

キャプチャセーフテストベクトルの動的テスト圧縮法

日大生産工 ○日下部 建斗 NTT データ 平井 淳士 日大生産工 細川 利典

日大生産工 山崎 紘史 日大生産工 新井 雅之

1. はじめに

大規模集積回路(Large Scale Integrated Circuits :LSI)の微細化や高速化, 電源電圧の低下にともない, 遅延故障のようなタイミング関連の欠陥を検出するためのテストは必要不可欠となっている [1]. また, 構造の単純さ, テスト生成や, 故障診断の容易性という観点から, LoC (Launch-On-Capture) [2]ベースの実速度スキューンテストが広く採用されている [3].

LoC ベースの実速度スキューンテストにおける消費電力は, シフト時の電力とキャプチャ時の電力に分類することができる. 過度のシフト電力は回路の発熱につながり, 回路の熱破壊を引き起こす. 一方, 過度のキャプチャ電力は過度の IR ドロップ [4]を引き起こし, 誤テストにより不必要な歩留りの低下を引き起こす [5].

次に, キャプチャ電力問題に着目する. キャプチャ電力削減において, キャプチャ動作時の信号線遷移数 (launch switching activity :LSA)を削減することが非常に重要である. キャプチャ電力削減のための手法は多数提案されており, それらは一般に, 回路構造変更による手法[6-9]と, テストデータ変更による手法[10-16]に分類することができる.

回路構造変更による手法では, テスト対象回路に対してテスト容易化のための回路構造変更を行う方法や, テスト専用の回路を付加する方法などがある. 一方で, テストデータ変更による手法では, 従来の故障検出重視のテスト生成手法を応用した方法やソフトコンピューティングの技術を応用した方法により, キャプチャ電力制約を満たしたテスト集合を生成している.

テストデータ変更による手法は一般的に, X 割当てによる手法と再テスト生成による手法に分類される. X 割当てによる手法[17]では, 初期テスト集合内のテストキューブに対して, 故障検出に関係のないドントケア(X)ビットに LSA を最小化するための論理値を適切に割当てることにより, 低消費電力テスト集合を生成する. しかしながら, 初期テスト集合において, 故障検出のための論理値割当ては消費電力を考慮せずに決定されている場合が多いため, X 割当て手法では低消費電力化の効果が初期テスト集合に依存するという問題がある. また, 再テスト生成による手法では, 高消費電力テストベクトルのみでしか検出できないアンセーフ故障[10]に対して, 消費電力制約を満たすようにテストベクトルの再生成を行う. 文献[11]では, PODEM ベースのテスト生成アルゴリズムに消費電力制約に基づくバックトラックや, テスト圧縮のための制約を追加している. しかしながら, 再テスト生成に基づく手法は, 決定論的アルゴリズムに基づく手法が

多く, テスト生成時間の長大化や実装の困難さという問題がある.

文献[19]では, アンセーフ故障の再テスト生成時に低消費電力初期テスト集合の中からアンセーフ故障の箇所から到達可能なファンアウトシステムを通して故障が検出されているテストベクトルと, 高消費電力初期テスト集合の中から目標のアンセーフ故障を検出しているテストベクトルを選択し, 故障励起に必要なケアビット以外のビットをX判定したテストキューブを合成する方法が提案されている.

本論文では, 文献[19]で提案された低消費電力テスト生成法の動的テスト圧縮部分の評価を行う. 本論文の構成は以下の通りである. 第2章では, 文献[19]のテスト生成法について説明する. 第3章では文献[19]のテスト圧縮手法について説明する. 第4章では文献[19]の動的圧縮部分の評価を行う. 第5章では実験結果を示し, 第6章では, 本論文のまとめと今後の課題について述べる.

2. キャプチャセーフテストパターンを利用した低消費電力テスト生成法

2. 1. 概要

文献[19]では, テストベクトルを故障励起のための割当てと, 故障伝搬のための割当てに分けて考える. 図1に文献[19]のテスト生成の概念図を示す. 図1の例では, 対象故障 f を励起するためのテストキューブ $(X, 1, X, 0, X)$ と故障伝搬のためのテストキューブ $(1, 1, X, X, 0)$ を組み合わせることで対象故障 f を検出するテストキューブ $(1, 1, X, 0, 0)$ を生成している. 文献[19]のテスト生成手法では, 故障励起のための(疑似)外部入力割当ては高消費電力テストベクトルを利用し, 故障伝搬のための(疑似)外部入力割当ては低消費電力テストベクトルを利用することで高速なテスト生成を実現する. 外部入力における故障励起のための割当てビット数と, 故障伝搬のための割当てビット数を比較すると, 故障励起のための割当てビット数の方が大幅に少なくなると考えられる. また, シミュレーションによる解析を行った結果, 故障励起のための割当てビット数は, 全(疑似)外部入力数に対して高々5~10%程度であった. このことから, 文献[19]のテスト生成手法では, 高消費電力テストベクトルを利用して得られた故障励起のための(疑似)外部入力割当てを, 故障伝搬可能な低消費電力テストベクトルと合成することで, 新たな低消費電力テストベクトルを生成することを考える. このテストベクトル合成で, 利用した低消費電力テストベクトルの割当て変更数は, 全(疑似)外部入力数に対して高々5~10%程度に抑えることができ, 利用した低消費電力テストベクトルと同程度の WSA 値

A Dynamic Test Compaction Method for Capture Safe Test Vectors

Kent KUSAKABE, Atushi HIRAI, Toshinori HOSOKAWA,
Hiroshi YAMAZAKI and Masayuki ARAI

をもつテストベクトルが生成できると考えられる。

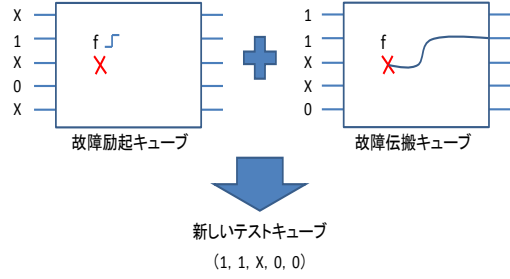


図 1. 低消費電力テスト生成手法の概念

2. 2. 低消費電力テスト生成手法全体フロー

文献[19]のテスト生成手法の全体フローを図 2 に示す。従来のテスト生成では、初期テスト集合から同定されたキャプチャセーフテスト集合[10]とキャプチャアンセーフテスト集合[10]を入力として処理を実行する。はじめに、再テスト生成対象故障(アンセーフ故障集合)の同定(Step1)を行い、再テスト生成対象故障集合内の故障に対してテストベクトル合成に基づくテスト生成を実行(Step3)する。動的テスト圧縮を実行し、二次故障を検出するようにテストベクトルを更新(Step4)し、 V_i' で故障シミュレーションを実行し、検出故障を故障集合 F_{target} から削除する(Step5)。生成されたテストベクトルを再生成テスト集合 T_{gen} に追加(Step6)する。テスト生成対象故障集合 F_{target} 内の全ての故障に対してテスト生成が完了したら、キャプチャセーフテスト集合 T_{safe} と再生成テスト集合 T_{gen} の和集合を最終テスト集合 T' として取得する(Step8)。

```

C:Circuit
Tsafe : Capture Safe Test Set
Tunsafe : Capture Unsafe Test Set
Test_generation(C, Tsafe, Tunsafe) {
1. Ftarget = select_target_faults(C, Tsafe, Tunsafe);
2. for_each fault fi in Ftarget {
3. vi' = synthesis_based_test_generation(C, Tsafe, Tunsafe, fi);
4. vi' = dynamic_comp(C, Tunsafe, Ftarget);
5. fault_simulation(C, vi', Ftarget);
6. Tgen = Tgen U vi';
7. }
8. T' = Tgen U Tsafe;
}

```

図 2. 低消費電力テスト生成手法の全体フロー

2. 3. 対象故障選択

テスト生成対象故障の選択方法について説明する。従来手法では、入力として与えられたテスト集合内のアンセーフテストベクトルでのみ検出可能な故障(アンセーフ故障)に対して再テスト生成を実行する。表 1 に対象故障選択の例を示す。表 2 の例では、初期テスト集合 $T=\{tp1, tp2, tp3, tp4, tp5\}$ に対して故障シミュレーションを実行した結果である。tp1 と tp2 はアンセーフテストベクトルである。低消費電力テスト生成では、アンセーフテストベクトルでのみ検出可能な故障を対象故障として再テスト生成を実行するため、アンセーフ故障集合 $F_{target}=\{f1, f2\}$ を算出する。

表 1. テスト生成対象故障選択

WSA		f1	f2	f3	f4	f5	f6	f7	f8	f9	f10
unsafe	tp1	●	●	○							
unsafe	tp2		●	○	○						
safe	tp3			○	○	○	○				
safe	tp4						○	○	○		
safe	tp5								○	○	○

2. 4. テストベクトル合成に基づくテスト生成

低消費電力テスト生成手法では、キャプチャセーフテストベクトルを利用し、テストベクトル操作を実行することで、対象故障に対するテスト生成を実現する。図 3 は低消費電力テスト生成のテストベクトル合成に基づくテスト生成アルゴリズムである。低消費電力テスト生成アルゴリズムでは、対象故障 f_{target} を検出するテストベクトルが生成されるまで Step2~Step16 の処理を繰り返す(Step1)。低消費電力テスト生成は、ベースとして利用するキャプチャセーフテストベクトル t_{base} を探索(Step2)し、 t_{base} の選択に失敗した場合、処理を終了する。 t_{base} の選択に成功した場合、対象故障 f_{target} を検出可能なアンセーフテストベクトルから故障励起のためのテストキューブ t_{ex} の抽出(Step4)を行い、 t_{base} と t_{ex} のベクトル合成により、新たなテストベクトル t_{gen} を生成(Step5)する。 t_{gen} で f_{target} を対象に故障シミュレーションを実行する(Step6)。ここで、 t_{gen} の WSA 値がキャプチャ電力制約閾値未満であり、かつ f_{target} が検出されている場合、 t_{gen} を新しいキャプチャセーフテストベクトルとして t' にコピーする(Step8)。それ以外の場合は、Step2 から処理を再開する。 t' が生成されるか、 t_{base} が存在しない場合、テスト生成を終了する。

```

C:Circuit
Tsafe : Capture Safe Test Set
Tunsafe : Capture Unsafe Test Set
Ftarget : Target Fault
synthesis_based_test_generation(C, Tsafe, Tunsafe, ftarget) {
1. while ftarget is not detected {
2. tbase = select_reuse_test_vector(Tsafe, ftarget);
3. if tbase exists {
4. tex = get_fault_excitation_cube(Tunsafe, ftarget);
5. tgen = test_vector_synthesis(tbase, tex);
6. fault_simulation(C, tgen, ftarget);
7. if (ftarget is detected && WSA(C, tgen) < threshold) {
8. t' = tgen;
9. }
10. else {
11. continue;
12. }
13. }
14. else {
15. break;
16. }
17. }
18. return t';
}

```

図 3. テスト生成アルゴリズム

2. 5. キャプチャセーフテストベクトルの選択

図4は利用するキャプチャセーフテストベクトルの選択例を示す。低消費電力テスト生成手法において、利用するキャプチャセーフテストベクトル t_{base} は以下の条件を満たす。

条件 1: t_{base} はキャプチャセーフテストベクトルである。

条件 2: テスト生成対象故障 f_{target} を含むファンアウトフリー領域を FFR_{target} と定義するとき、 t_{base} は FFR_{target} の出力信号線に伝搬した故障影響を(疑似)外部出力に伝搬可能である。

図4の例において、利用するキャプチャセーフテストベクトル t_{base} として、キャプチャセーフテストベクトル2が選択される。

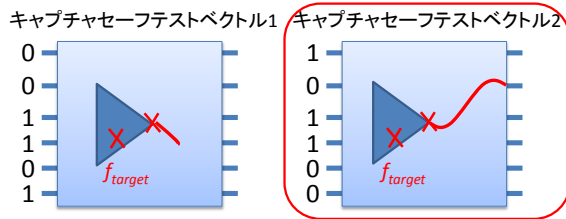


図4. テストベクトル t_{base} の選択例

2. 6. 故障励起テストキューブの抽出

対象故障 f_{target} を励起するテストキューブの抽出手順はX判定手法 [6]の考えに基づく。故障励起のためのテストキューブの抽出では、初期テスト集合において対象故障 f_{target} を検出可能なアンセーフテストベクトルに対してバストレースを実行することで、高速に対象故障 f_{target} を励起するための入力割当てを抽出する。また、低消費電力テスト生成手法では、対象故障 f_{target} を励起するための入力割当てに追加して、 f_{target} の故障影響を f_{target} を含むファンアウトフリー領域の出力まで伝搬するための入力割当てを抽出することで、テストベクトル合成に基づくテスト生成の成功率を高めている。図5は故障励起テストキューブ抽出の例である。図5の例では、対象故障 f_{target} の励起とファンアウトフリー領域の出力までの伝搬のためのテストキューブ t_{ex} は(X, 1, 0, X, X, X)である。

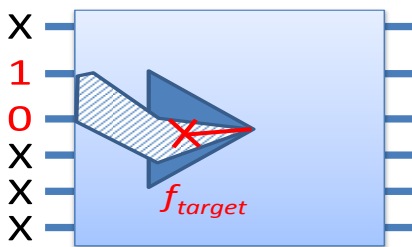


図5. 故障励起テストキューブ抽出例

2. 7. テストベクトル合成

キャプチャセーフテストベクトル t_{base} と対象故障 f_{target} を励起するためのテストキューブ t_{ex} を表2の規則に基づき合成することにより、新しいテストベクトルを生成する。図6にテストベクトル合成の例を示す。図6の例では対象故障 f_{target} を励起するためのテストキューブ $t_{ex} = (X, 1, 0, X, X, X)$ とキャプチャセーフテ

ストベクトル $t_{base} = (1, 0, 1, 1, 0, 0)$ を表3の合成規則に基づき合成することにより、新たなテストベクトル $t_{gen} = (1, 1, 0, 1, 0, 0)$ を生成している。ただし、再生成されたテストベクトル t_{gen} が対象故障 f_{target} を検出する保証はない。

表2. テストベクトル合成規則

$t_{ex} \backslash t_{base}$	0	1	X
0	0	1	0
1	0	1	1

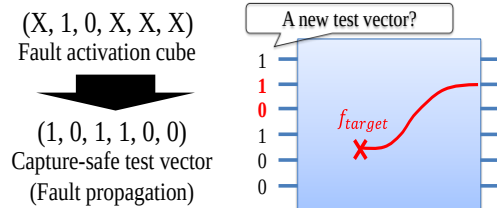


図6. テストベクトル合成例

3. テスト圧縮

3. 1. 動的圧縮

低消費電力テスト生成アルゴリズムにおいて、1個のキャプチャセーフテストベクトルに対して合成可能な故障励起割当てが複数存在する場合、それらを逐次的に合成することにより動的にテスト圧縮を実行することが可能である。

テストベクトル合成により、新しいキャプチャセーフテストベクトルが生成されたとき、生成されたキャプチャセーフテストベクトルを T_{base} とし、二次故障を対象故障として選択し、その二次故障励起割当て t_{ex} を合成する。このとき、二次故障選択により二次故障が検出されないか、キャプチャ電力制約を違反した場合には最後に実行したテストベクトル合成の処理を取り消す。

低消費電力テスト生成の動的圧縮を以下にまとめる。

- ・二次故障は故障励起テストキューブが、一次故障の故障励起テストキューブと衝突しないものを選択する。
- ・二次故障の検出に失敗または低消費電力化できない場合は動的圧縮を打ち切る。

3. 2. 静的圧縮

従来手法の静的圧縮は二重検出法 [7]のアルゴリズムに基づく。通常二重検出法では、必須テストベクトルに対する故障シミュレーションを実行したのち、冗長テストベクトルに対する故障シミュレーションはテスト生成により出力された順番の逆順に実行される。しかしながら、この方法では圧縮後のテスト集合のキャプチャ電力が高くなってしまふことが考えられる。そのため文献[19]では、冗長テストベクトルに対する故障シミュレーションをWSA値の昇順に実行することにより、キャプチャ電力の低いテストベクトルを優先的に採用する。

4. 実験結果

文献[19]の動的圧縮において、二次故障の選択を再検討する必要があると考えた。表3に一次故障に対する二次故障数の表を示す。表3において、回路は回路名を表しており、最大は一故障に対して選択された二次故障の最大数、最小は一故障に対して選択された二次故障の最小数、平均は二次故障選択数の平均、最頻値は二次故障選択した場合に何個選択された場合が最も多いのか、割合は全一次故障に対して最頻値である二次故障数がどのくらいあるのかという割合を表している。表4より二次故障数が最小の場合、0が多く一度も二次故障が選択されずに動的圧縮を終了している。また、最頻値は全回路とも1であり、二次故障選択はしているが、あまり効率は良くないことがわかる。このため二次故障選択の部分を再検討し、多くの二次故障を選択することで更に圧縮効率を高めることが期待できる。

表3. 一次故障に対する二次故障数

回路	最大	最小	平均	最頻値	割合
s5378	13	0	2	1(77)	57%
s9234	30	0	2.5	1(94)	59%
s13207	16	0	2.6	1(32)	46%
s15850	14	0	2	1(38)	62%
s35932	136	1	5.5	1(128)	55%
s38417	18	0	1.5	1(788)	75%
s38584	6	0	2	1(4)	57%

5. まとめ

本論文では、文献[19]の動的圧縮部分の評価を行った。文献[19]の動的圧縮では、全体の46~75%のテストパターンに二次故障数が1であった。これは、二次故障のテスト生成に失敗した後の二次故障選択が行われていないためである。今後の課題として二次故障選択を改善することが挙げられる。

参考文献

- [1] Y. Sato, S. Hamada, T. Maeda, A. Takatori, Y. Nozuyama and S. Kajihara, "Invisible Delay Quality - SDQM Model Lights Up What Could Not Be Seen," *Proc. ITC*, p. Paper 47.1, 2005.
- [2] J. Savir and S. Patil, "Scan-based transition test," *IEEE Trans. Comput. Aided Design Int. Circuits & Syst.*, vol. 13, no. 8, pp. 1057-1064, 1994.
- [3] L. -T. Wang, C. -W. Wu and X. Wen, "VLSI Test Principles and Architectures," *Design for Testability, San Francisco, CA, USA.*, 2006.
- [4] J. Saxena, K. M. Butler, V. B. Jayaram, S. Kundu, N. V. Arvind, P. Sreepakash and M. Hachinger, "A case study of IR-drop in structured at-speed testing," *Proc. Int. Test Conf.*, pp. 1098-1104, 2003.
- [5] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," *Proc. VLSI Test Symp.*, pp. 4-9, 1993.
- [6] N. Ahmed, M. Tehranipoor and V. Jayaram, "Transition delay fault test pattern generation considering supply voltage noise in a SOC design," *Proc. Design Autom. Conf.*, pp. 533-538, 2007.
- [7] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault and S. Pravosoudovitch, "A gated clock scheme for low power scan testing of logic lcs or embedded cores," *Proc. ATS*, pp. 253-258, 2001.
- [8] P. Rosinger, B. M. Al-Hashimi and N. Nicolici, "Scan architecture with mutually exclusive scan segment activation for shift- and capture-power reduction," *IEEE Trans. Comput. Aided Design Int. Circuits & Syst.*, vol. 23, no. 7, pp. 1142-1153, 2004.
- [9] C. Zhen and X. Dong, "Low-Capture-Power at-speed testing using partial launch-on-capture test scheme," *Proc. VTS*, pp. 141-146, 2010.
- [10] X. Wen, K. Miyase, S. Kajihara, H. Furukawa, Y. Yamato, A. Takashima, K. Noda, H. Ito, K. Hatayama, T. Aikyo and K. K. Saluja, "A Capture-Safe Test Generation Scheme for At-Speed Scan Testing," *Proc. European Test Symposium*, pp. 55-60, 2008.
- [11] X. Wen, Y. Yamashita, S. Kajihara, L. -T. Wang, K. Saluja, and K. Kinoshita, "A New Method for Low-Capture-Power Test Generation for Scan Testing," *IEICE Trans. Inf. & Syst.*, Vol. E89-D, No. 5, pp. 1679-1686, 2006.
- [12] V. R. Devanathan, C. P. Ravikumar and V. Kamakoti, "A stochastic pattern generation and optimization framework for variation-tolerant, power-safe scan test," *Proc. ITC*, pp. 1-10, 2007.
- [13] M. -F. Wu, K. -S. Hu, and J. -L. Huang, "LPTest: A flexible low-power test pattern generator," *J. Electron. Test.*, vol. 25, no. 6, pp. 323-335, 2009.
- [14] X. Wen, Y. Yamashita, S. Kajihara, L. -T. Wang, K. K. Saluja and K. Kinoshita, "Low-Capture-Power Test Generation for Scan Testing," *Proc. VTS.*, pp. 265-270, 2005.
- [15] S. Remersaro, X. Lin, Z. Zhang, S. M. Reddy, I. Pomeranz and J. Rajski, "Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs," *Proc. ITC, Proc. ITC*, 2007.
- [16] X. Wen, K. Miyase, S. Kajihara, T. Suzuki, Y. Yamato, P. Girard, Y. Ohsumi, and L. -T. Wang, "A Novel Scheme to Reduce Power Supply Noise for High-Quality At-Speed Scan Testing," *Proc. Test Conf. ITC.*, pp. 1-10, 2007.
- [17] K. Miyase and S. Kajihara, "XID: Don't Care Identification of Test Patterns for Combinational Circuits," *IEEE Trans. Comput. Aided Design Int. Circuits & Syst.*, vol. 23, no. 2, pp. 321-326, 2004.
- [18] S. Kajihara, I. Pomeranz, K. Kinoshita and S. M. Reddy, "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits," *IEEE Trans. Comput. Aided Design Int. Circuits & Syst.*, vol. 14, no. 12, pp. 1496-1504, 1995.
- [19] A. Hirai, T. Hosokawa, Y. Yamauchi, and M. Arai, "A low capture power test generation method using capture safe test vectors," *Proc. ETS*, pp. 1-2, 2015.