

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アルゴリズム

日大生産工（学部）○仲本 千騎 日大生産工 細川 利典 京産大 吉村 正義

1. まえがき

近年、超大規模集積回路(Very Large Scale Integrated circuits: 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、自動車制御などの高い信頼性が要求されるものにも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、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ついては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、各 VLSI 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。劣化による障害発生を防ぐ手段の一つとして、現状では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、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]。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。

スキャン設計[3]と組み込み自己テスト(Built-In Self-Test: BIST)[4]を組合せたスキャンベース BIST に基づくフィールドテスト手法[5-7]が提案されている。BIST は、他のテスト手法と比較して、実動作速度テストの適用が容易、テスト生成ツールの必要性の排除など、いくつかの利点[1,8]があるが、スキャンベース BIST に基づくフィールドテストではテスト実行時間の増大が課題として挙げられる。フィールドテストは電源オン/オフの短い時間で VLSI を網羅的にテストすることが理想である。そのため、データパスとコントローラから構成されるレジスタ転送レベル(Register Transfer Level: RTL)回路に対する非スキャン設計ベースのフィールドテストに焦点を当て、故障検出率の改善、小面積化、テスト実行時間の短縮を実現する手法が提案されている[9-12]。文献[9]では、故障検出率の向上を図るため、コントローラにおける n 回状態遷移被覆[9]を用

いて状態信号系列を生成している。しかしながら、文献[9]ではデータパスの動作に着目していないため、データパス中のハードウェア要素(レジスタ、マルチプレクサ、演算器、信号線)のテストを十分に行うことができていない。また、文献[10]ではこれらの課題を解決するために、コントローラの k サイクル連続で実行される状態遷移をすべて n 回列挙し、それらをすべて実行可能であり、かつできる限り短い状態信号系列を生成する n 回 k 連続状態遷移被覆信号系列を用いたフィールドテスト手法が提案され、さらに RTL でのフィールドランダムテストビリティの評価尺度として、テスト可能率が提案された。しかしながら、テスト可能率[10]と論理回路の故障検出率との相関を評価した結果、強い正の相関関係は認められなかった。それゆえ、文献[11]では、新たな評価尺度として推定フィールドランダムテストビリティが提案された。与えられた状態信号系列から生成される制御信号系列とレジスタ転送レベルでのデータパスの構造を用いた構造的記号シミュレーション[11]が実行され、RTL における推定フィールドランダムテストビリティが算出される。

本論文では、文献[12]で評価された悲観構造的記号シミュレーションによる低い推定フィールドランダムテストビリティを改善させるような制御信号のドントケア(X)割当て手法を提案する。

本論文は以下のように構成されている。第2章では非スキャンベースのフィールドテストのためのアーキテクチャを説明する。第3章では構造的記号シミュレーションを用いたデータパスの推定フィールドランダムテストビリティを説明する。第4章では制御信号のドントケア割当てアルゴリズムを提案する。第5章では高位設計ベンチマーク回路 kim[13]を用いた実験結果を示す。最後に第6章で、結論と今後の課題について述べる。

2. 非スキャンベースフィールドテスト

2.1. 非スキャン設計

一般に、現状のフィールドテストではスキャンテストが用いられることが多い。これは、テストパターンとしてランダムパターンを用いた場合でも、ある程度高い故障検出率が保証されるためである。しかしながら、シフト動作時には各スキャンフリップフロップ(Flip-Flop: FF)に値を設定/観測するため、シフト動作によるテスト実行時間の増大という問題点が挙げられる。この理由から、本論文では、文献[9-12]と同様に非スキャンテストによるフィールドテストを対象とする。また、文献[9-12]で述べられている非スキャンテストの問題点である故障検出率を改善する手法を 2.2 節で説明する。

2.2. RTL フィールドテストアーキテクチャ

非スキャンベースのフィールドテスト手法を実現するために、RTL 回路にテスト容易化設計(Design for

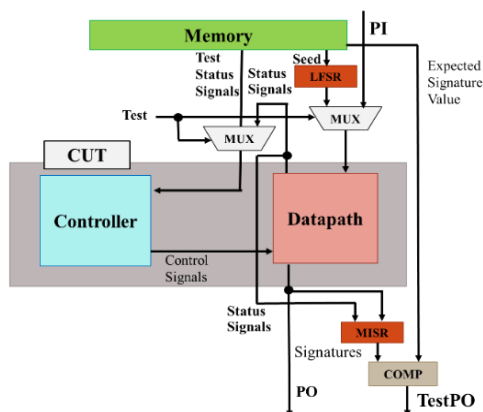


図1. フィールドテストアーキテクチャ

Testability : DFT)を行う。図 1 に本論文で用いる RTL 回路の非スキャンベースのフィールドテストアーキテクチャを示す。図 1 における CUT は、コントローラとデータパスである。また、擬似乱数パターン発生器 (Pseudo Random Pattern Generator : PRPG) としてフェーズシフト付 [15] 線形帰還型シフトレジスタ (Linear Feedback Shift Register : LFSR) [14], テスト応答圧縮回路 (Test Response Compactor : TRC) として他入力シグネチャレジスタ (Multiple Input Shift Register : MISR) [14] を用いている。さらに、LFSR の初期値となるシード値やシグネチャの期待値、 n 回 k 連続状態遷移被覆に基づき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、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に伝搬する故障影響を容易に観測するために、状態信号線を MISR に入力する。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を用いた推定フィールドランダムテストバリエーション算出

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を構造的記号シミュレーションと呼ぶ [10]。構造的記号シミュレーションは、すべての時刻の外部入力、定数から可制御であることを示す C シンボル [10] を割当て、伝搬規則 [10] に従って伝搬させ、その後外部出力から入力方向に向けて可観測であることを示す O シンボル [10] を割当て、伝搬規則 [10] 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実行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、制御信号線がある。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 : 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 [12] は構造的記号シミュレーション [10] の処理とほぼ同じであり、あるハードウェア要素の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

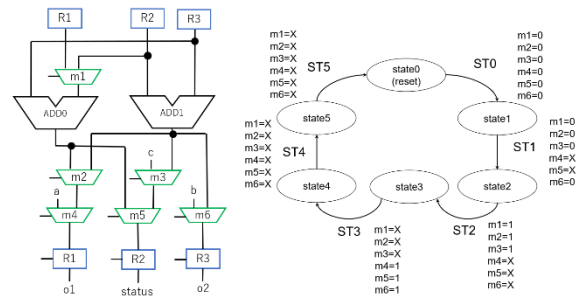


図 2. 例題 RTL 回路

ける X の論理値割当てにより、テスト可能なハードウェア要素を最大限に見積る。

(定義 4 : 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 [12] は構造的記号シミュレーション [10] 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、テスト可能なハードウェア要素を最小限に見積る。

本論文では、悲観構造的記号シミュレーションの結果を楽観構造的記号シミュレーションの結果に近づけるような X 割当てを行う。

4. 構造的記号シミュレーションを用いたドントケア割当て

4.1. 楽観構造的記号シミュレーションと悲観構造的記号シミュレーションのテスト実行回数の比較

本節では、楽観構造的記号シミュレーション [12] 及び悲観構造的記号シミュレーション [12] のテスト可能なハードウェア要素例を示す。図 2 にデータパスとコントローラの RTL 回路例を示す。表 1 に図 2 で示した回路例に対応した楽観構造的記号シミュレーションと悲観構造的記号シミュレーションのテスト可能表を示す。表 1 のテスト可能表は、1 回 1 連続状態遷移を得られ、被覆する状態信号系列から各状態遷移の各ハードウェア要素のテスト可能性の情報を示す。1 列目に状態遷移、2 列目に ADD0 の出力信号線のテスト可能性、3 列目に ADD1 の出力信号線のテスト可能性、4 列目に R1 の出力信号線のテスト可能性、5 列目に R2 の出力信号線のテスト可能性、6 列目に R3 の出力信号線のテスト可能性を示す。

テスト可能表の ADD0 に着目すると、楽観構造的記号シミュレーションでは ST1~ST4 まで 4 回テスト可能であるが、悲観構造的記号シミュレーションでは 1 度もテスト可能でない。ADD1 は楽観構造的記号シミュレーションでは ST2~ST4 まで 3 回テスト可能であるが悲観構造的記号シミュレーションでは 1 回のみテスト可能である。R2 も楽観構造的記号シミュレーションでは ST1~ST5 まで 5 回テスト可能であるが、悲観

表 1. テスト可能表

	ADD0	ADD1	R1	R2	R3
ST0			O		O
ST1	O		O	O	O
ST2	O	O	O	O	O
ST3	O	O	O	O	O
ST4	O	O	O	O	O
ST5			O	O	O

楽観構造的記号シミュレーション

	ADD0	ADD1	R1	R2	R3
ST0			O		O
ST1			O		O
ST2			O	O	O
ST3		O	O		O
ST4			O		O
ST5			O		O

悲観構造的記号シミュレーション

構造的記号シミュレーションでは1回のみテスト可能である。このように、楽観構造的記号シミュレーションと悲観構造的記号シミュレーションではテスト可能なハードウェア要素数が異なる。したがって、悲観構造的記号シミュレーションの制御信号がXのマルチプレクサにCシンボルやOシンボルを伝搬できるような制御値を割当ててすることで、ある時刻においてテスト可能なハードウェア要素数を増加させ、推定フィールドランダムテストバリエーションの向上を図る。

4.2. CシンボルX割当て手法

本節ではCシンボルを伝搬させるためのX割当て手法及び例を説明する。

マルチプレクサmの制御信号がXであり、mの入力信号線に1つ以上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、制御信号にCシンボルが伝搬するような制御値を割当てる。Cシンボルを伝搬可能な全てのマルチプレクサに対してX割当てが完了したら、構造的記号シミュレーションを再度行う。

具体例を示す。図2のRTL回路において、マルチプレクサm2の制御信号がXであり、ADD0の出力信号線に接続されている0番目の入力信号線にCシンボルが伝搬されており、1番目の入力信号線にはCシンボルが伝搬されていないとき、m2の制御信号に0番目の入力信号線のCシンボルを出力信号線に伝搬させるために制御信号を0に割当てる。m2の0番目の入力信号線にCシンボルが伝搬されておらず、1番目の入力信号線にCシンボルが伝搬されている場合、制御信号に1を割当てる。

4.3. OシンボルX割当て手法

本節では、制御信号がXのマルチプレクサに対して制御値を割当て、Oシンボルを伝搬させるX割当ての手法、例を示す。

Algorithm *Don't Care Filling to Control Signals of a Multiplexor*

for O Symbol Using Estimated Field Random Testability

Input : $EFRT, SSS, T_L, H, control$

Output : control

```

1.  $M_{list} = Mux\_Output\_Testable(H, SSS);$ 
2. for (each  $i \in M_{list}$ ) then
3.    $value = Mux\_Control\_Signal(i, Mlist);$ 
4.   if ( $value == 'X'$ ) then
5.      $A_{list} = Create\_EFRT\_ALU\_List(H, EFRT);$ 
6.     for (each  $j \in A_{list}$ ) then
7.        $signal = Able\_Connect\_ALU(j, M_{list}, A_{list});$ 
8.       if ( $signal == true$ ) then
9.          $control = X\_Filling\_Mux(M_{list}, A_{list}, H);$ 
10.        break;
11.      endif
12.    endfor
13.  endif
14.   $SSS = Calculate\_SSS(H, control)$ 
15.   $EFRT = Calculate\_EFRT(H, T_L, SSS, control);$ 
16. endfor
17. return control;
18. End
```

図3. Oシンボル伝搬アルゴリズム

図3は、ある時刻tのOシンボルの伝搬アルゴリズムである。まず、このアルゴリズムでは入力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 と、構造的記号シミュレーション結果 SSS、ハードウェア要素のテストライブラリ集合 T_L 、データパスのハードウェア要素の集合 H、コントローラ control を入力とし、出力は制御信号中の一部のXに論理値が割当てられたコントローラ control である。

X割当てを行う可能性があるマルチプレクサの集合 M_{list} を作成する。具体的には、ハードウェア要素の集合 H から出力信号線が時刻 t でテスト可能であるマルチプレクサのリストを作成する(1行目)。 M_{list} の各マルチプレクサ i について、3行目から15行目の処理を繰り返し実行する(2行目)。変数 value にマルチプレクサ i の制御信号値を代入する(3行目)。value が X の場合は、5行目から12行目の処理を行う(4行目)。H中の演算器から演算器の集合 A_{list} を作成する。 A_{list} に格納された演算器は推定フィールドランダムテストバリエーションをキーとして昇順にソートされる。(5行目)。 A_{list} の各演算器 j について、7行目から11行目の処理を行う(6行目)。マルチプレクサ i から演算器 j にOシンボルを伝搬可能である場合、signal に true、可能でなければ false を代入する(7行目)。signal が true の時、8行目から11行目の処理を行う(8行目)。マルチプレクサ i、マルチプレクサ i から演算器 j へ伝搬可能な経路にあるマルチプレクサに対して、i から j にOシンボルを伝搬できるような制御値を割当てる(9行目)。for文を抜け出す(10行目)。構造的記号シミュレーションを再度行う(14行目)。推定フィールドランダムテストバリエーションを再計算する(15行目)。X割当てを行った制御信号系列が格納された control を返す(17行目)。

以下に、Oを伝搬させるX割当ての例を図2のRTL回路を用いて説明する。

図2のRTL回路のマルチプレクサ m5, m3 の制御信号がXであり、m5の出力信号線がテスト可能であり、ADD0よりADD1の推定フィールドランダムテストバリエーションが低い時の状況について説明する。まず、 M_{list} に制御信号がXかつ出力信号線がテスト可能であるマルチプレクサ m5 を代入する。次に、m5の制御値であるXをvalueに代入する。valueはXであるため、 A_{list} にADD1, ADD0の順で推定フィールドランダムテストバリエーションが昇順になるような演算器のリストを作成する。m5からADD1へm3を経由してOシンボルを伝搬可能であるため、signal に true が代入され、m5に制御値1, m3に制御値1を代入する。構造的記号シミュレーションを再度行い、推定フィールドランダムテストバリエーションを再計算する。X割当てを行ったコントローラ control を返す。

5. 実験結果

本実験では、kim[13]を対象回路とする。オリジナルコントローラに対して1回1連続状態遷移被覆する状態信号系列を生成した。その後、楽観構造的記号シミュレーション及び悲観構造的記号シミュレーションを実行し、各ハードウェア要素のテスト実行回数を求めた。演算器のテスト実行回数を向上させるため、制御信号のX割当てを行い、その結果どの程度悲観構造的記号シミュレーション実行後のテスト実行回数を楽観

構造的記号シミュレーション実行後のテスト実行回数に近づけられるかを算出した。

表 2 にベンチマーク RTL 回路 kim に含まれる各ハードウェア要素のテスト実行回数及び推定フィールドランダムテストタビリティを示す。1 行目に各ハードウェア要素, 2 行目に各ハードウェア要素の推定フィールドランダムテストタビリティが 100%, すなわち完全なテストを実現するためのテスト実行回数, 3 行目に楽観構造的記号シミュレーションを用いた場合の各ハードウェア要素のテスト実行回数と推定フィールドランダムテストタビリティ, 4 行目に悲観構造的記号シミュレーションを用いた場合の各ハードウェア要素のテスト実行回数と推定フィールドランダムテストタビリティを示す。

表 2 の実験結果より, 演算器の推定フィールドランダムテストタビリティを上げるような X 割当てをすることによって SUB1, ADD1 の推定フィールドランダムテストタビリティをそれぞれ 30.09%, 28.37%向上できた。また, 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, X 割当て後の kim 回路全体の推定フィールドランダムテストタビリティはそれぞれ 95.20%, 81.00%, 82.24%だった。したがって, X 割当てすることによって, 悲観構造的記号シミュレーションの推定フィールドランダムテストタビリティが 1.24% 向上し, 楽観構造的記号シミュレーションの推定フィールドランダムテストタビリティに近づけることができた。しかしながら, SUB1, ADD1 以外の推定フィールドランダムテストタビリティは向上しなかった。また, X 割当て前の全時刻の X 数は 564 個であり, X 割当てすることで X 数は 542 個に減少した。

6. むすび

本論文では 1 回 1 連続状態遷移被覆する状態信号系列を用いて, 楽観及び悲観構造的記号シミュレーションを実行後のテスト実行回数を基に X 割当てを行い, テスト実行回数と推定フィールドランダムテストタビリティを算出することで X 割当て手法の評価をした。算出した推定フィールドランダムテストタビリティを基に X 割当てすることによって, 推定フィールドランダムテストタビリティを向上することができた。しかしながら, 一部の演算器の推定フィールドランダムテストタビリティしか向上させることができなかった。したがって, 演算器だけではなく, レジスタやマルチプレクサなど, 他のハードウェア要素の推定フィールドランダムテストタビリティも考慮した X 割当てアルゴリズムを検討する必要がある。

今後の課題としては, 複数のベンチマーク回路を用いて実験することで本アルゴリズムの有用性を評価することが挙げられる。

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] W. Wang, et al., “Compact Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology,” IEEE Trans. on Device and Material

表 2. kim 回路の実験結果

		各ハードウェア要素									
		R1	R2	R3	R4	R5	R6	SUB0	SUB1	ADD0	ADD1
完全なテストのための テスト実行回数		8	8	8	8	8	8	44	-	106	106
1回1連続 状態遷移 被覆 (37系列)	楽観	33 (100)	33 (100)	22 (100)	23 (100)	8 (100)	22 (100)	18 (94.82)	6 (69.11)	18 (97.12)	9 (90.54)
	悲観	31 (100)	28 (100)	8 (100)	19 (100)	5 (96.88)	3 (92.19)	10 (87.92)	1 (36.31)	15 (95.13)	2 (55.09)
	X割当て後	31 (100)	28 (100)	8 (100)	19 (100)	5 (96.88)	3 (92.19)	10 (87.92)	4 (66.40)	15 (95.13)	6 (83.46)

Reliability, Vol.7, No.4, pp.509-517, 2007.

- [3] H. Fujiwara, Logic Testing and Design for Testability, The MIT Press, 1985.
- [4] Edward J. McCluskey, “Built-In Self-Test Techniques,” IEEE Design & Test of Computers, vol.2, no.2, pp.21-28, April 1985.
- [5] J.A.Waicukauski, E.Lindbloom, B.L.Rosen, and V.S.Iyengar, “Transition fault simulation” IEEE Design and Test of Computers, Vol.4, pp.32-38, 1987
- [6] S Grosh, S Bhunia, A Raychowdhury and K Roy “Delay Faults Localization in Test-Per-ScanBIST” 12th IEEE IOLTS’06, July 2006.
- [7] F.Yang, S.Chakravarty, N Devta-Pasanna, S.M.Reddy and I. Pomeranz “An Enhanced Logic Architecture for Online Testing” 14th IEEE IOLTS’08, July 2008.
- [8] M. Abadir and M. Breuer, “Constructing optimal test schedules for VLSI circuits having built-in test hardware,” in Proc. Int. Symp. Fault-Tolerant Comput., pp. 165–170, June 1985.
- [9] 池ヶ谷祐輝, 石山悠太, 細川利典, 吉村正義, “*n* 回状態遷移被覆に基づく非スキャンオンラインテスト法,” vol. 119, no. 247, DC2019,2019 年 10 月.
- [10] 豊岡雄大, 渡辺悠樹, 細川利典, 吉村 正義 “フィールドテストのための *k* 連続状態遷移に基づく状態信号系列を用いたフィールドランダムテストタビリティの評価,” 信学技報, vol.122, no.134, DC2022-4,pp.19-24, 2022 年 7 月.
- [11] Yudai Toyooka, Haruki Watanabe, Toshinori Hosokawa, Masayoshi 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”, 2023 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), pp.1-6, Oct 2023.
- [12] 豊岡雄大, 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推定フィールドランダムテストタビリティ改善のための追加状態信号系列生成について” 信学技報, vol.123, no.389, DC2023-102,pp.47-52,2024 年 2 月.
- [13] M. T. -C. Lee, “High-Level Test Synthesis of Digital VLSI Circuits,” Artech House Publishers, 1997.
- [14] Michael L. Bushnell and Vishwani D. Agrawal. “Essentials of Electronic Testing for Digital, Memory & Mixed -Signal VLSI Circuits,” Kluwer Academic Publisher, 2000.
- [15] P. H. Bardell, “Design considerations for parallel pseudorandom pattern generators,” *J. Electron. Testing: Theory Applicat.*, vol. 1, no. 1, pp.73–87, 1990.