

パーシャルスキャン設計とコントローラ拡大を用いた 遷移故障用テスト容易化機能的 k 時間展開モデル生成法

日大生産工(院) ○石山 悠太 日大生産工 細川 利典
日大生産工(院) 池ヶ谷 祐輝

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、高速化が急速に進展している[1]。それに伴い、LSIのテストの高品質化と低コスト化が重要な課題となっている。高い故障検出効率を達成するためには、何らかのテスト容易化設計(Design for Testability : DFT)が必要である。

現在、スキャン設計[1]がLSIのDFT手法として普及している。特に回路中のすべてのフリップフロップ(Flip-Flop:FF)をスキャンFFで構成するフルスキャン設計では、すべてのFFが可制御・可観測となるため、故障モデルに依らず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や、テスト実行時間の増大などの問題点がある。これらのフルスキャン設計の問題点を解決するために、縮退故障[1]モデルのためのレジスタ転送レベル(Register Transfer Level : RTL)回路におけるDFT手法が提案されている[2]。

文献[2]では、データパスのテスト容易な構造に着目したテスト容易化機能的 k 時間展開モデル[2]を用いた、パーシャルスキャン設計に基づくDFT手法とテスト生成手法が提案されている。回路全体において高い故障検出効率を達成するために、コントローラのDFTとデータパス中のすべてのハードウェア要素(演算器、マルチプレクサ、レジスタ)に対応するテスト容易化機能的 k 時間展開モデルを生成し、その生成されたモデルの動作を実行可能にするために、有限状態機械(Finite state machine:FSM)で設計されたコントローラを拡大する。特に、文献[2]ではコントローラの無効状態にテスト用の新たな状態遷移を設計する。そのコントローラ中の状態レジスタをスキャン設計することで無効状態を含む任意の状態にシフト動作で遷移することが可能となり、任意の状態から k サイクル間状態遷移を実行することで、各ハードウェア要素に対するテスト容易化機能的 k 時間展開モデルの動作をテスト生成において容易に探索可能としている。文献[2]の手法により、フルスキャン設計における縮退故障の故障検出効率を維持したまま、面積オーバーヘッドとテスト実行時間の削減を可能とした。しかしながら、近年のLSIの微細化や高速化に伴い、縮退故障モデルのテストだけでなく、遷移故障モデル[1]などのタイミング欠陥に対するテストが重要である。

それゆえ、本論文では、文献[2]の手法に基づく遷移故障検出のためのテスト容易化機能的 k 時間展開モデル生成法を提案する。文献[2]で提案されたテスト容易

化機能的 k 時間展開モデルを、遷移故障を検出するためのモデルに拡張し、文献[2]と同様にフルスキャン設計の故障検出効率を維持したまま、面積オーバーヘッドとテスト実行時間の削減を目指す。

2. パーシャルスキャン設計とコントローラ拡大を用いたテスト容易化設計法

2-1 縮退故障モデルのためのテスト容易化機能的 k 時間展開モデル生成

テスト容易化機能的 k 時間展開モデル(Easily Testable Functional k -Time Expansion Models : ETF k -TEM)[2]とは、データパスのテスト容易な構造に着目して生成された k サイクルテスト生成モデルである。従来手法[2]では、回路全体の縮退故障を検出するためのETF k -TEMの生成法を提案した。ETF k -TEMは、データパス中のすべてのハードウェア要素(演算器、マルチプレクサ、レジスタ)をテストできるように生成する必要がある。図1にデータパス例を示し、図2に図1のADD0をテスト可能にするETF3-TEMの例を示す。

図1において、 $i1$, $i2$ は外部入力、 $o1$ は外部出力、 $R1$, $R2$, $R3$ はホールド機能付レジスタ、 $M1$ ~ $M5$ はマルチプレクサ、 $m1$ ~ $m5$, $r1$ ~ $r3$ は制御信号線、 $ADD0$ は加算器、 $SUB0$ は減算器である。また、図2において $t1$ ~ $t3$ は時刻を表す。ETF k -TEMにおいて、外部入力または時刻1のスキャンレジスタからテスト対象ハードウェア要素の入力に何らかの値を伝搬でき、そのテスト対象ハードウェア要素の出力から何らかの値を時刻 k の外部出力またはスキャンレジスタの入力まで伝搬できるものを、そのETF k -TEMで動作可能であるという[2]。ここで、多入力のハードウェア要素の左から i 番目の入力を入力 i ($0 \leq i \leq n-1$, n は入力数)と表す。図2のETF3-TEMにおいて、 $ADD0$ の入力0と入力1、 $R1$, $R2$, $M1$ の入力0、 $M1$ の入力1、 $M2$ の入力0、 $M3$ の入力0、 $M4$ の入力1が動作可能である。あるハードウェア要素が動作可能なとき、そのハードウェア要素は縮退故障テスト可能であると呼ぶ。

表1に、図2のETF3-TEMの動作を実行するためのテスト動作制御・状態信号系列[2]を示す。表中に示す $r1$ ~ $r3$, $m1$ ~ $m5$ は図2中の制御信号線に対応し、 $t1$ ~ $t3$ は図2中の時刻 $t1$ ~ $t3$ に対応している。表中の値は、時刻 $t1$ ~ $t3$ における制御信号線 $r1$ ~ $r3$, $m1$ ~ $m5$ の制御信号値(0,1,X)である。表1に示す3時刻の制御信号系

A Generation Method of Easily Testable Functional k Time Expansion Model
for a Transition Fault Model Using Controller Augmentation and Partial Scan Designs

Yuta ISHIYAMA, Toshinori HOSOKAWA, and Yuki IKEGAYA

列が連続してコントローラから出力されることで、図2のETF3-TEMの動作が実行可能である。

2-2 パーシャルスキャン設計

パーシャルスキャン設計とは、回路中の特定のレジスタのみをスキャンレジスタで設計するテスト容易化設計手法である。

文献[2]では、コントローラ中の状態レジスタとコントローラの出力である制御信号に付加する1ビットの観測用レジスタと状態信号に接続されているデータパス中の状態信号レジスタのみをスキャンレジスタとして設計する。コントローラ中の状態レジスタをスキャン設計する理由としては、コントローラのテスト容易性を向上させるためである。さらに、スキャンテスト時のシフト動作でコントローラ中の任意の状態に遷移が可能となり、データパスを含む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。制御信号にEXOR ツリーと1ビットの観測用レジスタを付加する理由としては、コントローラ内の故障を観測することが困難になるためである。状態信号レジスタをスキャン設計する理由としては、状態信号レジスタに接続されているハードウェア要素とコントローラのテスト容易性の向上のためである。

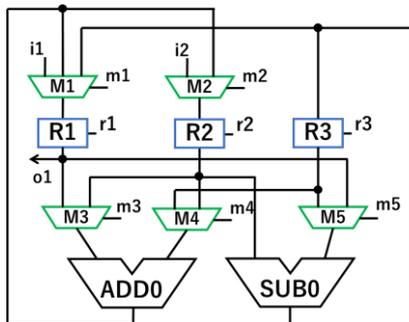


図 1. データパス例

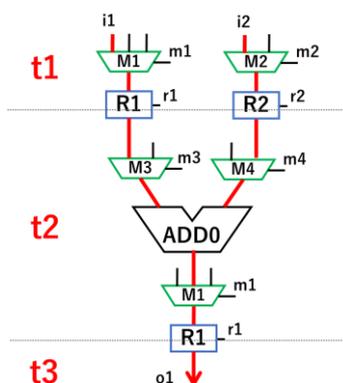


図 2. ETF3-TEM 例

表 1. テスト動作制御・状態信号系列の例

	r1	r2	r3	m1	m2	m3	m4	m5
t1	1	1	X	00	0	X	X	X
t2	1	X	X	01	X	0	1	X
t3	X	X	X	XX	X	X	X	X

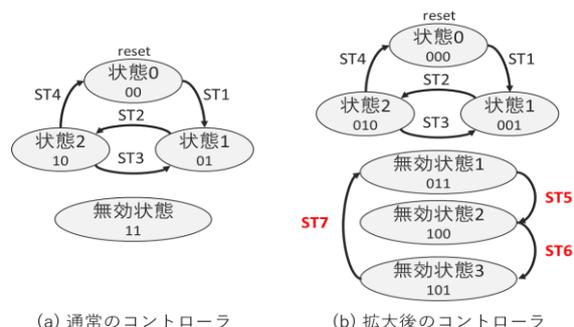
2-3 コントローラ拡大

コントローラ拡大[3]とは、コントローラに状態や状態遷移を追加する設計手法のことである。コントローラ中にはリセット状態から遷移し得ない状態が存在する場合があります、その状態を無効状態[3]という。文献[2]では、コントローラの状態レジスタをスキャン設計しているため、テスト時において、無効状態に遷移可能となる。したがって、文献[2]では、ETF k -TEMの動作を実行するためのテスト動作制御・状態信号系列を出力する状態遷移を無効状態にのみ設計する。この無効状態のことをテスト活性化状態と呼ぶ。

データパスのテスト容易な構造に着目して生成されたETF k -TEMの動作を実行する機能は、コントローラには備わっていない可能性がある。ETF k -TEMを考慮したテスト動作制御・状態信号系列を新たな状態遷移として無効テスト状態の状態遷移として設計することにより、ETF k -TEMの動作を実行可能となる。なお、ETF k -TEMを考慮したコントローラ拡大時に無効状態数が不足する場合は、コントローラの状態レジスタのビット幅を増加させて、コントローラは無効状態数を増加させる。図4にコントローラ拡大の例を示す。この例では、図2のETF3-TEMの動作を実行するためのテスト動作制御・状態信号系列(表1)を、無効テスト状態の状態遷移として図4の(a)の拡大前のコントローラに適用する。図2のETF3-TEMの動作を実行するためのテスト動作制御・状態信号系列(表1)は3サイクル必要であるため、拡大する状態遷移も3つ必要である。しかしながら、無効状態数が1つしか存在しないため、表1のテスト動作制御・状態信号系列を無効状態で設計することができない。よって、状態レジスタのビット幅を2から3に増加させることにより、さらに4つの無効状態が増加する。図4の(b)は、表1のt1~t3をST5~ST7の状態遷移としてそれぞれ無効状態のテスト活性化状態として設計した後のコントローラである。

2-4 k サイクルキャプチャテスト

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 であるテスト手法である。図3に、 k サイクルキャプチャテスト($k=3$)のスキャンイネーブルとクロックの波形を示す。スキャンイネーブルが0のときにキャプチャモードとなり、 k サイクル間($k=3$)順序動作を行う。



(a) 通常のコントローラ

(b) 拡大後のコントローラ

図 3 コントローラ拡大の例

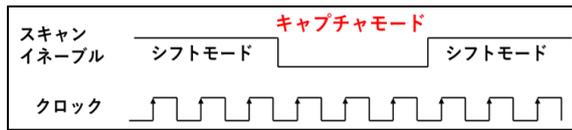


図 4. k サイクルキャプチャテスト例 ($k=3$)

2-5 DFT 手順

文献[2]の DFT は、RTL のコントローラとデータパスに適用される。まず、データパスの状態信号レジスタと、コントローラの状態レジスタをスキャン設計する。次に、コントローラの出力である制御信号線に観測ポイントである EXOR ツリーを挿入する。その EXOR ツリーの 1 ビット出力に 1 ビットのレジスタを付加しスキャン設計する。次に、データパスのすべてのハードウェア要素（演算器、マルチプレクサ、レジスタ、制御信号線）をテスト可能とするような ETFk-TEM を生成する。1 つのテスト対象ハードウェア要素に対して生成した ETFk-TEM の動作を実行するためのテスト動作制御・状態信号系列を生成し、コントローラの無効状態中にテスト活性化状態を設計する。その後は、生成した ETFk-TEM に含まれるテスト可能なハードウェア要素以外のテスト対象ハードウェア要素に対して ETFk-TEM を生成する。すべてのテスト対象ハードウェア要素がテスト可能となるまで以上の操作を繰り返す。生成された複数の ETFk-TEM からテスト動作制御・状態信号系列を生成し、すべての ETFk-TEM の動作を実行するために、コントローラの無効状態中にテスト活性化状態を設計する。拡大したコントローラを含んだ RTL を論理合成し、 k サイクルキャプチャテスト生成を実行する。

3. 提案手法

3-1 スロー・ファースト・スローテスト方式

非スキャン設計やパーシャルスキャン設計において、遷移故障を検出するためのテスト方式として SFS 方式が提案されている[4]。図 5 に、SFS テスト方式の概要図を示し、説明する。SFS テスト方式は、状態正当化、故障励起、故障伝搬の 3 つのフェーズに分かれている。状態正当化フェーズでは、故障を励起するために複数クロック間順序動作させる。非スキャン設計やパーシャルスキャン設計では、キャプチャモードの初期時刻におけるすべての非スキャン FF の値が初期化されていないため、故障励起するために複数クロック間順序動作させる必要がある。状態正当化フェーズではクロック周波数の低いクロックを用いる。したがって、状態正当化フェーズでは遷移故障を励起することは不可能である。故障励起フェーズでは、状態正当化フェーズによって伝搬された値により故障励起する。故障励起フェーズでは、遷移故障を励起する必

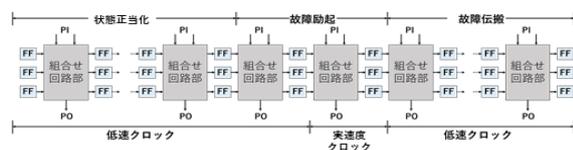


図 5. SFS テスト方式の概要

要があるため、クロック周波数が実速度のクロック周波数を用いる。故障伝搬フェーズでは、故障励起フェーズでキャプチャされた故障影響を外部出力もしくはスキャン FF で観測するために複数クロック間順序動作させる。故障伝搬フェーズでは、故障励起フェーズでキャプチャされた故障影響を外部出力でも観測可能にするために、クロック周波数の低いクロックを用いる。

3-2 遷移故障モデルのためのテスト容易化機能的 k 時間展開モデル生成法

遷移故障モデルのためのテスト容易化機能的 k 時間展開モデルは、3-1 節で説明した SFS テスト方式の考え方に基いて生成する。SFS テスト方式は、状態正当化、故障励起、故障伝搬の 3 つのフェーズで成り立っている。それぞれのフェーズを ETFk-TEM で実現することで、遷移故障を検出することが可能であると考える。図 6 に図 1 の ADD0 をテスト可能にする ETF4-TEM の例を示す。図 6 中の t_1, t_2, t_3, t_4 は時刻を示している。

まずは、状態正当化フェーズについて考える。ADD0 の故障を励起させるためには ADD0 の入力まで何らかの値を伝搬させなければならない。したがって、状態正当化フェーズでは ADD0 の入力へ値を伝搬することが可能なレジスタまでの経路を考える。さらに、その経路の中から k が最小になるような経路を選択する。ADD0 へ値を伝搬することが可能なレジスタは $R1, R2, R3$ であるが、 $R3$ を通る経路に関しては、外部入力から $R3$ に値を伝搬するまでに 2 時刻必要なため $R1, R2$ を通る経路を選択する。図 6 中の時刻 t_1 に状態正当化のための経路を示す。 $R1$ までの経路は $i_1 \rightarrow M1$ の入力 $0 \rightarrow R1$ の入力であり、 $R2$ までの経路は、 $i_2 \rightarrow M2$ の入力 $0 \rightarrow R2$ の入力である。

次に、故障励起フェーズについて考える。ADD0 の故障を励起させるためには、ADD0 の入力まで何らかの値を伝搬させ、次の時刻においても ADD0 の入力まで何らかの値を伝搬させなければならない。したがって、故障励起フェーズでは 2 時刻連続で ADD0 の入力へ値を伝搬する経路を考える。さらに、その経路の中から k が最小になるような経路を選択する。図 6 中の時刻 t_2 と t_3 に故障励起のための経路を示す。まず t_2 について説明する。状態正当化フェーズで $R1, R2$ が初期化済みであるため、 $R1, R2$ から ADD0 の入力への経路を考える。ADD0 の入力 0 への経路は $R1 \rightarrow M3$ の入力 $0 \rightarrow ADD0$ の入力 0 であり、ADD0 の入力 1 まで

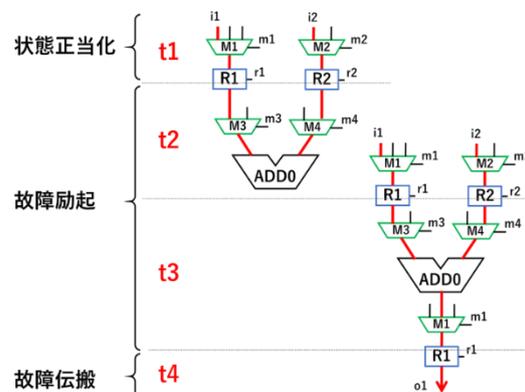


図 6. 遷移故障モデルのための ETF4-TEM

の経路は, $R2 \rightarrow M4$ の入力 $1 \rightarrow ADD0$ の入力 1 である。故障励起するためには次の時刻 $t3$ においても $ADD0$ の入力へ何らかの値を伝搬させなければならない。この例では時刻 $t1$ と同一の経路を選択する。さらに、励起した故障をレジスタでキャプチャするために $ADD0$ の出力からレジスタまたはスキャンレジスタまでの経路を考える。この例では RI までの経路を考える。 $ADD0$ の出力から RI までの経路は, $ADD0$ の出力 $\rightarrow MI$ の入力 $1 \rightarrow RI$ の入力である。

最後に故障伝搬フェーズについて考える。 $ADD0$ の故障をキャプチャした RI から外部出力またはスキャンレジスタの入力までの経路を考える。さらにその経路の中から k が最小になる経路を選択する。図 6 中の $l4$ に故障伝搬のための経路を示す。この例では, RI の出力が外部出力であるため、故障伝搬経路は $RI \rightarrow o1$ である。

4. 実験結果

本 DFT 手法の有効性を示すために、10 種類の RTL ベンチマーク回路を用いた実験結果を示す。本実験の故障モデルは 遷移故障であり、データパス及びコントローラ内の全故障を評価対象とする。本実験では、対象の 10 種類の回路に対して、本 DFT 手法を適用し、比較する回路として、DFT を適用しないオリジナル回路とフルスキャン設計を適用した回路を用いて実験を行った。本 DFT 手法におけるパーシャルスキ設計とフルスキャン設計のスキャンパス数は 1 本とした。論理合成には Synopsys 社の DesignCompiler を使用し、テスト生成には Synopsys 社の TetraMAX を使用した。

表 2 に、10 種類の RTL ベンチマーク回路の回路情報を示し、表 3 にテスト生成結果を示す。

面積オーバーヘッドについては、フルスキャン設計の面積オーバーヘッドから、平均 3.32%、最大 13.12% の削減に成功した。しかしながら、Fig17 のみ削減に失敗した。これは、拡大したテスト活性化状態数が多く、コントローラ的面積が大きくなってしまったためと考えられる。

テスト実行時間については、フルスキャン設計のテ

スト実行時間から、平均 91.45%、最大 95.50% の削減に成功した。

故障検出効率については、フルスキャン設計の故障検出効率から、平均 3.61%、最大 18.79% 低下した。この結果から、故障検出効率を向上させるためのテスト容易化機能的 k 時間展開モデル生成法を考える必要がある。

5. おわりに

本論文では、遷移故障検出のためのテスト容易化機能的 k 時間展開モデル生成法を提案した。10 種類の RTL ベンチマーク回路を用いた実験では、提案する DFT 手法を適用することで、面積オーバーヘッドをフルスキャン設計と比べて 1 回路を除いた 9 回路で削減し、テスト実行時間はすべての回路において削減に成功した。しかしながら、一部の回路においては故障検出効率に課題が残る。今後の課題としては、SFS テスト方式に対応したテスト生成ツールの作成と、故障検出効率を向上させるためのテスト容易化機能的 k 時間展開モデル考案などが挙げられる。

参考文献

- [1] 藤原 秀雄, デジタルシステムの設計とテスト, 工学図書株式会社, 2004.
- [2] Yuta Ishiyama, Toshinori Hosokawa, Hiroshi Yamazaki, "A Design for Testability Method for k-Cycle Capture Test Generation", IEEE 25th International Symposium on On-Line Testing and Robust System Design(IOLTS'19), pp.40-43, Jul 2019.
- [3] M.L.FLottes, B.Rouzeyre, L.Volpe, "A Controller Resynthesis Based Method for Improving Datapath Testability", IEEE International Symposium on Circuits and Systems(ISCAS), pp. 347-350, May 2000.
- [4] A. Krstic, Kwang-Ting Cheng, S.T. Chakradhar "Testing High Speed VLSI Devices Using Slower Testers", Proceedings 17th IEEE VLSI Test Symposium, Apr 1999.

表 2. 回路情報

回路名	ビット幅	DFT手法	SFF/FF	テスト活性化状態数	面積	面積オーバーヘッド[%]
ARF	32bit	オリジナル	0/324	0	30007	0.00
		フルスキャン	324/324	0	31422	4.72
		本手法	9/329	132	31274	4.22
BPF	32bit	オリジナル	0/260	0	17888	0.00
		フルスキャン	260/260	0	19119	6.88
		本手法	9/265	124	18928	5.81
DFCT	32bit	オリジナル	0/259	0	20516	0.00
		フルスキャン	259/259	0	21552	5.05
		本手法	8/264	69	21125	2.97
ex2	32bit	オリジナル	0/163	0	13867	0.00
		フルスキャン	163/163	0	14519	4.70
		本手法	7/167	39	14121	1.83
ex4	32bit	オリジナル	0/195	0	8878	0.00
		フルスキャン	195/195	0	9658	8.79
		本手法	6/198	12	9013	1.52
FFT	32bit	オリジナル	0/323	0	21943	0.00
		フルスキャン	323/323	0	23187	5.67
		本手法	8/328	92	23093	5.24
Fig17	32bit	オリジナル	0/201	0	23385	0.00
		フルスキャン	201/201	0	33963	45.23
		本手法	11/203	315	34710	48.43
Kim	32bit	オリジナル	0/198	0	4736	0.00
		フルスキャン	198/198	0	5528	16.72
		本手法	9/201	43	5195	9.69
Maha	32bit	オリジナル	0/198	0	4155	0.00
		フルスキャン	198/198	0	5033	21.13
		本手法	9/201	80	4947	19.06
Sehwa	32bit	オリジナル	0/262	0	4847	0.00
		フルスキャン	262/262	0	5910	21.93
		本手法	9/265	76	5274	8.81

表 3. テスト生成結果

回路名	DFT手法	キャプチャサイクル数	総故障数	故障検出効率 [%]	テスト系列数	テスト実行時間 [サイクル]	
ARF	フルスキャン		2	107540	86.34	667	219101
	本手法		5	113692	86.24	721	9867
BPF	フルスキャン		2	64726	96.35	530	139652
	本手法		5	69834	96.87	612	8384
DFCT	フルスキャン		2	75376	99.40	232	61045
	本手法		5	77522	95.75	444	5749
ex2	フルスキャン		2	47680	98.78	242	40337
	本手法		6	48292	98.29	298	3783
ex4	フルスキャン		2	29356	99.99	209	41579
	本手法		5	29368	99.21	245	2846
FFT	フルスキャン		2	81328	95.47	253	82803
	本手法		5	86976	98.47	526	6789
Fig17	フルスキャン		2	123766	93.03	473	97167
	本手法		6	130612	83.63	838	12737
Kim	フルスキャン		2	15956	99.56	209	42209
	本手法		5	17614	96.79	320	4606
Maha	フルスキャン		2	15182	99.60	244	49244
	本手法		5	18778	80.89	383	5028
Sehwa	フルスキャン		2	16884	99.99	216	57719
	本手法		5	17152	96.93	298	4019