLSIのオンラインテストに関するテストパターン生成器の圧縮効果を高めるためのkサイクルテスト集合に対してのドントケア判定法について提案した。

今後の課題として、kサイクルテスト集合のドントケア判定の実装と評価、さらにデータ圧縮の評価が挙げられる。

表2.実験結果

circuit	exptime	X rate									
		0%	20%	40%	60%	80%	0%	20%	40%	60%	80%
		fault coverage					sim time(sec)				
s208	5	32.93%	18.27%	16.35%	8.89%	6.25%	0.01	0.01	0.01	0.01	0.01
s298	5	23.66%	17.11%	16.61%	16.61%	12.92%	0.01	0.01	0.01	0.01	0.01
s386	5	56.99%	23.32%	12.95%	14.25%	5.18%	0.03	0.03	0.03	0.03	0.03
s820	5	25.55%	13.90%	7.44%	6.77%	5.61%	0.08	0.08	0.08	0.08	0.08
s1488	5	48.59%	43.92%	32.53%	22.04%	20.16%	0.39	0.39	0.36	0.36	0.36