

## パーシャルスキャン設計とコントローラ拡大を用いた テスト容易化機能的 $k$ 時間展開モデルに基づくテスト容易化設計法

日大生産工(院) ○石山 悠太 日大生産工 細川 利典  
日大生産工 山崎 紘史

### 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. 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, 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ので, 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が増大するという問題点がある.

また, フルスキャン設計の問題点を解決するために, 非スキャンテストに基づいたレジスタ転送レベル(Register Transfer Level : RTL)における DFT 手法が提案されている[2]. これらの DFT 手法はコントローラとデータパスから構成される回路を対象にし, そのデータパスのみに着目した手法である. そのため, データパスに対する高い故障検出効率を実現するためには, データパスとコントローラがテスト時に分離されていることが前提となる. テスト時に RTL データパスとコントローラを分離するためには付加回路が必要である. 文献[4]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  $k$  時間展開モデルを用いたテスト生成手法が提案されている. 文献[4]の手法では, データパスのテスト容易な構造に基づいてテスト生成を行うための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現可能にするためにコントローラを拡大する. テスト生成時は, 拡大したコントローラの機能に着目し, 生成したテスト容易化機能的  $k$  時間展開モデルの動作を実現するような制御信号・状態信号系列を制約として与える. テスト容易化機能的  $k$  時間展開モデルを用いてテスト生成することにより, 演算器の故障のテスト生成に関しては高速に高い故障検出効率を達成することができている. しかしながら, 文献[4]の手法は, データパス内の演算器のテスト生成のみを対象とした DFT 手法であり, さらにテスト容易化機能的  $k$  時間展開モデルを使用するために, 時系列値を制約値として与えることのできる専用の制約付テスト生成を必要とする.

本論文では, 演算器だけではなく回路全体のテスト容易性を考慮し, 回路構造に基づいた一般のテスト生成を用いて高い故障検出効率を達成するために, 回路中のす

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, コントローラ)に対応するテスト容易化  $k$  時間展開モデルの生成とその動作を制御可能にするための無効テスト状態[2]の状態遷移を設計するというコントローラ拡大に基づく DFT 手法を提案する. また, 本 DFT 手法は, コントローラ中の状態レジスタとデータパスからコントローラへの状態信号に接続している状態信号レジスタと制御信号線に付加する EXOR ツリーの出力レジスタをスキャン設計したパーシャルスキャン設計を前提とする. コントローラをスキャン設計することにより, 無効テスト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で, 任意の状態から  $k$  サイクル間状態遷移を実行することにより, 各ハードウェア要素に対するテスト容易化機能的  $k$  時間展開モデルが回路構造に基づくテスト生成において実現可能であると考えられる. パーシャルスキャン設計とコントローラ拡大により, フルスキャン設計よりも少ないハードウェアオーバーヘッドで同等の故障検出効率を達成することを目指す.

### 2. テスト容易化設計手法

#### 2-1 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. パーシャルスキャン設計の利点として, フルスキャン設計と比較し, ハードウェアオーバーヘッドとテスト実行時間を削減できることが挙げられる. しかしながら, 順序回路のテスト生成を必要とするため, フルスキャン設計と比較して, 故障検出効率の低下が問題となる.

本手法では, コントローラ中の状態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と制御信号線に付加する EXOR ツリーの出力レジスタのみをスキャン設計する. 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く, 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. これは, スキャンテスト時のシフト動作で任意の状態に遷移が可能となるからである. 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. 状態信号レジスタはコントローラに接続されているため, スキャン設計をし

## A Controller Augmentation Method for $k$ -cycle Capture Test Generation Using Partial Scan Designs

Yuta ISHIYAMA, Toshinori HOSOKAWA, and Hiroshi YAMAZAKI

ないと状態信号レジスタに接続されているハードウェア要素の可観測性が低く、コントローラの可制御性も低くなる。状態レジスタと状態信号レジスタをスキャンすることによりコントローラの可制御性は向上するが、コントローラの出力である制御信号線はデータパスに接続されているためコントローラの可観測性は低い可能性がある。可観測性を向上させるために、制御信号線にEXOR ツリーを挿入し、EXOR ツリーの出力に1ビットの観測用レジスタを付加する。この観測用レジスタをスキャン設計することによりコントローラの可観測性を向上させる。

## 2-2 テスト容易化機能的 $k$ 時間展開モデル

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)[4]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。ETF- $k$ TEM は、全てのハードウェア要素(演算器、マルチプレクサ、レジスタ、コントローラ)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におけるETF-3TEMの例を示す。図1において、 $i1\sim i8$ は外部入力、 $o1$ 、 $o2$ は外部出力、 $R1$ 、 $R2$ 、 $R3$ 、 $R5$ 、 $R6$ 、 $R7$ 、 $R8$ はホールド機能付レジスタ、 $R0$ 、 $R4$ はホールド機能

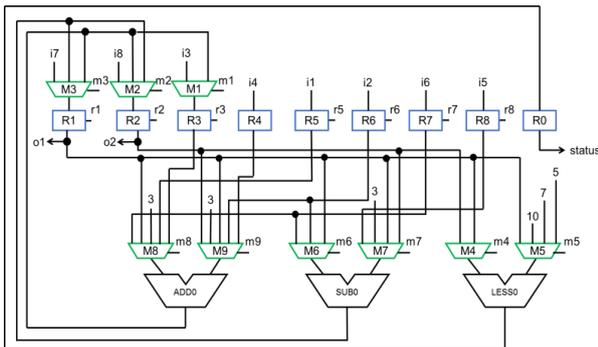


図1. データパス例

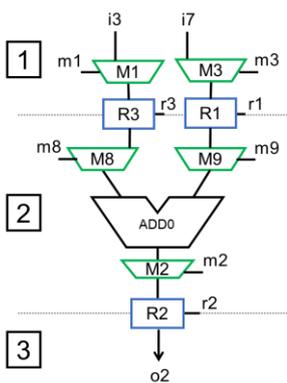


図2. 表1で動作可能なETF-3TEM例

なしレジスタ、 $ADD0$ は加算器、 $SUB0$ は減算器、 $LESS0$

表1. テスト動作制御・状態信号系列の例

	r1	r2	r3	r5	r6	r7	r8	m1	m2	m3	m4	m5	m6	m7	m8	m9	s
t1	1	X	1	X	X	X	X	0	XX	00	X	XX	XX	XX	XXX	XXX	X
t2	X	1	X	X	X	X	X	X	01	XX	X	XX	XX	XX	100	010	X
t3	X	X	X	X	X	X	X	X	XX	XX	X	XX	XX	XX	XXX	XXX	X

は比較器、 $M1\sim M9$ はマルチプレクサ、 $m1\sim m9$ 、 $r1\sim r7$ は制御信号、 $s$ は状態信号である。表1に図2のETF-3TEMを実現するための各時刻の制御信号値(0,1,X)と状態信号値(0,1,X)の時系列であるテスト動作制御・状態信号系列[4]を示す。表1において、 $t1\sim t3$ は時刻を表し、図2における四角で囲まれた時刻1~3にそれぞれ対応している。図2のハードウェア要素の入出力において、外部入力または時刻1のスキャンレジスタからハードウェア要素の入力に何らかの値を伝搬でき、ハードウェア要素の出力から何らかの値を外部出力または時刻3のスキャンレジスタに伝搬できるものは、 $ADD0$ 、 $R1$ 、 $R2$ 、 $R3$ 、 $M1$ の入力0、 $M2$ の入力1、 $M3$ の入力0、 $M8$ の入力4、 $M9$ の入力2である。したがって、表1のテスト動作制御・状態信号系列[4]で動作する演算器は $ADD0$ で、動作するレジスタは $R1$ 、 $R2$ 、 $R3$ で、動作するマルチプレクサの入力は $M1$ の入力0、 $M2$ の入力1、 $M3$ の入力0、 $M8$ の入力4、 $M9$ の入力2である。また、これらのハードウェア要素は図2のETF-3TEMで動作可能であると呼ぶ。

## 2-3 $k$ 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

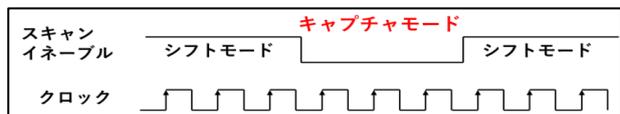


図3.  $k$  サイクルキャプチャテスト例 ( $k=3$ )

## 2-4 コントローラ拡大

コントローラ拡大[2]とは、コントローラに状態や状態遷移を追加するテスト容易化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[2]という。本手法では、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、本手法では、ETF- $k$ TEMの動作を実現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無効テスト状態[2]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の動作を実現する機能は、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の動作を実現可能となる。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、コントローラの無効状態数を増やす。図4にコントローラ拡大の例を示す。この例では、図2のETF-3TEMの動作を実現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

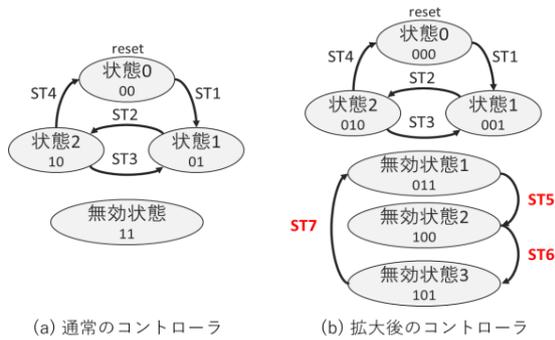


図 4. コントローラ拡大の例

前のコントローラに適用する．図 2 の ETF-3TEM の動作を実現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  $t1 \sim t3$  を  $st5 \sim st7$  の状態遷移としてそれぞれ設計した後のコントローラである．

### 3. テスト容易化機能的 $k$ 時間展開モデルにおけるハードウェア要素のテスト可能性

#### 3-1 コントローラのテスト可能性

本論文では，コントローラの論理回路は図 5 のようにモデル化される．図 5 において，状態レジスタは有限状態機械の状態を識別するためのレジスタである．次状態遷移論理は，次の状態遷移先の状態を決定するための組合せ回路で，その出力値（次状態）は，状態レジスタの値（現在状態）と状態信号の値によって決定される．出力論理は，制御信号線の値を決定する組合せ回路で，その出力値（制御信号線の値）は，状態レジスタの値（現在状態）と状態信号の値によって決定される．2-1 節で述べたように，コントローラ中の状態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はスキャン設計が適用され，可制御・可観測なスキャンレジスタが用いられている．次状態遷移論理のテストはスキャンレジスタから制御可能で，故障の影響はスキャンレジスタで観測可能であり，1 時間展開モデルを用いてテスト生成が可能である．一方，出力論理のテストはスキャンレジスタから制御可能であるが，故障の影響は制御信号線に伝搬され，データパス中の外部出力またはスキャンレジスタで観測しなければならない．したがっ

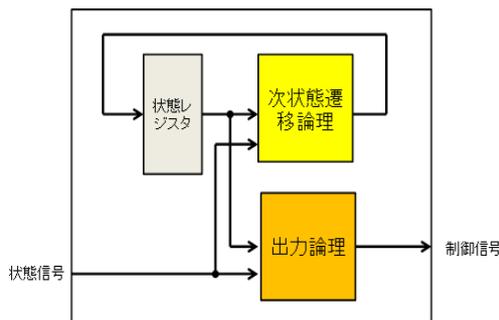


図 5. コントローラの論理回路モデル

て，出力論理の故障を確実に検出するために制御信号線に観測ポイントを挿入する．本手法では制御信号線に EXOR ツリーを挿入し，EXOR ツリーの出力に 1 ビットの観測用レジスタを付加する．この観測用レジスタをスキャンすることで出力論理の可観測性を向上させる．

#### 3-2 演算器のテスト可能性

<定義 1: テスト容易化機能的  $k$  時間展開モデルでテスト可能な演算器>

演算器  $f$  が ETF- $k$ TEM  $j$  で動作可能であるとき，演算器  $f$  は  $j$  でテスト可能であると呼ぶ．

例 1. 図 2 の ETF-3TEM において，演算器  $ADD0$  は動作可能であるので， $ADD0$  は図 2 の ETF-3TEM でテスト可能である．

#### 3-3 マルチプレクサのテスト可能性

<定義 2: テスト容易化機能的  $k$  時間展開モデルでテスト可能なマルチプレクサの入力  $i$ >

マルチプレクサ  $m$  の入力  $i$  が ETF- $k$ TEM  $j$  で動作可能であるとき，マルチプレクサ  $m$  の入力  $i$  は  $j$  でテスト可能であると呼ぶ．

例 2. 図 2 の ETF-3TEM において，マルチプレクサ  $M1$  の入力  $0$  は動作可能であるので， $M1$  の入力  $0$  は図 2 の ETF-3TEM でテスト可能である．

<定義 4: テスト容易化機能的時間展開モデル集合でテスト可能なマルチプレクサ>

ETF- $k$ TEM の集合  $T$  が与えられたとき，マルチプレクサ  $m$  の各入力  $i$  が  $T$  中の少なくとも 1 個の ETF- $k$ TEM でテスト可能であるとき，マルチプレクサ  $m$  は  $T$  でテスト可能であると呼ぶ．

#### 3-4 レジスタのテスト可能性

<定義 5: テスト容易化機能的  $k$  時間展開モデルでテスト可能なレジスタ>

レジスタ  $r$  が ETF- $k$ TEM  $j$  で動作可能であるとき，レジスタ  $r$  は  $j$  でテスト可能であると呼ぶ．

例 4. 図 2 の ETF3-TEM において，レジスタ  $R1, R2, R3$  は動作可能であるので， $R1, R2, R3$  は図 2 の ETF3-TEM でテスト可能である．

#### 3-5 回路全体のテスト可能性

<定義 7: テスト容易化機能的  $k$  時間展開モデル集合でテスト可能な回路>

ETF- $k$ TEM の集合  $T$  が与えられたとき，回路  $C$  中の各演算器，各レジスタ，各マルチプレクサが  $T$  中の少なくとも 1 個の ETF- $k$ TEM でテスト可能であるとき，回路  $C$  はテスト可能である．

## 4. DFT 手順

本 DFT 手法は，コントローラとデータパスに適用する．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，データパスの全てのハードウェア要素（演算器，マルチプレク

サ、レジスタ)をテスト可能とするようなETF-kTEMを生成する。生成するための方針は以下の通りである。

(方針1)  $k$ の値は可能な限り小さくする。

(方針2) テスト可能となる多入力ハードウェア要素の各入力に何らかの値を伝搬するための(時刻, 外部入力またはスキャンレジスタ)の集合は可能な限り独立であるようにする。

1つのテスト対象ハードウェア要素に対して生成したETF-kTEMの動作を実現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テスト状態に設計する。その後は、生成したETF-kTEM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ETF-kTEMを生成する。全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。

## 5. 実験結果

本論文では、本DFT手法の有効性を示すために、3種類の動作合成ベンチマーク回路[2]を用いた実験結果を示す。本実験の故障モデルは単一縮退故障であり、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、実験対象の3種類の回路に対して、本DFT手法を適用し、比較するために、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ャン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。テスト生成にはSynopsys社のTetraMAXを用い、テスト生成時のバックトラックリミットは10000とした。

表2に回路情報を示し、表3にテスト生成結果を示す。

表2において、比較対象であるフルスキャン設計の面積オーバーヘッドは約17%~20%に対して、本DFT手法の面積オーバーヘッドはすべての回路において下回っている。

表3において、本DFT手法の故障検出効率はフルスキャン設計の故障検出効率に対してほぼ同等であると

言える。また、テスト実行時間においては約82~88%の削減に成功した。

## 6. おわりに

本論文では、パーシャルスキャン設計を用いた $k$ サイクルキャプチャテストのためのコントローラ拡大法を提案した。3種類の動作合成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、フルスキャン設計と比べて故障検出効率を維持しつつ、面積オーバーヘッドの削減とテスト実行時間の削減をすることができた。今後の課題として、コントローラの無効テスト状態の状態遷移を圧縮することにより、無効テスト状態数を削減することにより面積オーバーヘッドを削減することが挙げられる。

## 謝辞

本研究に際し、貴重な意見を頂きました株式会社ソシオネクストの濱田周治様に深く感謝いたします。

本研究は一部、株式会社ソシオネクストとの共同研究による。

## 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] S. Ohtake, T. Masuzawa, and H. Fujiwara, "A non-scan approach to DFT for Controllers Achieving 100% Fault Efficiency," Journal of Electronic Testing: Theory and Applications (JETTA), Vol. 16, No. 5, pp.553-566, Oct. 2000.
- [3] 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.

表2. 実験結果 (回路情報)

回路名	ビット幅	テスト容易化設計手法	スキャンFF/総FF数	増加した状態レジスタビット幅	追加状態遷移数	回路面積	面積オーバーヘッド (%)
Schwa	32bit	オリジナル	0/262	0	0	4848	0.00
		フルスキャン	262/262	0	0	5910	21.91
		本手法	10/266	3	227	5637	14.21
Maha	32bit	オリジナル	0/198	0	0	4177	0.00
		フルスキャン	198/198	0	0	4983	19.30
		本手法	10/202	3	134	4845	15.99
Kim	32bit	オリジナル	0/198	0	0	4776	0.00
		フルスキャン	198/198	0	0	5581	16.86
		本手法	10/202	3	216	5537	15.93

表3. 実験結果 (テスト生成結果)

回路名	Schwa			Maha			Kim		
	オリジナル	フルスキャン	本手法	オリジナル	フルスキャン	本手法	オリジナル	フルスキャン	本手法
テスト容易化設計手法	オリジナル	フルスキャン	本手法	オリジナル	フルスキャン	本手法	オリジナル	フルスキャン	本手法
テストサイクル数 $k$	10	1	4	10	1	4	10	1	4
故障検出率(%)	73.47	99.84	98.87	69.98	99.59	99.31	72.50	99.05	98.86
故障検出効率(%)	73.57	100.00	99.01	70.25	99.99	99.70	73.14	99.99	99.70
総故障数	17142	18204	20188	15294	16046	18098	17558	18360	20898
検出故障数	12595	18175	19960	10682	15980	179473	12729	18185	20659
テスト不能故障数	22	29	29	59	65	71	155	173	177
テストパターン数	27	103	246	39	128	252	37	112	270
テスト生成時間(sec)	79.18	0.17	64.08	61.23	0.29	117.31	94.56	0.34	99.65
テスト実行時間(cycle)	815	27457	3299	962	25801	3371	898	22601	3595