

複数機能的時間展開モデルを用いた 推定フィールドランダムテストビリティ向上のための X 割当てアルゴリズム

仲本 千騎[†] 細川 利典^{††} 吉村 正義^{†††}

[†]日本大学大学院 生産工学研究科 〒275-8575 千葉県習志野市泉町 1-2-1

^{††}日本大学 生産工学部 〒275-8575 千葉県習志野市泉町 1-2-1

^{†††}京都産業大学 情報理工学部 〒603-8555 京都府京都市北区上賀茂本山

E-mail: [†]cika25015@g.nihon-u.ac.jp, ^{††}hosokawa.toshinori@nihon-u.ac.jp,
^{†††}yoshimura.masayoshi@cc.kyoto-su.ac.jp

あらまし ミッションクリティカルなシステムに搭載されている VLSI は経年劣化による欠陥を検出するために、フィールドテストが必要とされている。電源オン/オフ時のような短時間に回路を網羅的にテストするために、非スキャンベースの組込み自己テスト手法が提案されている。また、RTL でのデータパスのテストビリティ尺度として、コントローラに与えられた状態信号系列とデータパスの回路構造に基づいて計算される推定フィールドランダムテストビリティ(EFRT)が提案されている。しかしながら、コントローラからデータパスに供給される制御信号系列にドントケア(X)が含まれている場合、その X 割当てによって EFRT が変動する。与えられた状態信号系列長で構造的記号シミュレーションを実行した後の最終時刻から順に機能的時間展開モデルによって EFRT を向上させる制御信号の X 割当て手法が提案されているが、まだ EFRT に課題が残る。本論文では、EFRT を改善するために、複数の機能的時間展開モデルを用いた制御信号系列に対する X 割当てアルゴリズムを提案する。

キーワード フィールドテスト, 組込み自己テスト, 悲観/楽観構造的記号シミュレーション, 状態信号系列, 制御信号系列, 推定フィールドランダムテストビリティ

A Don't Care Filling Algorithm to Improve Estimated Field Random Testability Using Multiple Functional Time Expansion Models

Kazuki NAKAMOTO[†] Toshinori HOSOKAWA^{††} Masayoshi YOSHIMURA^{†††}

[†]Graduate School of Industrial Technology Nihon University ,1-2-1 Izumityo, Narashino, Chiba 275-8575 Japan

^{††}College of Industrial Technology Nihon University ,1-2-1 Izumityo, Narashino, Chiba 275-8575 Japan

^{†††}Faculty of Information Science and Engineering Kyoto Sangyo University, Kamigamo Motoyama,
Kita-ku, Kyoto, Kyoto 603-8555 Japan

E-mail: [†]cika25015@g.nihon-u.ac.jp, ^{††}hosokawa.toshinori@nihon-u.ac.jp,
^{†††}yoshimura.masayoshi@cc.kyoto-su.ac.jp

Abstract Field testing is required for VLSIs on mission-critical systems to detect defects due to aging. Non-scan-based built-in self-test methods have been proposed to comprehensively test circuits in short periods of time such as when the power is turned on and off. As a testability measure for data paths at register transfer level, estimated field random testability (EFRT), which is calculated based on the status signal sequence given to the controller and the circuit structures of the data paths, was proposed. However, when control signal sequences supplied from controllers to data-paths contain don't-cares (X), the EFRT varies depending on the X assignment. An X filling method for control signal sequences has been proposed that uses a functional time expansion model in reverse order from the final time step after structural symbolic simulation is performed for a given status signal sequence length; however, EFRT still remains insufficient. In this paper, we propose an X filling algorithm for control signal sequences using multiple functional time expansion models for data-paths to improve EFRT.

Keywords field testing, logic built-in self-test, pessimistic /optimistic structural symbol simulation, status signal sequences, control sequences, estimated field random testability

1. まえがき

近年、超大規模集積回路(Very Large Scale Integrated circuits: VLSI)は社会基盤を支える重要な技術として、航空、宇宙、医療機器、自動車制御などのミッションクリティカルなシステムに広く搭載されている。これらのシステムでは、VLSIの故障が人命に関わる重大な事故や経済損失を引き起こす可能性があり、品質と信頼性に対する要求が高まっている。一方で、半導体技術の発展に伴う回路の複雑化、大規模化、高速化が進み、製造ばらつきや回路の経年劣化が問題となっている[1,2]。特に、システム稼働中の回路の経年劣化に対しては、従来のライフタイム予測や出荷前の信頼性試験および寿命試験では限界があり、各VLSIの実際の使用環境や動作条件の多様性によって、劣化進行の事前予測が困難である[2]。劣化による障害発生を防ぐ手段の一つとして、動作マージン設計が行われているが、回路の製造ばらつきや動作環境、使用年数などの最悪ケースを考慮して決定するため、動作マージンが過大となりVLSIの性能を犠牲にする可能性がある[1]。それゆえ、過大な動作マージンを避けるためにVLSIがシステムに搭載された後のフィールドテストが重要となる。

フィールドテスト手法として、スキャン設計[3]と組み込み自己テスト(Built-In Self-Test: BIST)[3]を組合せたスキャンベース BIST[4-6]が提案されている。BISTは、実動作速度でのテスト実行が容易であり、外部テスト装置が不要であるなどの利点[1,3]を持つ。しかしながら、スキャンベース BIST はシフト動作に時間を要するため、フィールドテストにおける限られた時間制約下では回路全体を網羅的にテストすることが困難である[6]。さらに、通常 DFT の実装は機能設計が完了し、論理合成が行われた後のゲートレベルで行われる。しかしながら、微細化された VLSI において、すでに性能・電力・面積(PPA)が最適化されているゲートレベル回路に変更を加えることは非常に困難である。そのた

め近年では DFT を上流工程である RTL(レジスタ転送レベル)へと前倒しするシフトレフト戦略の採用が進んでいる[7]。

前述の時間制約や設計手法の課題を背景として、RTL において非スキャン BIST の故障検出率向上のための設計手法[8-10]が提案されている。本論文では、フィールドテストのための制約は以下の 1)と 2)とする。
1)フィールドテストは非スキャンテストを行う。
2)データパス内にはテスト回路が挿入されない。

文献[8]では、BIST と非スキャンテスト手法を RTL で組み合わせることで故障検出率の向上を図っている。文献[9]では複数モジュールの並行テストによるテスト実行時間の短縮、文献[10]ではさらにハードウェアオーバーヘッドの低減が図られている。しかしながら、文献[8-10]は、いずれも製造テストを対象としており、前述のフィールドテストの制約 2)を満たしていない。フィールドテストの制約を満たすアプローチとして、文献[11]が提案されている。文献[11]では、RTL とゲートレベル回路の故障検出率との相関を示す評価尺度として推定フィールドランダムテストバリエーション(EFRT)が提案されている[11]。文献[12]では、コントローラの制御信号系列中にドントケア(X)が含まれている場合、テストバリエーションの向上に有利な論理値を割当てると仮定して算出する楽観 EFRT とテストバリエーションの向上に不利な論理値を割当てると仮定して算出する悲観 EFRT を示した。加えて、文献[12]では与えられた状態信号系列で構造的記号シミュレーション[11]を実行した後の最終時刻から順に機能的時間展開モデル[13]によって EFRT を向上させる制御信号系列の X 割当て手法が提案されている。しかしながら、文献[12]では、制御信号系列が複数の時刻にわたって共有される場合に生じる状態信号系列の衝突が考慮されていない。文献[12]の手法では最大時刻から順に X 割当てを行うが、ある X 割当てによって確定した制御信号値は、その制御信号系列を共有する他の時刻にも反映される。その結果、最大時刻を起点とした割当てが他の時刻における EFRT 向上の妨げとなる場合があり、衝突する状態信号系列間での割当て順序や割当て値を考慮しなければ EFRT を最大化できないという課題が残る。本論文では、この課題を解決するため、文献[12]で使用されていた機能的時間展開モデルに代えて、複数の状態遷移の衝突を考慮した複数機能的時間展開モデルを適用し、衝突を考慮した上で制御信号系列の X に対して EFRT が向上する論理値(0,1)の割当て手法を提案する。

2. 非スキャンベースフィールドテストアーキテクチャ

2.1. フィールドテスト手法の概要

前述のフィールドテストの制約を満たすために本

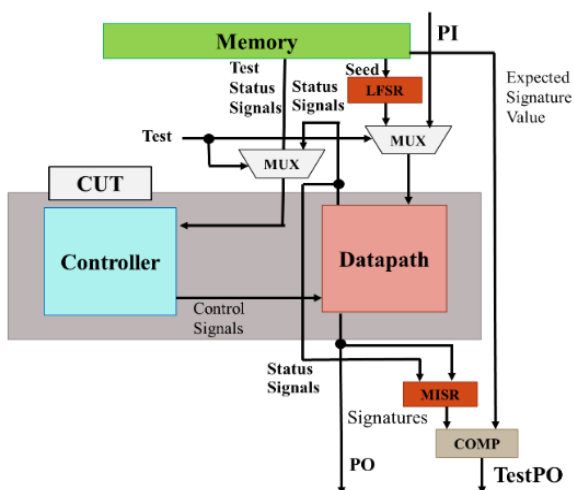


図 1. RTL フィールドテストアーキテクチャ

論文では文献[11]で提案された 3 つのアプローチを組合わせたテスト手法を採用する．1 つ目のアプローチとして，非スキャン回路でテストを実行する．これにより，フィールドテストを短時間で行うことができる．2 つ目のアプローチとして，決定的パターンとランダムパターンを併用する．これにより，小面積オーバーヘッドで故障検出率の向上が期待できる．3 つ目のアプローチとしてコントローラの n 回 k 連続状態遷移被覆[11]をフィールドテストに用いる．これにより，故障検出率の向上を図る．

2.2. RTL フィールドテストアーキテクチャ

図 1 に RTL フィールドテストアーキテクチャを示す．データパスと FSM で設計されたコントローラから構成される RTL 回路を研究対象とする．フィールドテスト手法を実現するために，LFSR の初期値となるシード値やシグネチャの期待値，状態遷移被覆によって生成した状態信号系列をメモリから与える．コントローラの入力は，データパスが出力する状態信号と，メモリに格納されているテスト用の状態信号とをマルチプレクサを用いて選択可能とする．データパスの状態信号線は故障の影響を容易に観測するために，MISR に接続する．データパスの入力は，本来の外部入力と LFSR が出力するランダムテストパターンとをマルチプレクサにより選択可能とする．データパスの出力は，外部出力と MISR への入力に分岐する．MISR はデータパスの出力を圧縮し，シグネチャとして期待値とともに比較器に入力される．比較器は，シグネチャとその期待値との比較結果によって正常信号もしくは異常信号をテスト用外部出力 TestPO へ出力する．フィールドテスト用に挿入したマルチプレクサの制御信号線は新たに付加した外部入力 Test と接続する．

3. 構造的記号シミュレーション

データパスの構造と制御信号系列に基づいて実行されるシミュレーション手法を構造的記号シミュレーション[11]という．構造的記号シミュレーションは，初期時刻から最大時刻に向けて外部入力，定数入力から制御可能であることを示す可制御シンボル(C シンボル)[11]を割当て，伝搬規則[11]に従って伝搬させる．次に，最大時刻から初期時刻に向けて観測可能であることを示す可観測シンボル(O シンボル)[11]を割当て，伝搬規則[11]に従って伝搬する．構造的記号シミュレーションはテスト可能な信号線を判定するために実行される．データパスはハードウェア要素とそれらを接続する信号線，及びコントローラへ出力する状態信号線から構成される．ハードウェア要素は多入力 1 出力の演算器，マルチプレクサ，レジスタ，外部入力，定数，外部出力，信号線，制御信号線から構成される．以下に，構造的記号シミュレーションで用いられる用語に

ついて定義する．

(定義 1：テスト可能な信号線)

時刻 t で，ある信号線に C シンボルと O シンボルが割当てられている時，その信号線を時刻 t でテスト可能な信号線という．

(定義 2：テスト実行回数)

データパスのある信号線がテスト可能である時刻数をその信号線のテスト実行回数という．

(定義 3：推定フィールドランダムテストバリティ)

データパス内の各ハードウェア要素のテスト実行回数から算出した推定検出故障数を総故障数で除した値を推定フィールドランダムテストバリティという．

(定義 4：楽観構造的記号シミュレーション)

楽観構造的記号シミュレーション[11]は，構造的記号シミュレーション[11]の処理とほぼ同じであり，あるハードウェア要素がテスト可能となるように X に論理値割当てされていることを仮定する．各状態遷移における X の論理値割当てにより，各ハードウェア要素のテスト実行回数を最大限に見積る．

(定義 5：悲観構造的記号シミュレーション)

悲観構造的記号シミュレーション[11]は，構造的記号シミュレーション[11]の処理とほぼ同じであり，あるハードウェア要素がテスト不可能となるように X の論理値割当てされていることを仮定する．各状態遷移における X の論理値割当てにより，各ハードウェア要素のテスト実行回数を最小限に見積る．

4. 推定フィールドランダムテストバリティ

以下に，EFRTで用いられる用語について定義する．

(定義 6：悲観 EFRT)

悲観構造的記号シミュレーションによって得られた EFRT を悲観 EFRT という．

(定義 7：楽観 EFRT)

楽観構造的記号シミュレーションによって得られた EFRT を楽観 EFRT という．

表 1. 楽観 EFRT と悲観 EFRT の比較

Circuit	n	1	1	2	2	3	3
	k	1	2	1	2	1	2
Kim	Len	37	38	77	78	118	120
	Opt	97.57	97.71	98.30	98.31	98.37	98.38
	Pes	82.61	82.76	92.41	92.57	95.31	95.38
	$\Delta O-P$	14.96	14.95	5.89	5.74	3.06	3.00
Maha	Len	55	115	110	116	165	192
	Opt	96.48	96.68	96.65	96.68	96.85	96.89
	Pes	90.42	95.68	95.58	95.68	96.58	96.66
	$\Delta O-P$	6.06	1.00	1.07	1.00	0.27	0.23
Sehwa	Len	57	59	116	140	187	188
	Opt	96.45	96.55	96.91	96.97	97.10	97.10
	Pes	83.07	84.27	89.12	89.63	90.33	90.41
	$\Delta O-P$	13.38	12.28	7.79	7.34	6.77	6.69
average $\Delta O-P$		11.47	9.41	4.92	4.69	3.37	3.31

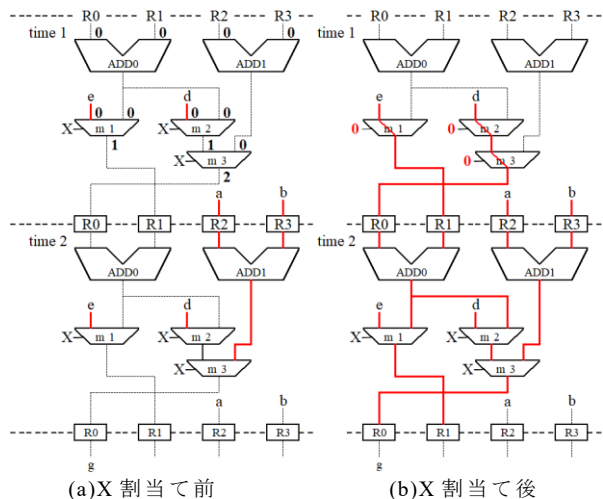


図 2. C シンボル伝搬のための X 割当て例

表 1 に各ベンチマーク回路に対する楽観 EFRT と悲観 EFRT の算出結果を示す. n 回 k 連続状態遷移被覆信号系列[11]を用いた時の系列長(Len), 楽観 EFRT(Opt), 悲観 EFRT(Pes)およびその差($\Delta O - P = Opt - Pes$)を比較する. 表 1 より, 1 回 1 連続状態遷移被覆である $n = 1$, $k = 1$ の条件を用いた場合, 楽観 EFRT と悲観 EFRT には平均で 11.47% の差が生じている. この差の原因は, 制御信号系列に含まれる制御値 X である. したがって, 本論文では, 悲観 EFRT を向上させるため, 制御信号系列中の X に対して適切に論理値(0,1)の割当てる手法を提案する.

5. EFRT 向上のための X 割当て手法

5.1. C シンボル X 割当て手法

初期時刻から最大時刻までの各時刻に対して C シンボルを伝搬させるための X 割当てを行う. マルチプレクサ m の制御値が X であり, m のいずれかの入力信号線に C シンボルが伝搬され, 出力信号線に C シンボルが伝搬されていない場合, m の出力信号線に C シンボルを伝搬できるような論理値を割当てて.

図 2 に ex1[14] の C シンボル伝搬のための X 割当て例を示す. 図 2(a) は X 割当て前, (b) は X 割当て後の例であり, 実線の信号線は C シンボルが伝搬されている信号線を表している. また, 図 2(a) の $time1$ では信号線のレベル付け例を示す. はじめに, マルチプレクサのレベル付けを行う. 外部入力, 定数入力, レジスタ出力と接続している信号線のレベルを 0 に初期化し, マルチプレクサを経由するごとに, 全入力信号線のうち, 最大レベルに 1 を加算した値をマルチプレクサの出力信号線のレベルとする. したがって, $time1$ の時, $m1$, $m2$ のレベルは 1, $m3$ のレベルが 2 となる.

C シンボルを伝搬させるための X 割当ては, マルチプレクサのレベルの昇順に X 割当てを実行する. $time1$ の時, レベルが 1 であるマルチプレクサ $m1$, $m2$ は第 0 入力信号線に C シンボルが伝搬されているが, 制御信

号値が X である. したがって, $m1$, $m2$ に制御信号値 0 を割当てることによってそれぞれの出力信号線に C シンボルが伝搬される. $m2$ の出力信号線に C シンボルが伝搬された結果, $m3$ の第 0 入力信号線に C シンボルが伝搬される. したがって, レベル 2 である $m3$ に制御信号値 0 を割当てることによって出力信号線に C シンボルが伝搬される. $time2$ では, $time1$ の X 割当てによってレジスタ $R0$, $R1$ に C シンボルが伝搬された結果, 外部入力, 定数入力に加えて $R0$, $R1$ から C シンボルが伝搬される. したがって, $time2$ の全てのマルチプレクサは全入力信号線に C シンボルが伝搬され, X 割当てが不要となる.

5.2. 複数機能的時間展開モデル

非スキャンフィールドテストにおける RTL データパスは, コントローラから印加される状態信号系列長分, 1 クロックサイクルを 1 時刻として時間展開される. O シンボルは複数時刻にわたって伝搬する場合があるため, 単一時刻のみを対象とした X 割当てと, 展開時間 ET 内の複数時刻にわたる伝搬を考慮した X 割当てでは伝搬経路が異なり, X 割当て後の制御信号系列も異なる[12]. さらに, 分岐を含む FSM において, 時間展開時の X 割当てを行う順序によって状態遷移の制約競合が生じる可能性がある. 例えば, ある時刻 t において, O シンボル伝搬のためマルチプレクサ m へ制御信号値 0 を割当てて一方, 同じ制御信号を参照する別の時刻 u において, マルチプレクサ m へ制御信号値 1 を割当てて場合, 制御信号の割当てに矛盾が生じる. 本論文では, ET 時間内での最適な伝搬経路を特定するとともに, 文献[15]で課題となっていた制約競合が生じた場合でも EFRT の向上が最大化されるような X 割当て手法を提案する.

5.3. 時間展開モデルを用いた O シンボル X 割当て手法

O シンボルを伝搬させるための X 割当てアルゴリズムを図 3 に示す. 本アルゴリズムでは構造的記号シミュレーション結果 SSS , コントローラ $Ctrl$, 展開時間 ET , 推定フィールドランダムテストバリエーション $EFRT$, データパス DP , 使用する状態信号系列の最大時刻 T_{max} を入力とする. 出力は制御信号系列の一部の X に論理値が割当てられたコントローラ $Ctrl$ である. 制御値が X であり, 出力信号線に O シンボルが伝搬されている X 割当て可能なマルチプレクサが存在しなくなるまで, 2 行目から 16 行目の処理を繰り返す(1 行目). 時刻 t が状態信号系列の最大時刻 T_{max} から 0 時刻目まで時刻を遡り, 5 行目から 13 行目までの処理を繰り返す(4 行目). 時刻 t のデータパス内のマルチプレクサのうち, 出力信号線に O シンボルが伝搬され, 制御値が X である O シンボル伝搬可能なマルチプレクサが存在するか確認する. 存在する場合, 6 行目から 12 行目の処理を行う(5

Algorithm Xfilling for Controller Using Time Expansion Model

Input : $SSS, Ctrl, ET, EFRT, DP, T_{max}$

Output : $Ctrl$

```

1. While (XFill_MUX ( SSS , DP , Ctrl ) != true ) do
2.   B_Ctrl = Ctrl ;
3.   B_EFRT = EFRT ;
4.   For  $t = T_{max}$  downto 0 do
5.     If ( Chk_XFill ( t , DP , Ctrl ) == true ) then
6.       Leveling ( t , ET , DP ) ;
7.       ( L_Ctrl , L_EFRT ) = Exp_XFill ( t , SSS , ET , EFRT ) ;
8.       If ( L_EFRT > B_EFRT ) then
9.         B_Ctrl = L_Ctrl ;
10.        B_EFRT = L_EFRT ;
11.      EndIf
12.      Rst_Level ( t , ET , DP ) ;
13.    EndIf
14.  EndFor
15.  Ctrl = B_Ctrl ;
16.  SSS = Calc_SSS ( DP , Ctrl ) ;
17. EndWhile
18. return Ctrl ;

```

図 3. O シンボル伝搬の X 割当てアルゴリズム

行目). $t-ET$ 時刻目から t 時刻目までの間でマルチプレクサのレベル付けを行う(6行目). t 時刻目から $t-ET$ 時刻目までの ET 時間内で X 割当てを行い, O シンボル伝搬可能な全経路を探索し, EFRT の増加量が最大となるような制御信号系列の X 割当てを行う[12](7行目). 7行目で算出した EFRT が現時点の B_EFRT よりも大きい場合, 9行目から 10行目の処理を行う(8行目). 7行目で算出した X 割当て済みコントローラを B_Ctrl に代入する(9行目). 7行目で算出した EFRT を B_EFRT に代入する(10行目). 6行目で行ったマルチプレクサのレベル付けをリセットする(12行目). $Ctrl$ に X 割当て済みコントローラ B_Ctrl を代入する(15行目). X 割当て済みコントローラを用いて構造的記号シミュレーションを実行し, SSSを更新する. X 割当てによってコントローラが更新されると C シンボル, O シンボルの伝搬が変化するため, SSS を再計算する必要がある(16行目). X 割当て済みコントローラを返す(18行目).

6. 実験結果

本実験では, RTL ベンチマーク回路 ex1, ex2, ex4, Kim, Maha, Schwa, ARF, BPF[14,15]を対象回路とするオリジナルコントローラに対して n 回 k 連続状態遷移被覆する状態信号系列を生成し, 楽観構造的記号シミュレーション, 悲観構造的記号シミュレーションを実行した. その結果, 得られた悲観 EFRT から複数機能的時間展開モデルを用いた制御信号系列の X 割当てを行い, X 割当て前後の EFRT の評価を行った.

表 2 に複数機能的時間展開モデルを用いた X 割当ての実験結果を示す. 1 列目に対象回路, 2 列目にコントローラに含まれる制御信号値 X の数, 3 列目, 4 列目

表 2. X 割当て結果

Circuit	#X	n	k	Len	Opt	Pes	Rand	DC	Heuristic Expanded Time				
									1	3	5	10	15
ex2	2	1	9	68.15	54.54	58.84	54.57		60.15	61.52	61.52	-	-
	61	3	1	13	76.77	71.60	72.99	71.61	73.64	74.65	74.65	74.65	-
	4	1	17	81.69	71.66	75.40	71.67		74.91	75.41	75.41	75.41	75.41
ex4	2	1	9	55.84	34.37	52.58	34.92		53.07	53.11	53.11	-	-
	27	3	1	13	72.36	48.95	69.62	49.60	69.39	69.50	69.50	69.63	-
	4	1	17	79.25	58.18	77.21	58.98		76.81	76.96	76.96	77.21	77.21
ARF	2	1	21	87.14	80.82	81.20	80.93		91.98	83.70	84.28	84.33	84.33
	394	3	1	31	93.43	87.35	87.42	87.46	88.15	90.65	90.83	90.84	90.84
	4	1	41	96.45	91.35	91.38	91.41		91.73	93.95	94.30	94.32	94.65
BPF	2	1	21	87.27	80.11	82.06	82.10		81.56	84.77	84.81	84.81	84.81
	254	3	1	31	93.17	86.75	88.60	88.85	87.69	91.28	91.28	91.28	91.28
	4	1	41	96.11	90.66	92.94	92.95		91.16	94.10	94.25	94.25	94.55
Kim	1	1	37	97.57	82.61	90.52	85.89		91.04	93.67	93.52	93.31	93.67
	1	4	41	97.88	83.34	91.88	86.46		91.65	94.71	94.04	94.88	94.84
	2	1	74	98.30	92.42	96.43	94.18		96.06	97.33	97.47	97.42	97.22
	2	2	78	98.10	92.57	95.31	94.49		96.47	97.49	96.37	97.64	97.64
	3	1	111	98.37	95.31	97.80	96.44		97.64	98.19	98.19	98.10	97.91
	3	2	120	98.38	95.38	96.96	96.61		97.72	97.81	98.00	97.01	98.06
Maha	1	1	55	96.48	90.42	92.95	92.14		94.42	95.16	95.18	95.19	95.21
	1	2	115	96.68	95.68	96.11	96.00		96.51	96.60	96.60	96.60	96.60
	2	1	110	96.65	95.58	95.98	95.89		96.42	96.47	96.50	96.50	96.50
	2	2	116	96.68	95.68	96.11	96.00		96.51	96.60	96.60	96.60	96.60
	3	1	165	96.85	96.58	96.85	96.62		96.85	96.85	96.85	96.85	96.85
	3	2	192	96.89	96.66	96.69	96.68		96.89	96.89	96.89	96.89	96.89
Schwa	1	1	57	96.45	83.07	93.51	90.34		91.95	94.18	94.20	94.20	94.25
	1	2	59	96.55	84.27	93.98	91.47		92.69	94.52	94.53	94.53	94.53
	2	1	116	96.91	89.12	95.52	95.66		95.33	96.49	96.45	96.46	96.45
	2	2	140	96.97	89.63	96.29	96.17		96.64	96.64	96.64	96.64	96.64
	3	1	187	97.10	90.33	96.66	96.63		95.96	97.02	97.02	97.02	97.02
	3	2	188	97.10	90.41	96.75	96.70		95.96	97.02	97.02	97.02	97.02

に n 回 k 連続状態遷移, 5 列目に状態信号系列長, 6 列目に楽観 EFRT, 7 列目に悲観 EFRT, 8 列目にオリジナルコントローラの制御信号系列に対して, 10000 回ランダムに X 割当てを行った時の最大の EFRT, 9 列目に Synopsys 社製 Design Compiler を用いて論理合成を行い, 生成されたゲートレベルネットリストから求めた, 制御値割当て結果を基に算出した EFRT を示す. 10 列目から 14 列目が時間展開モデルを用いた X 割当て結果であり, 2 行目に展開時間 ET を示す.

ex2 の 4 回 1 連続状態遷移では, $ET = 1$ の時 EFRT は 74.91%であったが, $ET \geq 3$ では X 割当ての探索空間が広がり, EFRT が 0.50%向上した. BPF の 4 回 1 連続状態遷移では, $ET = 1$ の時 EFRT は 91.16%であり, ランダム X 割当ての 92.94%より 1.78%低かったが, $ET = 3$ の時は 94.10%と EFRT が 2.94%向上し, ランダム X 割当てより 1.16%高くなった. また, $ET \geq 5$ の時は 94.25%, $ET = 15$ の時は 94.55%となった. Schwa の 3 回 1 連続状態遷移では, $ET = 1$ の時の EFRT は 95.96%であり, Design Compiler が論理合成時に割当てた X 割当て結果を用いた EFRT である 96.63%より 0.67%低くなった. $ET \geq 3$ では EFRT は 97.02%となり, Design Compiler の結果から 0.39%EFRT が高かった. 文献[15]では $ET = 5$ の場合の EFRT は 95.59%であり, ランダム割当てより 0.04%, EFRT が低かった. 提案手法では EFRT が 97.02%

表 3. 厳密解と提案手法の評価

Circuit	#X	n	k	Len	Opt	Pes	Exact Result		Heuristic Expanded Time			
									1		5	
							EFRT	Time(s)	EFRT	Time(s)	EFRT	Time(s)
ex1	7	2	1	5	79.11	61.36	72.49	35	67.30	0.07	72.49	0.07
		3	1	9	89.9	73.93	79.07	571	79.07	0.07	79.07	0.07

であり, FSM の衝突を考慮することでランダム X 割当てよりも EFRT を向上することができた. 実験に用いた全ての回路において, ランダムで 10000 回 X 割当てを行った最大の EFRT より $ET = 15$ における提案手法の X 割当ての最大値の方が高かった. 同様に, 全ての回路で Design Compiler が論理合成時に割当てた X 割当てから算出した EFRT より $ET = 15$ における提案手法の X 割当ての EFRT の方が高かった.

表 3 に提案手法のヒューリスティックアルゴリズムの評価結果を示す. 1 列目に対象回路, 2 列目にコントローラに含まれる制御信号値 X の数, 3 列目, 4 列目に n 回 k 連続状態遷移, 5 列目に状態信号系列長, 6 列目に楽観 EFRT, 7 列目に悲観 EFRT, 8 列目, 9 列目に ASP ソルバーClingo[16]を用いて算出した厳密解の EFRT と実行時間(s), 10 列目から 13 列目は提案手法の展開時間とその EFRT, 実行時間を示す. ex1 の 3 回 1 連続状態遷移では制御値 X のマルチプレクサが 10 個存在し, $2^{10} = 1024$ 通りの割当てが存在する. その結果得られた厳密解の EFRT は 79.07%で実行時間は 571 秒であった. 提案手法では, 展開時間 $ET = 1$ を用いた場合の EFRT が 79.07%であり, 実行時間 0.07 秒で厳密解と同等の EFRT を約 8157 倍高速に得ることができた. その他の回路では, 制御値 X のマルチプレクサ数が多く, 制御値 X の MUX が #X 個存在する場合 $2^{\#X}$ 通りの X 割当てが存在し, 現実的な時間で厳密解を算出することはできなかった. 一方, 提案手法では探索空間の増大に依らず, 短時間で EFRT の向上が見込める X 割当てを算出することができるため, 厳密解の算出が困難な規模の回路に対しても, 提案手法は有効である.

7. むすび

本論文では, 非スキャンベースのフィールドテストにおいて, n 回 k 連続状態遷移被覆する状態信号系列と複数機能的時間展開モデルを用いた EFRT を向上させるための制御信号系列の X 割当て手法を提案した. 時間展開 ET を $ET=1$ から $ET=15$ へ探索空間を広げることで EFRT を向上させることができた. 実験結果より, ランダム X 割当てと比較して平均 1.16%, Design Compiler による論理合成時の X 割当てと比較して平均 3.72%EFRT を向上させることができた. また, 厳密解と比較した評価において, 提案手法は厳密解と同等の EFRT を約 8157 倍高速に得ることができ, 厳密解の算出が困難な規模の回路に対しても, 提案手法は有効であることを確認した.

今後の課題として, 楽観 EFRT が 100%の状態信号系列を用いることで EFRT をさらに向上させることが挙げられる.

文 献

- [1] 藤原秀雄, “デジタルシステムの設計とテスト,” 学術図書株式会社, 2004.
- [2] I. Moghaddasi, S. Gorgin and J. Lee, “Dependable DNN Accelerator for Safety-Critical Systems: A Review on the Aging Perspective,” IEEE Access, vol.11, pp. 89803–89834, 2023.
- [3] L. Wang, C. Wu and X. Wen, “VLSI Test Principles and Architectures: Design for Testability,” Morgan Kaufmann Publishers, 2006.
- [4] T. Strauch, “Non-interfering Online and In-field SoC Testing,” 2024 IEEE International Workshop on Rapid System Prototyping, 2024.
- [5] I. Pomeranz, “Chip Aging and Double Transition Faults,” IEEE International Test Conference, 2025.
- [6] I. Pomeranz, “Varying Periods of In-Field Testing With Storage and Counter Based Logic Built-In Self-Test,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2025.
- [7] H. Iwata, Y. Maeda, J. Matsushima, O. Laouamri, N. Khanna, and J. Mayer, “A New Framework for RTL Test Points Insertion Facilitating a “Shift-Left DFT” Strategy,” IEEE International Test Conference, 2023.
- [8] A. Ahmadi, N. Paraman, M. S. Rusli and S. Issak, “Hybrid non scan with built-in self-test for fault coverage improvement,” AIP Conference, 2023.
- [9] K. Yamaguchi, H. Wada, T. Masuzawa, and H. Fujiwara, “A BIST Method Based on Concurrent Single-Control Testability of RTL Data Paths,” Proceedings 10th Asian Test Symposium, 2001.
- [10] K. Yamaguchi, M. Inoue, and H. Fujiwara, “Hierarchical BIST: Test-Per-Clock BIST with Low Overhead,” Electronics and Communications in Japan, 2007.
- [11] Y. Toyooka, H. Watanabe, T. Hosokawa, and M. Yoshimura, “An Evaluation of Estimated Field Random Testability for Data Paths at Register Transfer Level Using Status Signal Sequences Based on k-Consecutive State Transitions for Field Testing,” IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, 2023.
- [12] 仲本千騎, 細川利典, 吉村正義, “レジスタ転送レベルにおけるデータパスの時間展開モデルを用いた推定フィールドランダムテストビリティ向上のための制御信号系列のドントケア割当て手法,” 信学技報, vol. 125, no.143, DC2025-40, pp. 93-98, 2025
- [13] T. Masuda, J. Nishimaki, T. Hosokawa, and H. Fujiwara, “A Test Generation Method for Data Paths Using Easily Testable Functional Time Expansion Models and Controller Augmentation,” Asian Test Symposium, 2015.
- [14] M. T-C. Lee, “High-Level Test Synthesis of Digital VLSI Circuits,” Artech House Publishers, 1997.
- [15] S. P. Mohanty, N. Ranganathan, E. Kougianos, and P. Patra, ‘Low-Power High-Level Synthesis for Nanoscale CMOS Circuits’, Springer, 2008.
- [16] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub, “Conflict-Driven answer set solving,” Artificial Intelligence, 2012.