ISSN 2186-5647

2-4

k サイクルキャプチャテスト生成のための テスト容易化設計に関する評価

日大生産工　○佐藤　護　　日大生産工　細川　利典

1. はじめに

近年，半導体集積技術の発展に伴い，設計される大規模集積回路(Large Scale Integrated circuits：LSI)の大規模化，複雑化が急速に進展している[1]．それに伴い，LSI のテスト生成が重要な課題となっている．一般的に LSI は，フリップフロップ（Flip-Flop：FF）を持つ順序回路で設計される．

フィードバックループを持つ順序回路に対するテスト生成は，高い故障検出効率を達成することが困難である可能性がある．そのため，順序回路に対する効果的なテスト生成アルゴリズム[2]や，与えられた順序回路に対してテスト生成容易な回路に変更するテスト容易化設計方法[3]が提案されている．

現在，スキャン設計[4]が LSI のテスト容易化設計方法として普及している．スキャン設計は，回路中の FF の値を観測・制御可能なスキャン FF に置き換え，各スキャン FF をシフトレジスタ状に接続する．その結果，回路の可制御性・可観測性を向上させることが可能となる．回路に存在するすべてのレジスタをスキャン FF で設計することをフルスキャン設計とよぶ．フルスキャン設計では，組合せ回路に対するテスト生成技術を適用することが可能となる．そのため，高い故障検出効率を達成することが可能となる．しかしながら，回路面積の増大や，消費電力の増加などのハードウェアオーバーヘッドの増大という問題点がある．

フルスキャン設計のハードウェアオーバーヘッドを削減するテスト容易化設計として，パーシャルスキャン設計[5]が提案されている．パーシャルスキャン設計[5]とは，回路中の特定のレジスタのみをスキャン FF で設計する手法である．しかしながら，順序回路のテスト生成技術を適用する必要があり，高い故障検出効率を得るためには，スキャン化するレジスタを特定する技術が重要となる．

スキャン化するレジスタを特定するためのアルゴリズムは数多く提案されているが，それらのアルゴリズムが適用されたパーシャルスキャン設計では，フルスキャン設計と同等の故障検出効率を得ることは困難である．また，フルスキャン設計と同等の故障検出効率を得るためのパーシャルスキャン設計法も提案されているが，その場合スキャン化するレジスタの割合が高くなる．

コントローラとデータパスから構成されるレジスタ転送レベル(Register Transfer Level：RTL)における非スキャン設計をベースとしたコントローラ拡大によるテスト容易化設計方法［6,7］が提案されている．[6]では，コントローラに状態や状態遷移を追加することで，高い故障検出効率を達成しているが，テスト生成時間に課題が残る．[7]では，コントローラに状態遷移を追加し，コントローラの状態遷移の情報をテスト生成の制約としたテスト容易化機能的時間展開モデル[7]を用いたテスト生成法を提案している．コントローラの状態遷移を制約にすることで，テスト生成時間は大きく改善されている．

本論文では，[7]のような特別なテスト生成法を用いないで，市販の順序回路のテスト生成ツールを用いて，高い故障検出効率を得ることのできるテスト容易化設計法を検討し，そのための評価を行う．

本論文では，コントローラ中のレジスタはスキャン設計することを前提としている．その理由として，コントローラ中のレジスタ数はデータパスのレジスタ数に比べて非常に小さいからであり，かつコントローラのテスト容易性の向上が回路全体のテスト容易性の向上に大きな影響を与えるためである．さらに，本論文ではコントローラの拡大と状態遷移のドントケア割当て，さらにデータパスの一部のレジスタのスキャン化をテスト容易化設計の候補として検討し，パーシャルスキャン設計において，スキャン化レジスタ数を最小限に抑えながら，フルスキャン設計と同等の故障検出効率の達成を目標とする．

2. k サイクルキャプチャテスト生成とテスト容易化設計手法

本章では，本論文で用いるテスト生成法とテスト容易化設計を説明する．

An Evaluation of Design for Testability for k-Cycle Capture Test Generation

Mamoru SATO and Toshinori HOSOKAWA

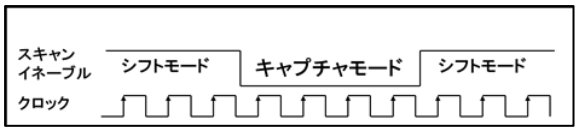

図 1　k サイクルキャプチャテスト例 (k=4)

2.1. k サイクルキャプチャテスト生成

本節では, k サイクルキャプチャテスト[7]について説明する. k サイクルキャプチャテストとは, スキャンテストのキャプチャモード時のサイクル数が k であるテスト方式である. 図 1 に, 4 サイクルキャプチャテストの例を示す. 図 1 で示すように, キャプチャモード時に k（k=4）サイクル間順序動作を行う. k を設定して k 時刻分の時間展開モデルを用いてテスト生成することを k サイクルキャプチャテスト生成と呼ぶ[7]. k の値を小さく設定することにより, テスト生成時間が短縮され, 効率的にテスト生成を実行できる. しかしながら, k の値が小さいと, 多くのレジスタをスキャン化する必要があるため, k の値の設定には注意を払う必要がある. 本論文では k の値を, 3～10 の間で検討する.

2.2. パーシャルスキャン設計

本節では, パーシャルスキャン設計について説明を行う. パーシャルスキャン設計とは, 回路中の一部のレジスタをスキャン FF で設計するテスト容易化設計手法である.

本論文では, k サイクルキャプチャテスト生成を前提としているので, コントローラの動作と k 時間展開したデータパス回路モデルをもとに, 可制御性・可観測性の低いレジスタに着目し, テスト容易化設計（スキャン化）を施す必要がある. パーシャルスキャン設計はそのスキャン化レジスタの 1 時刻目を外部入力および k 時刻目を外部出力と見なすことが可能であり, テスト容易性の向上を見込めるが, スキャン化レジスタの割合が高いとハードウェアオーバーヘッドも大きくなる. 本論文では, スキャン化の割合を 20%以下に目標を置く.

2.3. コントローラ拡大

本節では, コントローラ拡大[8]について説明する. コントローラ拡大[8]とは, コントローラに状態遷移や状態の追加, 無効状態の状態遷移の設計を行うことである.

本論文では, 文献[9]と同様にコントローラ拡大を用いて, テスト容易化機能的時間展開モデル[9]の動作の実現をねらう. しかしながら, 本論文では, 専用のテスト生成を用いないため, 市販のテスト生成ツールがテスト容易化機能的時開モデル通りの制御を行うとは限らないため, 高い故障検出率が得られるとは限らない. よって, 可能な限りテスト容易化機能的時開モデルのとおり制御されるような設計をし, テスト容易な動作を制御するようにテスト生成される確率を高くする必要がある.

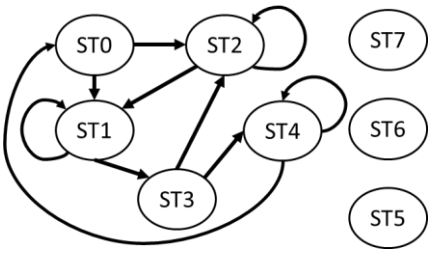

図 2　コントローラ拡大前

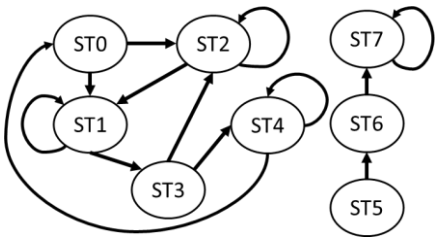

図 3　コントローラ拡大後

図 2 に初期コントローラ, 図 3 にコントローラ拡大の例を示す. 図 2 において, ST0～ST4 はリセット状態と, そのリセット状態から到達可能な有功状態である. このコントローラの状態割当てを, バイナリーエンコーディング, グレイエンコーディングなどの状態レジスタ数が$\log_2 N$個になるアルゴリズムを適用することを仮定すると, $2^{\log_2 N} - n$個の無効状態が存在する. 図 2 では, ST5,ST6,ST7 が無効状態である. レジスタ転送レベルでは, 無効状態の状態遷移が設計されておらず, 論理合成を用いて, 論理回路を合成する時点で面積や性能などを最適化するように, 各無効状態遷移を決定する.

本論文では, テスト容易化設計として, レジスタ転送レベルで無効状態とその状態遷移を明に設計する. 図 2 において, ST0→ST1 の状態遷移で外部入力から外部入力レジスタに値を設定し, ST4→ST4 の状態遷移で外部出力レジスタの値を外部出力で観測できるものとする. 最短の状態遷移パスが動作可能であったとしても, ST0→ST1→ST2→ST3→ST4→ST4 と 5 サイクル必要である. 一般にこのサイクル数が増加すると, テスト生成が困難であることが知られている. 図 3 で示すように, 無効状態 ST5, ST6, ST7 の各状態遷移を ST5→ST6(スキャンインで ST5 に遷移し, 外部入力から外部入力レジスタへ値を設定), ST6→ST7(テスト困難な演算器のテストを実行し,), ST7→ST7(外部出力でテスト結果を観測)を設計する.

2.4. 状態遷移のドントケア割当て

本節では, 状態遷移のドントケア割当てについて説明を行う. 状態遷移のドントケア割当てとは,

各時刻におけるデータパス回路の演算器やレジスタのアイドル状態における制御信号が 1，または 0 のどちらでもよいドントケア（X）であることに着目した手法である．ドントケアを 1 及び 0 に設定し，アイドル状態の演算器やレジスタに対して制御を行い，テスト容易性の向上を狙う手法である．本手法は，テスト容易化設計のための面積オーバーヘッドを最小限に留めることが可能である．このドントケアの値を書き換えても通常動作には影響せず，テストピンの挿入も必要ないため，ほとんどハードウェアオーバーヘッドなしでテスト容易化設計を施すことが可能となる．しかしながら，テスト容易性の向上に関して，パーシャルスキャン設計，コントローラ拡大と比較すると，その効果は大きくないと考えられる．

3. k サイクルキャプチャテスト生成のためのテスト容易化設計の検討

本章では，k サイクルキャプチャテスト生成のためのテスト容易化設計の検討経過について説明を行う．本手法ではテスト容易化設計された回路に対して k 時間展開し，テスト生成を行う．

各テスト容易化設計手法の通常回路に対する面積オーバーヘッド・故障検出効率の増加の割合・テスト実行時間への影響度が異なる．本論文では，各テスト容易化設計の効果を評価する．

図 4 に Sehwa の RTL データパス回路を示す．この回路を用いて各テスト容易化設計の検討を行う．

Sehwa の RTL データパス回路は，外部入力＝{i1，i2，i3，i4，i5，i6，i7，i8}，外部出力＝{PO1, PO2}，演算器＝{ADD0, SUB0, LESS0}，レジスタ＝｛R0，R1，R2，R3，R4，R5，R6，R7，R8｝，MUX＝｛M1，M2，M3，M4，M5，M6，M7，M8，M9｝で構成されている．このオリジナル回路に対しての市販テスト生成ツールを用いたテスト生成結果は故障検出効率が 72.88%であった．コントローラ中のレジスタのみスキャン設計すると，故障検出効率は 93.81%となった．この結果から，コントローラのテスト容易性の向上が回路全体のテスト容易性の向上において重要であることがわかる．また，フルスキャン設計では故障検出効率は 95.57%となった．

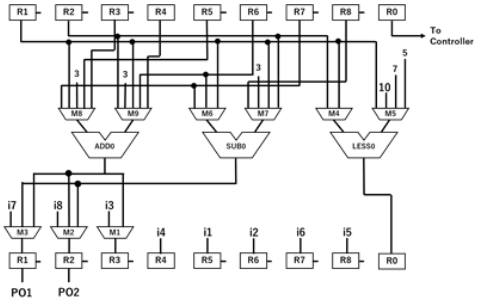

図 4　Sehwa データパス図

本研究の目標は，フルスキャン設計に近い故障検出効率を達成することとなるので，故障検出効率の目標を 95%にした．

次に状態遷移のドントケア割当ての検討実験を行った．オリジナルの RTL コントローラにはアイドル状態の演算器やレジスタの制御に対して X が割当てられており，動作が未定義である．しかしながら，論理合成時には X に 1 か 0 のいずれかの値が割当てられる．この X の値をテスト容易化設計として任意の値に制御することによって，故障検出効率を向上させることを目標とする．はじめに，オリジナルのコントローラの制御信号がドントケアである部分をすべて 1，すべて 0 で満たした回路をそれぞれ Sehwa_1 と Sehwa_0 とし，テスト生成をおこなった．Sehwa_1 では故障検出効率が 81.67%，Sehwa_0 では故障検出効率 73.73%となった．この結果から，制御信号のドントケア割当てにより故障検出効率において 10%程度差がでることがわかった．しかしながら，この割当ては，狙った故障を検出しているわけではない．表 1 に，Sehwa のオリジナル回路における各ハードウェア要素の未検出故障数を示す．

表 1　未検出故障一覧

要素名	全故障数	未検出故障数	故障検出率	要素名	全故障数	未検出故障数	故障検出率
mux9	913	291	68.1	mux2	422	81	80.8
mux8	588	190	67.7	mux7	458	52	88.6
mux5	232	154	33.6	mux6	422	83	80.3
LESS0	374	183	51.1	ADD0	532	11	97.9
mux4	292	174	40.4	REG4	96	0	100.0
REG5	388	388	0.0	SUB0	566	0	100.0
REG6	388	0	100.0	REG0	6	0	100.0
REG3	388	98	74.7	REG1	388	0	100.0
mux1	292	98	66.4	mux3	422	0	100.0
REG8	388	0	100.0	REG2	388	0	100.0
REG7	388	388	0.0	pi	325	65	80.0

表 1 から，REG5 の多くの故障が未検出であることがわかる．しかしながら，Sehwa_1，Sehwa_0 では REG5 の多くの故障が検出されており，ドントケア割当てにより REG5 の故障が容易に検出可能であることがわかったため，次の検討実験の方針とした．

次の検討実験では，REG5 の故障をすべて検出することを目標とし，制御信号の X 割当てを行った．REG5 の故障を検出するために，REG5 に接続されている外部入力 i1 から，外部出力 PO1 までのテスト容易な経路(R5→M8→ADD0→M3→R1→PO1)を一意に選択するように経路上の要素に対してドントケア割当てを行った．その回路にテスト生成を行った結果，REG5 の故障を検出することができなかった．原因の考察として，いかにコントローラのドントケアを制御しようとも，テスト生成時にそのテスト容易な経路を制御する状態遷移が実行されなかったため，効果が低いことが考察される．

次に，本論文で前提としているコントローラに

対するスキャン設計を行い, 同様の割当てを行い, 制御信号の X 割当てをそのテスト容易な経路を選択するように行って，テスト生成した結果，REG5 の多くの故障が検出され, REG5 の未検出故障数を 388 から 2 まで削減できた．コントローラのスキャン設計と制御信号のテスト容易なドントケア割当ての組合せが効果的であることがわかった．

一方，REG5 をスキャン化した結果，すべての故障を検出することが可能となった．くわえて，他の要素がテスト困難になることもなかった．この結果から, データパス中のテスト容易性の低いレジスタをスキャン化することが効果的であるが，面積オーバーヘッドは大きい．コントローラ拡大に関してはまだ実験結果を出すことができていないため，省略する．

4. 実験結果

本章では, 本手法の有用性検討の結果のまとめ, 詳細な回路情報の説明，実験環境，考察のまとめを行う．

実験環境は，OS：Linux Red Hat Enterprise Linux Server release7.3(Maipo)，CPU：Intel® Xeon® CPU E3-1271 v3@3.60GHz，メモリ：32GB のマシンを使用した．また，論理合成ツールに DesignnVisionL-2016.03-SP5-2，テスト生成ツールに TetraMax(4.8.6)を使用した．実験では制御信号のドントケア割当て, コントローラのスキャン設計, データパスのパーシャルスキャン設計, k サイクルキャプチャテスト生成のそれぞれの有用性の比較実験を行った．対象回路は Sehwa である．実験結果の図を載せた後，それぞれに対する簡単な考察を載せて本章を終える．

表 2　実験結果まとめ

回路種類	回路面積	全故障数	故障検出率	故障検出効率
Sehwa_original	2547	9092	72.65	72.88
Sehwa_ContScan	2563	8678	93.09	93.39
Sehwa_fullscan	3499	9106	95.34	95.57
Sehwa_0	2542	9020	73.49	73.73
Sehwa_1	2610	9344	81.42	81.67
Sehwa_REG5,ContScan	2670	9140	91.88	92.17

Sehwa_original とは，テスト容易化設計を適用してない通常の回路 Sehwa に対してテスト生成をした結果である．Sehwa_ContScan とは，Sehwa のコントローラ中のレジスタのみにスキャン設計を施し，テスト生成をした結果である．Sehwa_original と比較して，故障検出効率が大きく向上しており, コントローラの状態を自由に設定できることは回路全体のテスト容易性を大きく向上させることがわかる．Sehwa_fullscan は, データパスとコントローラすべてのレジスタに対してスキャン設計を施した結果である．Sehwa_0 はコントローラのドントケアをすべて 0 に割当て，Sehwa_1 はコントローラのドントケアをすべて 1 に割当て，合成した回路である．Sehwa_REG5,ContScan は，コントローラ中のレジスタをスキャン設計し, REG5 を含むテスト容易な経路を制御できるように制御信号のドントケア割当てを行った回路に対してテスト生成した結果である．

5. おわりに

本論文では，k サイクルキャプチャテスト生成のためのテスト容易化設計に関する評価の検討結果をまとめた．結果として，3 種類のテスト容易化設計を適切に使用する評価関数を定義することによって, フルスキャン設計と同等の高い故障検出率を達成し得る回路が生成されると考えられる．今後の課題として，コントローラ拡大によるテスト容易化設計の検討とデータパスのレジスタのスキャン化の検討が挙げられる．

参考文献

[1] 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, 2004.

[2] Hideo.Fujiwara , Logic Testing and Design for Testability, The MIT Press,1985.

[3] T.P.Kelsey, K. K. Saucier, and S.Y.Lee, "An Efficient Algorithm for Sequential Circuit Test Generation, " IEEE Trans. on Computers, vol. 42, no.11, pp. 1361-1371, Nov. 1993.

[4] T. Hosokawa, S. Takeda, H. Yamazaki, M. Yoshimura, “Controller Augmentation and Test Point Insertion at RTL for Concurrent Operational Unit Testing, ” IEEE IOLTS'17 23rd, pp.17-20, Thessaloniki, Greece, July.2017.

[5] E.B.Eichelberger and T.W. Will iams, "A logic design structure for LSI testabi li ty, " in Proc. Design Automation Conf., pp.462-468, June, 1977.

[6] R. Nara, K. Satoh, M. Yanagisawa, T. Ohtsuki, and N. Togawa, “Scan-Based Side-Channel Attack Against RSA Crypto-Systems Using Scan Signatures”, IEICE Trans. on Fundamentals, Vol. E93-A, No. 12, pp2481-2489, 2010.

[7] E. K. Moghaddam, J. Rajski, S. M. Reddy, M. Kassab, “At-Speed Scan Test with Low Switching Activity,” IEEE VLSI Test Symposium, pp177-182, 2010.

[8] L.M.FLottes , B.Rouzeyre , L.Volpe,”A Controller reSynthesis Based Methos For Improving Datapath Testability”, IEEE International Symposium on Circuits and Systems, pp. 347 -350, May 2000.

[9] T. Masuda, J. Nishimaki, T. Hosokawa and H. Fujiwara, "A Test Generation Method for Datapaths Using Easily Testable Functional Time Expansion Models and Controller Augmentation," IEEE the 24th Asian Test Symposium (ATS'15), pp. 37-42, Nov. 2015.