

C-018 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

石山悠太^{†1} 細川利典^{†2} 山崎紘史^{†2}
 YUTA ISHIYAMA^{†1} TOSHINORI HOSOKAWA^{†2} HIROSHI YAMAZAKI^{†2}

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][3]。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。一方、データパスとコントローラを分離しないことを前提とした RTL 回路に対する DFT 手法も提案されている[4]。文献[4]ではデータパスのテスト容易な構造に基づいて、テスト時にその動作を制御させるようにコントローラを拡大する。しかしながら、これまで提案されてきた順序回路のテスト生成アルゴリズム[5][6]は、回路構造のみからテスト系列を生成する。それゆえ、テスト容易な構造に基づいて拡大したコントローラの機能通りにテスト生成するとは限らない。文献[7]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[7]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[7]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[3]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。

^{†1} 日本大学大学院生産工学研究科

^{†2} 日本大学生産工学部

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)[7]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、 $R0, R4$ はホールド機能なしレジスタ、 $ADD0$ は加算器、 $SUB0$ は減算器、 $LESS0$ 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[7]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のスクラ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のスクラ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[7]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。

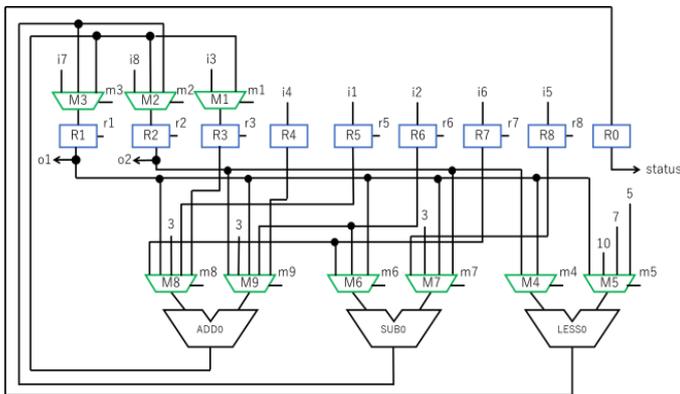


図1. データパス例

表1. テスト動作制御・状態信号系列の例1

	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

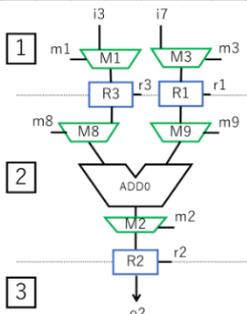


図2. 表1で動作可能なETF-3TEM例

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($k=3$)のスキャンインネーブルとクロックの波形を示す。スキャンインネーブルが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

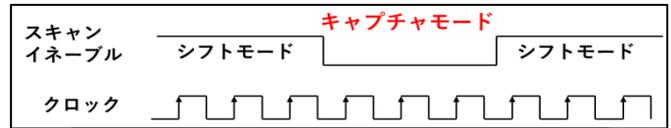
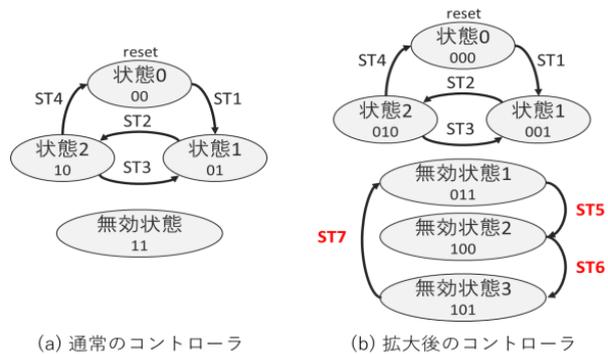


図3. k サイクルキャプチャテスト例 ($k=3$)

2-4 コントローラ拡大

コントローラ拡大[3]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[3]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[3]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、コントローラの状態レジスタを増やす。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。



(a) 通常のコントローラ (b) 拡大後のコントローラ

図4. コントローラ拡大の例

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、コントローラはミラー型の有限状態機械で設計されていると仮定する。したがって、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。したがって、コントローラの出力論理中の故障を制御信号線まで伝搬するようにスキャンレジスタの値を制御し、制御信号線の故障を検出する ETF- k TEM を組み合わせたテストを考えなければならない。したがって、コントローラのテストを可能にするためには、少なくとも制御信号線の故障を検出する ETF- k TEM を生成するテスト動作制御・状態信号系列を無効テスト状態の状態遷移として設計する必要がある。しかしながら、制御信号線の故障を検出するための ETF- k TEM を用いたとしても、出力論理の故障は検出できない可能性がある。したがって、出力論理の故障を確実に検出するためには制御信号線に観測ポイントを挿入する必要があるが、面積オーバーヘッドが大きいことが懸念される。

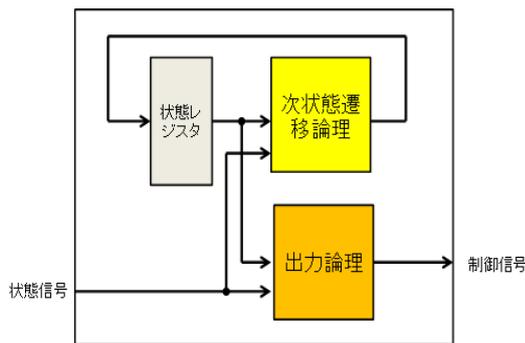


図 5. コントローラの論理回路モデル

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 3: 制御信号線の縮退故障テストのためのテスト容易化機能的 k 時間展開モデル>

長さ k のテスト動作制御・状態信号系列が与えられたとき、データパスを k 時間展開した回路モデルに各時刻の制御信号値と状態信号値を制約として ETF- k TEM j を生成する。ただし、故障が仮定される制御信号線値は正常値 $\alpha (\in \{0, 1\})$ に対応するマルチプレクサ m の入力 i のみではなく、故障値 $\bar{\alpha}$ に対応するマルチプレクサ m の入力 h も時間展開する。生成された ETF- k TEM j 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マルチプレクサ m の入力 i 及び h までに何らかの値を伝搬する経路が存在し、かつ m の出力から何らかの値を外部出力または時刻 k のスキャンレジスタの入力まで伝搬する経路が存在するとき、 j でマルチプレクサ m の制御信号の $\bar{\alpha}$ 縮退故障をテスト可能であると呼ぶ。

表 2 に示すテスト動作制御・状態信号系列で動作可能な ETF-4TEM を図 6 に示す。

例 3. 図 6 の ETF-4TEM において、時刻 $t2$ の外部入力 $i3$ から時刻 $t2$ のマルチプレクサ M1 の入力 0 に何らかの値を伝搬可能な経路が存在し、かつ時刻 $t1$ の外部入力 $i1$ と $i4$ から時刻 $t2$ の M1 の入力 1 に何らかの値を伝搬可能な経路が存在し、かつ、時刻 $t2$ の M1 の出力の何らかの値を時刻 $t4$ の外部出力 $o1$ に伝搬可能な経路が存在するので、この ETF-4TEM で M1 の制御信号線の 1 縮退故障はテスト可能である。

表 2. テスト動作制御・状態信号系列の例 2

	r1	r2	r3	r5	r6	r7	r8	m1	m2	m3	m4	m5	m6	m7	m8	m9	s
t1	X	X	X	1	X	X	X	X	XX	XX	X	XX	XX	XX	XXX	XXX	X
t2	X	X	1	X	X	X	X	0	XX	XX	X	XX	XX	XX	011	100	X
t3	1	X	X	X	X	X	X	X	XX	10	X	XX	XX	XX	100	100	X
t4	X	X	X	X	X	X	X	X	XX	XX	X	XX	XX	XX	XXX	XXX	X

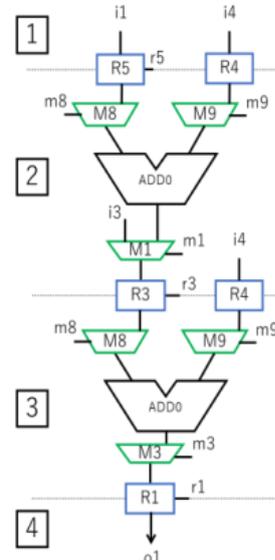


図 6. 表 2 で動作可能な ETF-4TEM 例

<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、かつマルチプレクサ m の各制御信号線の各縮退故障が 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

(1) ホールド機能なしレジスタ

<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>

制御信号を持たないレジスタ 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 7 は表 3 で動作可能な ETF-3TEM である。図 7 において、制御信号を持たないレジスタ $R4$ は動作可能であるので、 $R4$ は図 7 の ETF-3TEM でテスト可能である。

表 3. テスト動作制御・状態信号系列の例 3

	r1	r2	r3	r5	r6	r7	r8	m1	m2	m3	m4	m5	m6	m7	m8	m9	s
t1	X	X	X	1	X	X	X	XX	XX	X	XX	XX	XX	XX	XXX	XXX	X
t2	1	X	X	X	X	X	X	XX	10	X	XX	XX	XX	011	100	X	
t3	X	X	X	X	X	X	X	XX	XX	X	XX	XX	XX	XXX	XXX	X	

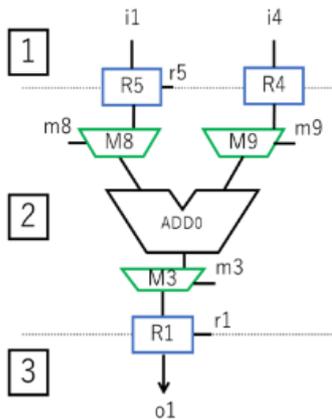


図 7. 表 3 で動作可能な ETF-3TEM 例

(2) ホールド機能付レジスタ

ホールド機能付レジスタをテスト可能にするためには、そのレジスタが動作する ETF- k TEM を生成することに加えて、そのレジスタの制御信号線の 1 縮退故障をテストするための ETF- k TEM を生成する必要がある。レジスタの制御信号線の 0 縮退故障は故障値がそのレジスタに伝搬できないので、テスト不能故障と考える。ホールド機能付レジスタは図 8 に示すようにマルチプレクサとホールド機能なしレジスタでモデル化できる。

例 5. 図 9 の ETF-4TEM において、時刻 $t2$ の外部入力 $i8$ から時刻 $t2$ の $R2$ (マルチプレクサ) の入力 0 に何らかの値を伝搬可能な経路が存在し、かつ時刻 $t1$ の外部入力 $i8$ から時刻 $t2$ の $R2$ (マルチプレクサ) の入力 1 に何らかの値を伝搬可能な経路が存在し、かつ時刻 $t2$ の $R2$ (マルチプレクサ) の出力の何らかの値を時刻 $t4$ の外部出力 $o1$ に伝搬可能な経路が存在するので、この ETF-4TEM で $R2$ の制御信号線の 1 縮退故障がテスト可能である。

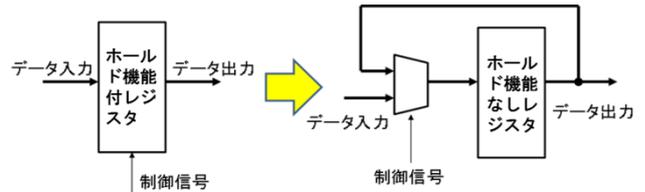


図 8. 制御信号付レジスタのモデル化

<定義 6 : 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制御信号付レジスタ>

ETF- k TEM の集合 T が与えられたとき、レジスタ r が T 中の少なくとも 1 個の ETF- k TEM で動作可能で、かつレジスタ r の制御信号線の 1 縮退故障が 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、レジスタ r は T でテスト可能であると呼ぶ。

表 4. テスト動作制御・状態信号系列の例 4

	r1	r2	r3	r5	r6	r7	r8	m1	m2	m3	m4	m5	m6	m7	m8	m9	s
t1	X	1	X	1	X	X	X	00	XX	X	XX	XX	XX	XXX	XXX	X	
t2	X	0	X	X	X	X	1	00	XX	X	XX	XX	XX	XXX	XXX	X	
t3	1	X	X	X	X	X	X	XX	01	X	XX	00	11	XXX	XXX	X	
t4	X	X	X	X	X	X	X	XX	XX	X	XX	XX	XX	XXX	XXX	X	

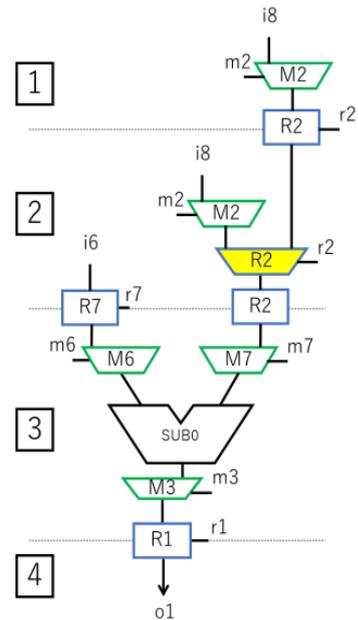


図 9. 表 4 で動作可能な ETF-4TEM 例

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 中の各演算器や各制御信号なしレジスタが T 中の少なくとも 1 個の ETF- k TEM でテスト可能であり、かつ各制御信号付レジスタやマルチプレクサが T 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、制御信号線）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[3]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。動作合成には内製の動作合成システム PICTHY を用い、論理合成には Synopsys 社の DesignCompiler を用いた。また、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 5 に回路情報を示し、表 6 にテスト生成結果を示す。

表 5 において 1 列目に実験対象回路名を示す。2 列目に回路のビット幅を示す。3 列目に回路に適用した DFT 手法を示す。本実験で用いる DFT 手法は 4 つあり、1, 5, 9 行目は DFT 手法を適用しないオリジナルを示す。2, 6, 10 行目はフルスキャン設計を示す。3, 7, 11 行目は本 DFT 手法における制御信号線を考慮せずに ETF-kTEM を生成してコントローラ拡大した手法を示す。4, 8, 12 行目は本 DFT 手法における制御信号線を考慮した ETF-kTEM を生成してコントローラ拡大した手法を示す。4 列目に総 FF 数とスキャン FF 数を示す。5 列目に無効状態を用いたコントローラ拡大によって状態レジスタのビット幅の増加による追加 FF 数を示す。6 列目に無効状態を用いたコントローラ拡大によって追加された状態遷移数を示す。7 列目に回路の総面積を示す。8 列目にオリジナル回路と比較した面積オーバーヘッド(%)を示す。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。特に制御信号線を考慮していない DFT 手法では全回路において平均約 6% であり、大きな面積オーバーヘッドの削減に成功している。

表 6 において、1 行目に実験対象回路を示す。2 行目に回路に適用した DFT 手法を示す。表 6 に示している 4 つの DFT 手法は表 5 に示した DFT 手法に対応している。それぞれの DFT 手法に対して 3 行目から 11 行目にテスト生成結果を示す。

3 行目に k サイクルキャプチャテストで用いたテストサイクル数 k を示す。4 行目に故障検出率(%)を示す。5 行目に故障検出効率(%)を示す。6 行目に総故障数を示す。7 行目に検出故障数を示す。8 行目にテスト不能故障数を示す。9 行目にテストパターン数を示す。10 行目にテスト生成時間(sec)を示す。11 行目にテスト実行時間(cycle)を示す。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約 1.2%~1.9% 低下している。また、制御信号線を考慮したテスト生成結果では、制御信号線を考慮しないテスト生成結果に比べて故障検出率は大きく向上しなかった。故障検出効率が低下したのは、コントローラと比較器の故障が検出できていないためである。コントローラの故障が検出できない原因は、コントローラの出力論理の故障がデータパスに伝搬した後、その故障値の観測が困難になっているからと考えられる。比較器の故障が検出できない原因は、本 DFT 手法の意図通りにテスト生成ツールが動作していないことが考えられる。

コントローラの故障が検出できない問題に対して、制御信号線に観測ポイントを挿入して実験した。表 7 に観測ポイント挿入後の回路情報を示し、表 8 に観測ポイント挿入後のテスト生成結果を示す。本実験で使用した観測ポイントとは、全ての制御信号線を EXOR ゲートの入力に接続し、出力を 1bit のスキャンレジスタに接続したものである。観測ポイント挿入後は、コントローラの故障が全て検出され故障検出効率は向上したが、面積オーバーヘッドがフルスキャン設計と比較して大きな差がなかった。

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面積オーバーヘッドは削減することができたが、故障検出効率に課題が残る。また、制御信号線に観測ポイントを挿入して故障検出効率の向上を図ったが、面積オーバーヘッドが大きくなってしまった。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と、未検出故障に対する DFT 手法を提案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] H. Wada, T. Masuzawa, K.K. Saluja, and H. Fujiwara, "Design for strong testability of RTL data paths to provide complete fault efficiency, " Proc. of 13th Int. Conf. on VLSI Design, pp.300-305, 2000.
- [3] 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.

- [4] L.M.Flottes, B.Rouzeyre, L.Volpe, "A Controller Resynthesis Based Methods for Improving Datapath Testability", IEEE International Symposium on Circuits and Systems, pp. 347 -350, May 2000.
- [5] W.T.Cheng, "The back algorithm for sequential test generation", Proc. 1988 IEEE Int. Conf. on Computer Design, pp. 66-69, Oct. 1988.
- [6] T.M. Niermann and J.H.Patel, "HITEC : A Test Generation Package for Sequential Circuit", in Proc. Of the European Design Automation Conf., pp.214-218, Feb.1991 .
- [7] 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 5. 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	状態レジスタに追加したFF数	追加状態遷移数	回路面積	面積オーバーヘッド (%)	
Sehwa	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	制御信号線考慮なし	8 / 264	2	54	5093	5.05
			制御信号線考慮あり	9 / 265	3	227	5541	14.35
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	制御信号線考慮なし	8 / 200	2	60	4436	6.20
			制御信号線考慮あり	9 / 201	3	134	4762	14.01
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	制御信号線考慮なし	8 / 200	2	71	5093	6.64
			制御信号線考慮あり	9 / 201	3	216	5451	14.13

表 6. テスト生成結果

回路名	Sehwa				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			制御信号線考慮なし	制御信号線考慮あり			制御信号線考慮なし	制御信号線考慮あり			制御信号線考慮なし	制御信号線考慮あり
テストサイクル数k	10	1	4	4	10	1	4	4	10	1	4	4
故障検出率(%)	73.47	99.84	97.99	98.02	69.98	99.59	98.04	98.43	72.50	99.05	97.86	97.87
故障検出効率(%)	73.57	100.00	98.11	98.17	70.25	99.99	98.39	98.85	73.14	99.99	98.69	98.71
総故障数	17142	18204	18144	20034	15294	16046	16272	17944	17558	18360	18914	20738
検出故障数	12595	18175	17779	19638	10682	15980	15953	17663	12729	18185	18510	20296
テスト不能故障数	22	29	22	29	59	65	58	76	155	173	159	177
テストパターン数	27	103	159	220	39	128	180	235	37	112	230	229
テスト生成時間(sec)	79.18	0.17	20.48	82.24	61.23	0.29	17.70	120.67	94.56	0.34	17.84	131.19
テスト実行時間(cycle)	815	27457	1967	2960	962	25801	2187	3020	898	22601	2829	3067

表 7. 観測ポイント挿入後回路情報

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 (%)
Sehwa	32bit	制御信号線に観測ポイント挿入	10 / 266	3	227	5537	14.21
Maha	32bit		10 / 202	3	134	4845	15.99
Kim	32bit		10 / 202	3	216	5537	15.93

表 8. 観測ポイント挿入後テスト生成結果

回路名	Sehwa	Maha	Kim
テスト容易化設計手法	制御信号線に観測ポイント挿入		
テストサイクル数k	4	4	4
故障検出率(%)	99.50	99.04	98.45
故障検出効率(%)	99.62	99.43	99.29
総故障数	20174	18098	20898
検出故障数	20074	17924	20574
テスト不能故障数	22	71	177
テストパターン数	241	240	251
テスト生成時間(sec)	57.87	111.94	98.07
テスト実行時間(cycle)	3256	3208	3330