2008－SLDM－137（3）
2008 / 11 / 17

実速度スキャンテストにおける キャプチャセーフテスト生成手法について

高嶋 敦之† 大和 勇太† 古川 寛† 宮瀬 紘平† 温 暁青† 梶原 誠司†

†九州工業大学大学院情報工学研究院 〒820-8502 福岡県飯塚市川津 680-4

E-mail: †{takashima, yamato, furukawa}@aries30.cse.kyutech.ac.jp, {k_miyase, wen, kajihara}@cse.kyutech.ac.jp

あらまし スキャンテストのキャプチャモードにおいて，回路内部での過度のラウンチ信号変化が原因となってタイミングエラーを引き起こす可能性がある．このような誤テストを回避するためのキャプチャセーフテスト生成は，テスト歩留まり低下を避けるために重要である．しかし，従来技術では，ラウンチ信号変化をある程度削減することはできるが，完全なキャプチャセーフ性を保証することはできない．そこで本論文では，高精度のキャプチャセーフ判定性基準，及びX判定とX割当てによる効果的なキャプチャセーフテスト生成技術を提案する．更に，ベンチマーク回路を用いた実験データで提案手法の有効性を示す．

キーワード 実速度スキャンテスト，キャプチャセーフテスト，キャプチャセーフ性判定，X判定，X割当て

A Capture-Safe Test Generation Scheme for At-Speed Scan Testing

Atsushi TAKASHIMA† Yuta YAMATO† Hiroshi FURUKAWA†
Kohei MIYASE† Xiaoqing WEN† Seiji KAJIHARA†

† Dept of CSE, Kyushu Institute of Technology, 680-4 Kawazu, Iizuka-shi 820-8502, Japan

E-mail: †{takashima, yamato, furukawa}@aries30.cse.kyutech.ac.jp, {k_miyase, wen, kajihara}@cse.kyutech.ac.jp

Abstract Capture-safety, defined as the avoidance of any timing error due to unduly high switching activity in capture mode during at-speed scan testing, is critical for avoiding test-induced yield loss. Although point techniques are available for reducing capture IR-drop, there is a lack of complete capture-safe test generation flows. The paper addresses this problem by proposing a novel and practical capture-safe test generation scheme, featuring (1) reliable capture-safety checking and (2) effective capture-safety improvement by combining X-bit identification & X-filling with low launch-switching-activity test generation. This scheme is compatible with existing ATPG flows, and achieves capture-safety with no changes to the circuit-under-test or the clocking scheme.

Keyword At-Speed Scan Test, Capture-Safety, Capture-Safety Check, X-Identification, X-Filling

1. はじめに

1.1. 背景

LSIの順序回路のテストにおいて，フルスキャン回路は広く用いられているテスト容易化設計手法である［1］．フルスキャン設計は，回路内部のフリップ・フロップをシフトレジスタ機構を付加したスキャン・フリップ・フロップ(スキャンFF)に置き換え，そのスキャンFFをチェーン状に繋げることにより構成されている．そして，フルスキャン回路は，外部からスキャンFFを直接可制御・可観測するシフトモードと，組合せ回路部の応答をスキャンFFに取り込むキャプチャモードの2つのモードで動作する．シフトモードでは，シフトイン操作でスキャンFF数のクロックを入れてテストパターンの印加をし，シフトアウト操作によってスキャンFFに格納されている値を同様に，スキャンFF数のクロックを入れて値を取り出す動作をする．一方，キャプチャモードでは，スキャンFFは通常のFFと同じ動作をし，組み合わせ回路部からの応答をスキャンFFに取り込むという動作をする．

近年，LSIの設計，製造技術の進歩により，製造プロセスの微細化，高集積化，高速化が進んでいる．それに伴って，タイミングに関係した欠陥の増加が大きな問題になってきている．このタイミングに関係した欠陥を検出し，テスト品質を確保するために，実速度スキャンテスト(At-Speed Scan Testing)を行うことが必須となっている[2]．しかしながら，実速度スキャンテストは主に2つの要因により誤テストによる歩留り低下を生じやすい．1つは，テスト時には回路内部での多くの信号値変化が起こり，過度のIRドロップが引き起こされることで，もう1つは，テスト応答のキャプチャサイクルが短く，IRドロップによる遅延の影響を受けやすいということである．そのため，スキャンテスト時の消費電力の削減は，実速度テストの重要な課題となっており，シフト消費電力とキャプチャ消費電力の削減が求められている．

1.2. スキャンテスト時の消費電力削減手法

スキャンテストの消費電力には，シフト時とキャプチャ時に発生するものが存在する．シフト時の消費電力は，テストベクトルがスキャン回路のスキャンチェーンに印加される際に発生する．このシフト時の消費電力の増大は，蓄積的に過度の発熱によって回路を破損させ，信頼性を低下させる可能性がある．一方，キャプチャ時の消費電力は，テストベクトルによる組合せ回路部分の出力応答が FF へ取り込まれる時に，その FF が保持している値と取り込まれる値が異なる場合に発生する．このキャプチャ時の消費電力の増大は，瞬間的な IR ドロップを引き起こす．その瞬間的な IR ドロップによって，FF の誤動作や回路素子の遅延の増加などが発生する．その結果，テスト時に正常な回路を不良品と誤って判定してしまう誤テストが起こり，歩留り低下の原因となる．

シフト時の消費電力の削減手法としては，クロックゲーティング [3] やスキャンチェーン操作 [4]，テストスケジューリング操作 [5]，テストベクトル操作 [6] など多くの手法が提案されている．一方，キャプチャ時の消費電力削減手法は，キャプチャクロックタイミングはシフト時のクロックタイミングよりも厳しいため周波数の周期と位相のタイミング操作が困難であり，またシフト時の消費電力削減のように回路操作を行うこともタイミング設計の観点から見て危険である．以上の特性から，キャプチャ時の消費電力削減は，回路変更なしでかつ，クロック変更の必要がないテストベクトル操作で行うアプローチが一般的である．現状では，シフト時のようにキャプチャ時の消費電力の削減は多くの手法が提案されているわけではなく困難である．

1.3. Launch-off-Capture (LoC) 手法

実速度のスキャンテストの手法として，Launch-off-Capture (LoC) 手法というものがある．LoC 手法のクロック方式を図 1 に示す．シフトモード (SE=1) では，テストベクトルをスキャン FF 数のクロックを入れる．つまり，最後のシフトパルス S_L でテストパターン v の値がすべての FF に値がセットされる．次に，キャプチャモード (SE=0) で，ラウンチキャプチャ (C_1) で組合せ回路部の応答を取り込む．そして，C_1 で 2 パターン目を 1 クロックでセットし，テスト応答キャプチャ (C_2) でスキャン FF に組合せ回路部の応答を再び取り込む．この C_1 と C_2 間のテストサイクルを実速度で行うことにより，タイミングに関係する欠陥を検出する．

図 1 の C_1 において信号値遷移 LSA(Launch Switching Activity)が過剰に起こってしまうと，過度な IR ドロップが発生し，その影響で論理素子の遅延が大きく増大してしまう．これによって、パス遅延がテストサイクルより大きくなってしまうと，C_2 で FF が誤った値を取り込んでしまう．これが原因で，テスト時に本来良品のチップを不良品と判定してしまうという誤テストが起こってしまい，歩留まり低下を引き起こす問題が起きる．この歩留まり低下を避けるために，テストベクトル v に対して，キャプチャ時の C_2 でタイミングエラーを起こさないというテストベクトルのキャプチャセーフ性を保証する必要性が高まっている．

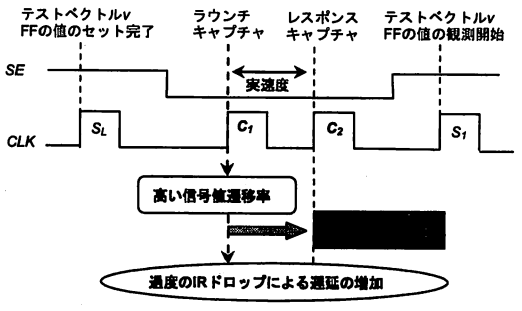

図 1 LoC 手法のクロック方式

1.4. キャプチャセーフ性の判定

テストベクトルのキャプチャセーフ性の判定は，実速度スキャンテストにおいて，パス遅延の増加がタイミングエラーを引き起こすかどうかを判定するものである．しかし，パス遅延を直接解析することは，非常に高価である．そのため，現実的には間接的にパス遅延の解析を行う．間接的なキャプチャセーフ性を判定する手法として，IR ドロップ見積もる手法[7]と，ラウンチ信号値遷移を見積もる手法 [8]がある．例えば IR ドロップを見積もる手法として，*SCAP* (*Switching Cycle Average Power*) を使用して平均キャプチャ電力を見積もる手法がある [7]．しかし，この *SCAP* を計算する手法は IR ドロップと良い相関性があるかもしれないが，物理情報が必要であるため計算コストがかかる．一方，ラウンチ信号値遷移 LSA を見積もる手法として *GTC* (*Global Toggle Constraint*), *GITC* (*Global Instantaneous Toggle Constraint*), *RITC* (*Regional Instantaneous Constraint*) のようなトグル制約条件によって見積もる手法がある [8]．このトグル制約条件による手法は，ゲートレベルの条件だけのため計算コストが安いという利点があるが，IR ドロップとパス遅延の影響との相関性が薄いかもしれないという欠点がある．

1.5. 提案と論文構成

本論文では，回路全体と部分領域での遷移を考慮した高精度なキャプチャセーフ性判定基準，集中的な X 判定と X 割当てによるテストベクトル操作，低 LSA テスト生成の 3 つの技術を組み合わせたキャプチャセーフ性を保証するためのキャプチャセーフなテスト集合を生成するキャプチャセーフテスト生成フローを提案する．

本論文の構成は以下の通りである．2 節では準備として提案手法に関する要素技術を述べ，3 節では提案手法の全体フローについて説明し，4 節ではフローの各フェーズについて説明する．また，5 節で実験結果を示し，6 章で本論文をまとめる．

2. 準備

2.1. テストベクトル操作

テストベクトル操作とは，あるテストベクトルに対し，X(ドントケア)判定という処理を行い，故障検出に関係のない未定値Xビットを抽出し，そのXビットに適当な論理値を割当てることによって，故障検出率を維持したまま目的にあったテストベクトルに変換する処理のことである．Xビットとは論理値0，1どちらでも割当てることができるものである．X判定とX割当てについて2.2.1節と.2.2.2節で詳細を説明する．図2ではテストベクトルvに対するテストベクトル操作の例を示している．まずvに対して，故障検出に関係のないビットをX判定により特定する．次にXビットに対して適当な論理を割当てる．このようにvは故障検出率を維持したまま、別のテストベクトルに変換できる．

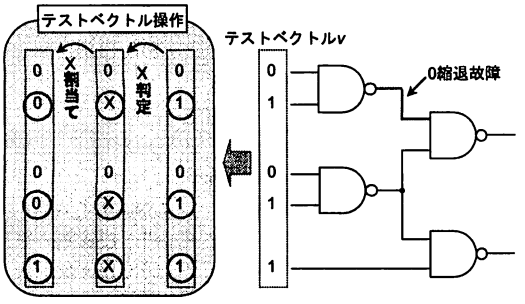

図2 X判定&X割当てによるテストベクトル操作

2.2.1. X判定

X判定にはシミュレーションと正当化/含意操作の2つのアプローチがある．本手法で使用している X 判定は，正当化/含意操作のXIDと呼ばれる手法である [9]．XIDの処理時間はおおよそ故障の数とテストベクトル数に比例し，一般にシュミレーションのものよりも短く，さらにこの処理は多くのXビットを特定できるという利点をもつ．このX判定により特定されたXビットを利用することで，テストベクトル操作を行うことが可能になる．

2.2.2. X割当て

X割当てにはランダム，マージ，アルゴリズムの3つのアプローチがある．ランダムX割当てはテストベクトルのXビットにランダムに0と1を割当てる．マージは2つ以上のテストベクトルを1つのテストベクトルに併合することで，Xビットの値を決定する．例えば，1X0と11Xをマージすると110になる．この考え方はテストベクトルの静的圧縮で用いられている[12].

アルゴリズム的X割当ての手法として，低LSA X割当て手法がある [10,11]．低LSA X割当てとは，テストキューブ中に存在するXビットに対し，キャプチャ時の回路内部の信号値遷移LSAを削減可能なように適切な論理値0,1を割当てる処理である．低LSA X割当て Preferred-Fill [10] や JP-Fill [11] などの多くの手法が提案されている．低LSA X割当てを行うと，変換前のテストベクトルvのLSA数よりも変換後のテストベクトルv'のLSA数の方が減少する．

3. 提案手法の全体フロー

提案するキャプチャセーフテスト生成フローの概要を図3に示す．このフローは，テストベクトル数と故障検出率をそれぞれ維持したままキャプチャセーフなテスト集合を生成することを目指している．提案手法は次の3つのフェーズで構成されている．

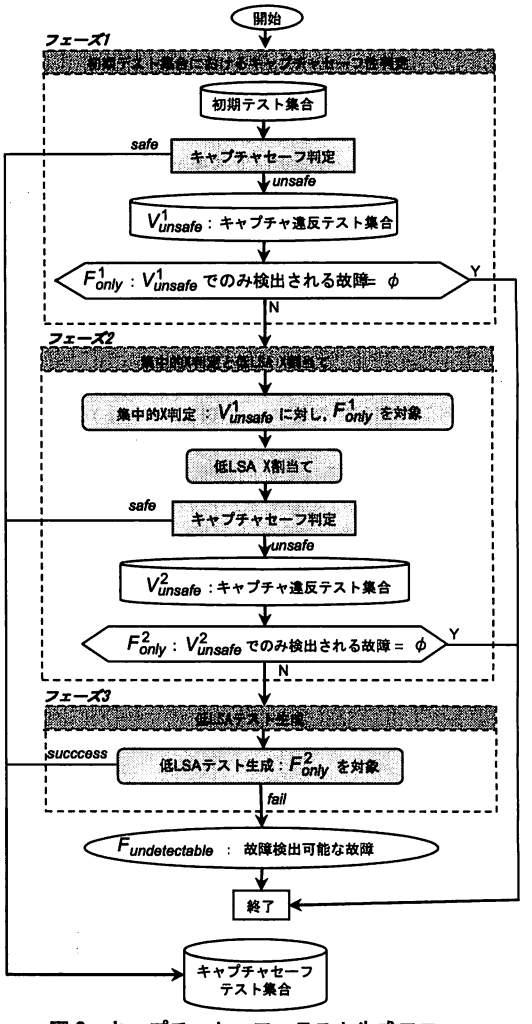

図3 キャプチャセーフ・テスト生成フロー

・**フェーズ1**： 与えられた初期テスト集合に対して，キャプチャセーフ性判定を行う．そして，キャプチャ違反テストベクトル(V^1_{unsafe})を特定する．次にそのベクトルでのみ検出される故障(F^1_{only})を抽出する．キャプチャセーフ性判定の詳細については4.1で述べる．

・**フェーズ2**： もしF^1_{only}が空でないならば，F^1_{only}に対して，

キャプチャセーフなテストベクトルに改善する処理を行う．キャプチャセーフ性の改善処理として，集中的 X 判定と低 LSA X 割当てを組み合わせた処理を行う．これによりテストデータ量と故障検出率をそれぞれ維持したままで，キャプチャセーフなテストベクトルに変換できる．集中的 X 判定と低 LSA X 割当ての詳細については，4.2 で述べる．

・フェーズ 3：もし F^2_{only} が空でないならば，F^2_{only} を検出するキャプチャセーフなテストベクトルを生成する低 LSA テスト生成を行う処理を実行する．この処理は直接テスト生成を行うため，テストデータ量の増大や故障検出の低下を引き起こすかもしれない．キャプチャセーフテスト生成の詳細については 4.3 で述べる．

4. 提案手法の要素技術

4.1. キャプチャセーフ性判定

1.3 節で述べたように，実速度スキャンテスト時のテストベクトルのキャプチャセーフ性はラウンチ信号値遷移 LSA に依存している．そこでまず，電源配線と回路モデルをについて説明する．そして，その回路モデルを考慮した LSA 解析の新しい制約条件と，新しい制約条件を利用したキャプチャセーフ性判定を説明する．

4.1.1. 電源配線と回路モデル

回路内部の電源配線とノード(FF やゲート)との関係を表現した一般的なモデルを図 4 に示す．同様のモデルが文献 [8] で使用されている．この回路モデルでは，ある一つの領域(Feed-Region:FR)は電源配線からの同じビアから供給されているノードの集まりであるとしている．そして，回路全体がその FR 領域(*feed-region*)の集まりで構成されている．信号値遷移 LSA 解析を行う際に，回路全体の LSA 数だけではなく，それぞれ FR 領域でも LSA 数も考慮する．また，キャプチャサイクル(図 2 中の C_1 と C_2 間)全体の LSA 数だけでなく，キャプチャサイクルの瞬間の LSA 数も考慮する．この 4 つの LSA 解析を行うことで，高精度のキャプチャセーフ性判定を実現することができる．

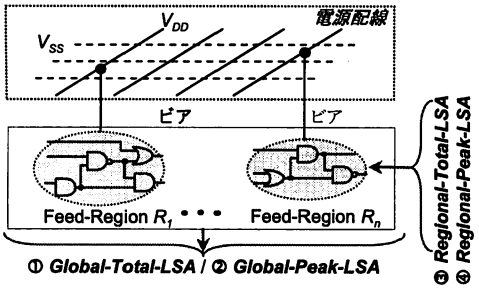

図 4　電源配線と回路モデル

4.1.2. LSA 解析の基準

入力ベクトルを v，回路内部に FR 領域が n 個($R_1,R_2,...,R_n$)の回路で，キャプチャサイクル間($0 \leqq j \leqq m$)のある一瞬の時刻を T_j する．その時の入力ベクトル v の時刻 T_j におけるある FR 領域 R_i の LSA を，以下の $LSA(v,R_i,T_j)$ で定義する．

$$LSA(v,R_i,T_j) = \sum_{k=1}^{p}((f_k+1)\times sk(T_j))$$

p は FR 領域 R_i でのノード数，f_k はノード $k(k=1,2,..,p)$ からのファンアウトブランチ数，$sk(T_j)$ は時間 T_j におけるノード k の出力の遷移確率をそれぞれ表している．一度，この $LSA(v, R_i, T_j)$ をキャプチャサイクル間のそれぞれの瞬間の T_j $(0 \leqq j \leqq m)$ において，それぞれの FR 領域 R_i $(i=1,2,..,n)$ で計算する．そして，得られた LSA 解析を基にして 4 つの新しい基準の LSA 数を以下で定義する．

① 回路全体の *LSA* ($LSA_{GT}(v)$)

$$LSA_{GT}(v) = \sum_{i=1}^{n}\sum_{j=1}^{m} LSA(v,R_i,T_j)$$

② 回路全体の瞬間的 *LSA* ($LSA_{GP}(v)$)

$$LSA_{GP}(v) = max(\sum_{i=1}^{n} LSA(v,R_i,T_1)\ ,\ \ldots\ ,\sum_{i=1}^{n} LSA(v,R_i,T_m))$$

③ 回路領域の *LSA* ($LSA_{RT}(v)$)

$$LSA_{RT}(v,R_i) = \sum_{j=1}^{m} LSA(v,R_i,T_j)$$

④ 回路領域の瞬間的 *LSA* ($LSA_{RP}(v)$)

$$LSA_{RP}(v,R_i) = max(LSA(v,R_i,T_1)\ ,\ \ldots\ ,LSA(v,R_i,T_m))$$

$LSA_{GT}(v)$ は回路全体のキャプチャサイクル間の LSA 数の総和，$LSA_{GP}(v)$ は回路全体のキャプチャサイクル間で LSA 数が最大になった瞬間の LSA 数，$LSA_{RT}(v)$ は回路領域でのキャプチャサイクル間における LSA 数の総和，$LSA_{RP}(v)$ は FR 領域でのキャプチャサイクル間で LSA 数が最大になった瞬間の LSA 数をそれぞれ表している．この 4 つの LSA 解析には，ゲートレベルの情報と単位遅延（Unit Daley）シミュレーションのみ使用しているため計算が容易である．また，ゲートの出力ノードに重みをかけることで，実際の電気容量によりよく近似しているため高精度であるという特徴を持つ．

4.1.3. キャプチャセーフ性判定

4.1.2 で定義した新しい LSA の解析基準を用いて，キャプチャセーフ性判定を行う．キャプチャセーフなベクトルは以下の制約条件を全て満たすテストベクトル v であると定義する．

$LSA_{GT}(v) \leqq Limit\_LSA_{GT}$
$LSA_{GP}(v) \leqq Limit\_LSA_{GP}$
$LSA_{RT}(v) \leqq Limit\_LSA_{RT}(R_i)$　(i=1,2,…,n)
$LSA_{RP}(v) \leqq Limit\_LSA_{RP}(R_i)$　(i=1,2,…,n)

リミットは楽観的過ぎず，かつ悲観的過ぎず適切に設定する必要がある．リミットの設定はテストの目的に依存する．一般的に，製造テストのリミットは通常動作時の消費電力の 2 倍まで許容するように設定している [14]．また，電力特性に基づいて，キャプチャセーフな条件のためのリミットは，機能トグル率，もしくはシミュレーション機能ベクトルを用いることで容易に得ることができる [7].

4.2. 集中的 X 判定と低 LSA X 割当て

図 5 は，初期テスト集合に対するキャプチャセーフ性判定

の結果，キャプチャ違反とは判定されたテストベクトル(V^1_{unsafe})に対し実行されるフローを表している．

X 判定とは 2.2.1 節で説明したように故障検出率を維持したままテストベクトルから X ビットを抽出する方法である．そして，この図 3 で使用している集中的 X 割当てとは，対象としているテストベクトル v に対して，v のみで検出される故障を保証するように X 割当てを行うことである．通常の X 判定では v で検出される全故障を保証するように，X 判定を行っている．しかし，v で検出される全故障の中には，他のテストベクトルで検出される故障も存在する．その故障を除くことで，X 判定で保証する故障数を減らす．それにより，多くの X を v から抽出することができるようになる．そのように，より多くの X を抽出することにより，低 LSA X 割当てによって，v をキャプチャセーフなテストベクトルに改善する機会が増加する．本提案手法フローでは低 LCP X 割当て手法として JP-Fill [11] を使用している．

また，集中的 X 判定と低 LSA X 割当ての処理をし，まだキャプチャセーフなベクトルに改善可能ならば，集中的 X 判定と低 LSA X 割当ての処理を繰り返し実行する．この N 回繰り返しにより，効果的にかつテストベクトル数を維持したまま，キャプチャセーフ性を持たせる可能性がある．

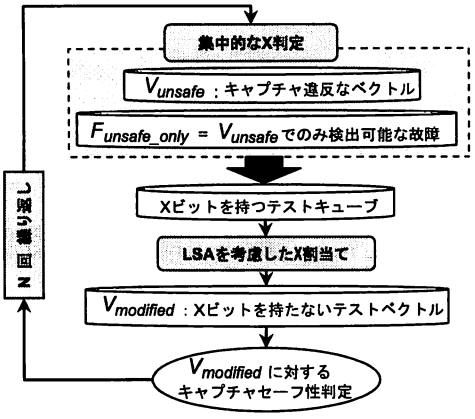

図 5　集中的 X 判定と低 LSA-X 判定フロー

4.3. キャプチャセーフテスト生成

フェーズ 2 での集中的 X 判定と低 LSA X 割当てを複数回繰り返す処理でも改善されないキャプチャ違反テストベクトル v に対して，その v でのみ検出される故障が存在しているならば，その故障を対象に低 LSA テスト生成する処理を実行する．低 LSA テスト生成とはキャプチャセーフなテストベクトルを生成する処理のことである．図 5 に低 LSA テスト生成フローの概要を表している．低 LSA テスト生成は，LSA に基づく動的圧縮の部分を除けば，一般の ATPG とほぼ同様の構造である．

4.3.1. 条件付きキャプチャセーフ性判定

LSA に基づいた動的圧縮において，X ビットを持つテストキューブ c に対してキャプチャセーフ性判定を利用する必要がある．しかし，X ビットを持たないテストベクトルでしかキャプチャセーフ性判定はできない．そのため，X ビットを持つ c に対して行うキャプチャセーフ性判定(図 6 の①)では，まず c の中の X ビットに対して低 LSA X 割当てを行い，テストベクトル v を得る．次に，v に対してキャプチャセーフ性判定を行う．この v に対するキャプチャセーフ性判定の結果を c に対する判定とみなす．この c に対するキャプチャセーフ性判定を条件付きキャプチャセーフ性判定と呼ぶ．

4.3.2. LSA に基づく動的圧縮

LSA に基づく動的圧縮について説明する．まず初期故障 f_p に対するテスト生成をし，X ビットを持つテストキューブ c_1 を得る．この生成された c_1 に対して，キャプチャセーフ性判定を行い，もしキャプチャ違反と判定されるならば f_p は検出不可能故障とし，f_p に対するテスト生成を中止する．これより故障検出率が低下する可能性がある．また，c_1 がキャプチャセーフと判定されたならば，2 次故障 f_s を選択し，c_1 の X ビットに f_s を検出可能にするよう論理値を割当て，テストキューブ c_2 とする．次に，c_2 に対してキャプチャセーフ性判定を行う．もしキャプチャセーフ性判定の結果がセーフであったならば，図 6 の②の処理から繰り返す．そして，キャプチャセーフ性判定の結果が違反であるならば，c_1 に戻し低 LSA X 割当てを行う．そのテストベクトルをキャプチャセーフなテストベクトルとする．

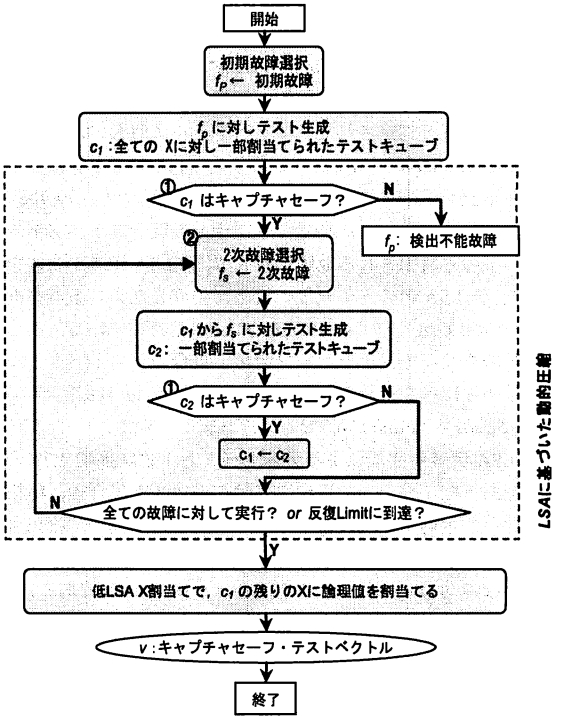

図 6　低 LSA テスト生成フロー

表 1　提案手法のフローの実験結果

Circuit	Init.Test Set					XID & X-Fill			Test Gen.		Final Test Set		
	Init. Fault Cov.	# of Detected faults	Safe Vec.	Unsafe Vec.		Unsafe Vec.		CPU time (sec.)	New Vec.	CPU time (sec.)	# of Vec.	# of Undet. Faults	Fault cov.
			# of Vec.	# of Vec.	# of Target Faults	# of Vec.	# of Target Faults						
s13207	79.5	11724	312	11	81	0	0	30	0	22	323	0	79.5
s15850	70.2	12316	195	26	258	0	0	24	0	28	221	0	70.2
s35932	82.5	44012	331	6	703	0	0	97	0	168	337	0	82.5
s38417	98	48024	201	69	1748	20	239	125	25	216	270	3	98
s38584	83.9	43699	241	171	3702	131	2995	258	189	1954	470	218	83.4
b15	89.9	36098	1115	26	325	4	52	203	9	90	1146	1	89.8
b17	92.6	106023	1231	19	240	4	59	682	8	642	1254	0	92.6
b20	89.8	37946	966	23	348	5	156	203	8	110	992	15	89.8
b21	89.9	39753	885	38	639	4	119	176	9	93	928	2	89.9
b22	89.9	60432	1086	37	888	15	498	446	25	308	1133	41	89.8

5. 実験結果

キャプチャセーフテスト生成フローの提案手法を 2.9GHz, 16GB メモリの計算機上で C 言語で実装し，ISCAS'89 の 5 回路と ITC'99 の 5 回路に対して実験を行った．キャプチャセーフ性判定では，1 つの FR 領域を 20 ゲートと設定し，LSA 解析は単位遅延シミュレーションで計算した．条件のリミットは初期テスト集合の最大値の 90%に設定し，故障モデルは遅延故障モデルを使用した．以上の条件で行ったキャプチャセーフテスト生成フローの実験の結果を表 1 に示す．

初期テスト集合における故障検出率，検出故障数，キャプチャセーフ判定によるキャプチャセーフとキャプチャ違反の分類結果をそれぞれ"*Init. Test Set*"下の"*Init. Fault Cov.*"，"*# of Detected Faults*"，"*Safe Vec.*"，"*Unsafe Vec.*"で示している．また"*Unsafe Vec.*"下の"*# of Target Faults*"は"*Unsafe Vec.*"でのみ検出される故障数を表している．初期テスト集合においてキャプチャ違反と判定されたテストベクトルに対して，集中的 X 判定と低 LSA X 割当てを 3 回繰り返し，それでもまだキャプチャ違反テストベクトルを"*XID & X-Fill*"下の"*Unsafe Vec.*"で表している．そして，まだキャプチャ違反テストベクトルによってのみ検出される故障に対して，低 LSA テスト生成を行った．低 LSA テスト生成後の新たに生成したテストベクトル数を"*Test Gen.*"下の"*# of Undet Faults*"で表している．そして，最終的なキャプチャセーフテスト集合のテストベクトル数，故障検出率をそれぞれ"*Test Gen.*"下の"# of *Vec.*"，"*# of Undet Faults*"，"*Fault Cov.*"でそれぞれ表している．

低 LSA テスト生成によるテストベクトル数は平均 1.7%の増加であった．また，故障検出率は初期テスト集合の故障検出率比べ，平均 0.03%の低下した．初期テスト集合で検出される故障のうち，最終キャプチャセーフテスト集合で検出されない故障"*# of Undet Faults*"の割合は平均で 0.1%であった．

6. まとめ

キャプチャセーフ性はテスト歩留まり低下を避けるために実速度スキャンテストにおいて要求されている．本論文では，高精度のキャプチャセーフ性チェック基準，及び X 判定と X 割当てによる効果的なキャプチャセーフテスト生成技術を提案した．提案手法の利点は回路修正とクロック変更の必要がないことである．実験結果により，提案手法の効果を示した．また、テストベクトル数の増加や故障検出率の低下が見られたが，許容範囲内に抑えることができた．

今後の課題としては，テストベクトル数の増加なしかつ故障検出率を維持したまま，キャプチャセーフなテスト集合を生成することが挙げられる．

参考文献

[1] M. Abramovici, M. Breuer, and A. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, 1990.

[2] L.-T Wang, et al.,(Editors), VLSI Test Principles and Architectures: Design for Testability, Elsevier, 2006.

[3] R. Sankaralingam, R. Orugani, and N. Touda, "Reducing Power Dissipation during Test Using Scan Chain Disable," *Proc. VLSI Test Symp.*, pp.319-324,2002

[4] O. Sinanoglu and A. Orailoglu, "Scan Power Minimization through Stimulus and Response Transformations," *Proc. Design, Automation and Test in Europe*, pp.404-409, 2004

[5] Y. Yoshida, M. Watati,"A New Approach for Low Power Scan Testing" *Proc. Intl. Test Conf.*,pp.480-487, 2003.

[6] S. Kajihara, K. Ishida, and K. Miyase, "Test Vector Modification for Power Reduction during Scan Testing," *Proc. VLSI Test Symp.*, pp.160-165, 2002.

[7] N. Ahmed, et al., "Transition Delay Fault Test Pattern Generation Considering Supply Voltage Noise in a SOC Design," *Proc. Design Automaton Conf.*, pp. 533-538, 2007.

[8] V. R. Devanathan, et al., "A Stochastic Pattern Generation and Optimization Framework for Variation-Tolerant, Power-Safe Scan Test," *Proc. Intl. Test Conf.*, Paper 13.1, 2007.

[9] K. Miyase and S. Kajihara, XID: Don't care identification of test patterns for combinational circuits. *IEEE Transactions on Computer-Aided Design* (2004), Vol 23, pp.321-326.

[10] S. Remersaro, et al., "Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs," *Proc. Int'l Test Conf., Paper* 32.2, 2006.

[11] X. Wen et. Al,"A Novel Scheme to Reduce Power Supply Noise for High-Quality At-speed Scan Testing," *Proc. Int'l Test Conf., Paper* 25.1, 2007.

[12] X. Lin, J. Rajski, I. Pomeranz, S. M. Reddy, "On Static Test Compaction and Test Pattern Ordering for Scan Designs," *Proc. Intl. Test Conf.*, pp. 1088-1097, 2001.

[13] S. Ravi, "Power-Aware Test: Challenges and Solutions," Proc. Intl. Test Conf., Paper 17.3,2006.