システムＬＳＩ設計技術　95－7
（2000．　2．　9）

出力線付加による順序回路のUIO系列生成とテスト

内山 啓介† 大豆生田 利章†† 伊藤 秀男††

† 千葉大学 自然科学研究科
†† 千葉大学 工学部
〒 263-8522 千葉市稲毛区弥生町 1-33
E-MAIL:{uchiyama,mame,ito}@icsd2.tj.chiba-u.ac.jp

あらまし　我々はこれまでに，状態制御系列，活性化系列，UIO系列からなる単一テスト系列と呼ばれるテスト系列を生成することによる，テスト生成複雑度が比較的低く，テスト系列が短い順序回路のテスト系列生成手法を提案してきた．しかし，これまでの研究により，UIO系列を持たない場合に故障検出率が低下してしまうという問題がわかっている．本稿ではこのような問題を考慮し，有限状態機械（FSM）に対し状態の識別を行うためのテスト用外部出力線を付加するテスト容易化合成を行い，UIO系列探索を容易にする手法を提案する．また，提案手法を順序回路テストに適用し，ベンチマークFSMを用いて評価実験を行った．その結果，従来手法であるHITECに比べ多くの回路で故障検出率が高いことを確認した．
キーワード　順序回路，UIO系列，付加外部出力線，状態遷移

UIO Sequence and Test Generation for Sequential Circuits Using Extra Outputs

Keisuke Uchiyama† Toshiaki Ohmameuda†† Hideo Ito††

† Graduate School of Science and Technology, Chiba University
†† Faculty of Engineering, Chiba University
1-33, Yayoi-cho, Inage-ku, Chiba 263-8522, Japan
E-MAIL:{uchiyama,mame,ito}@icsd2.tj.chiba-u.ac.jp

Abstract　*The authors have proposed a test generation method for sequintail circuits. It has advantages making generation complexity low and length of test sequence short. But there is problem that fault coverage is low when there is not UIO sequence. So, in this paper,we propose a method generating UIO sequence used Extra Output(EO), which is used to distinguish states. Our method adds a EO for finite state machine used synthesis for testability(SFT). The experimental results show that fault coverage is hegher than the conventional method "HITEC".*

key words　sequential circuit, UIO sequence, extra output, state transition

1 はじめに

状態遷移を利用した順序回路のテスト生成手法は，順序回路のどの記述レベルを用いてテスト生成を行なうかにより，機能レベルのテスト生成手法とゲートレベルのテスト生成手法の2つに大きく分けることができる．機能レベルのテスト生成手法は，全ての遷移を外部出力から観測する checking experiment 手法 [12]，状態遷移故障を検出する手法 [13] 等がある．しかし，これらの手法はテスト系列長が長くなるという問題点がある．もう一つのゲートレベルのテスト生成手法の大部分は，回路構造に依存する手法である．回路構造に依存する手法には，故障の影響を伝搬する前方操作とドントケア状態からテスト状態を正当化する後方操作を用いる手法 [7] 等がある．これらの手法は，ゲート数の増加や回路構造の複雑さがテスト生成複雑度に与える影響が大きいという問題がある．

このような背景をもとに，我々はこれまでに，比較的低いテスト生成の複雑度で，従来手法よりも短い単一縮退故障テストのためのテスト系列を生成する研究をおこなってきた [2][3]．この手法は，機能レベルとゲートレベルの中間に位置づけられる手法である．ここで生成するテスト系列は単一テスト系列と呼ばれる部分的系列で構成する．単一テスト系列は，活性化状態へ遷移を行う状態制御系列と，順序回路内の組合せ回路の縮退故障を活性化する活性化ベクタ，活性化ベクタ印加後の状態を外部に伝搬させるための UIO（Unique Input/Output）系列 [18] からなる．この手法ではテスト生成の前にあらかじめ組合せ回路部の最小活性化ベクタ集合と UIO 系列を求めておく．これらの情報を利用して，局部的に系列長が最小となる部分系列を選択して連結していくため，短いテスト系列が得られる．しかし，これまでの研究により，UIO 系列を持たない場合に故障検出が低下してしまうという問題がわかっている．これは UIO 系列を持たないことにより内部状態の観測が困難になり，FF に伝搬した故障値の影響を外部から観測できなくなってしまうためである．また，テスト生成時間の中で UIO 系列生成は大きな割合を占めており，タイムオーバヘッドの面でも問題となっている．このような問題に対し UIO 系列を持たない状態に UIO 系列を持たせることと，UIO 系列生成時間の短縮を目的とし，有限状態機械 (FSM:Finite State Machine) に対し状態の識別を行うためのテスト用外部出力線 Extara Output(EO) を付加することによるテスト容易化合成（SFT:Synthesis for Testability) を行う手法が提案されている [4]．テスト容易化合成手法は，現在盛んに行われている設計手法であるトップダウン設計を背景に，設計の上流行程で（ここでは論理合成時にテスト機能を組み込んだ回路に対して）論理圧縮を行うので，面積オーバヘッドを抑えることができる．この手法はテスト生成時間の中で大きな部分を占める UIO 系列生成時間を短縮するために，状態識別用の EO を付加した上で，UIO 系列を生成するための探索空間の探索を幅優先探索で行い生成の高速化を行う手法である．しかし，EO を付加することにより探索空間が大きくなることと幅優先探索のため広範囲の空間探索を行うことができず UIO 系列の生成数に関しては問題が残った．

本稿ではこのような問題点を考慮し，EO を利用した UIO 系列生成の改良手法（テスト容易化合成手法）を提案する．本手法ではあらかじめ EO を考慮しない探索空間を深さ優先探索で探索し，状態のグループ分けを行う．その分割されたグループ内の状態を判別するための論理値を EO に設定する．これにより，探索空間が大きくなってしまうという問題を避け，より広い空間探索が可能となり多くの UIO 系列を生成できる．また，本手法を順序回路テストに適用し，シミュレーションにより評価を行い，本手法が順序回路テストに対しても有効であることを示す．

本稿の構成は以下のようになっている．2節で語句の定義について述べ，3節で提案手法と UIO 系列について述べる．4節では評価実験について述べ，5節でまとめと今後の課題について述べる．

2 諸定義

ここで，本稿で使用する用語の定義と，テスト系列の基本系列について述べる．

2.1 用語の定義

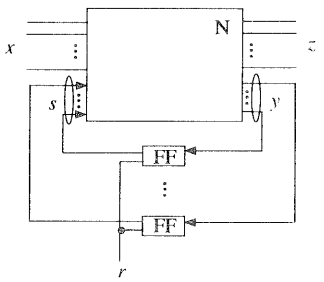

図 1　順序回路のモデル

図1に示すような，組合せ回路（N）とフリップフロップ（FF）からなる同期式順序回路（M）を対象とする．ここで，x, s, z, y は一次入力，現状態入力，一次出力，次状態出力であり，r はリセット入力である．状態遷移関数を

f，出力関数を g とすると次の関係がある．

$$y = f(x, r, s), \quad z = g(x, r, s)$$

x, s, z, y の集合をそれぞれ X, S, Z, Y で表す．

lビットの一次入力線を各々$x(0), x(1), \ldots, x(l-1)$ と表記すると，$x = x(0)x(1)\cdots x(l-1)$ である．ここで，$x = x\&r$（&はビット結合子），すなわち $x = x(0)x(1)\cdots x(l-1)r$ として y を次のように書き直す．

$$y = f(x, s), \quad z = g(x, s)$$

以後，この表現を用いる．

M へクロックに同期して一次入力 $x_0, x_1, \ldots, x_{n-1}$ に次々と加える系列を $\tilde{x} = x_0 x_1 \cdots x_{n-1}$ と記す．このときの y の出力系列 $\tilde{y}$，および z の出力系列 $\tilde{z}$ を次式で示す．

$$\tilde{y} = \tilde{f}(\tilde{x}, s), \quad \tilde{z} = \tilde{g}(\tilde{x}, s)$$

更に，この時の $\tilde{y}$ の最後の出力を単に y，および $\tilde{z}$ の最後の出力を単に z とし，次式で記す．

$$y = f(\tilde{x}, s), \quad z = g(\tilde{x}, s)$$

本稿では対象にする順序回路は強連結であり，その動作は状態遷移図（状態遷移表）によって与えられると仮定する．

[定義 1]（有効状態と無効状態） 順序回路 M の状態遷移図に現れる状態を有効状態とし s_v と表記する．M の全状態集合 S において，有効状態ではない状態を無効状態とし s_i と表記する s_v, s_i それぞれの集合を S_v, S_i と表記する（$S = S_v \cup S_i$）．

以後，単に状態と記す時は有効状態を示すことに注意を要する．

[定義 2]（活性化ベクタ） N を，$x(= x\&r), s$ を入力，z, y を出力とする組合せ回路とみるとき，N の単一固定縮退故障集合 F を検出する m 個のテストからなるテスト集合を

$$\begin{aligned} V &= \{v(0), v(1), \ldots, v(k), \ldots, v(m-1)\}, \\ & v(k) = (x_k, s_k) \quad (k = 0, \ldots, m-1) \end{aligned}$$

と表す．x_k, s_k は $v(k)$ の成分であり，$x_k \in X, s_k \in S$ である．$v(k), x_k, s_k$ を各々，活性化ベクタ，活性化入力，活性化状態と呼ぶ．

[定義 3]（リセット入力とリセット状態） M の FF をリセット状態に設定する外部入力をリセット入力と言い，図 1 に示すように，r と表す．また，リセット状態を s_r と記す．電源投入後の初期状態はドントケア状態であり，s_* と表記する（$s_* \in S$）．

本手法では，テスト系列の最短化のため s_r には，全ての有効状態への平均遷移距離が最も短い中央状態 [6] を用いる．

[定義 4]（転送系列） M の状態を i から j へ遷移させる入力系列を $\tilde{t}_{ij}$ と記し，転送系列という．ここではいくつかの $\tilde{t}_{ij}$ の中で最短の長さの入力系列を単に転送系列と呼ぶ．

[定義 5]（状態制御系列） M の状態を i から j へ遷移させる入力系列として，状態制御系列 $\tilde{a}_{ij}$ を定義する．

$$\tilde{a}_{ij} = \tilde{t}_{ij} \mid r\, \tilde{t}_{s_r j}$$

ただし，$i = j$ の場合は，$\tilde{t}_{ij}$ は空系列と定義する．また，$\tilde{t}_{ij} | r\tilde{t}_{s_r j}$ は，$\tilde{t}_{ij}$ と $r\tilde{t}_{s_r j}$ の 2 つの系列の短い方を示す．

状態制御系列の集合を以下のように表す．

$$\begin{aligned} \tilde{A}_{ij} &= \{\tilde{a}_{ij}\}, \\ \tilde{A}_i &= \bigcup_{j=0\sim n-1} \tilde{A}_{ij} \ (n : \text{状態数}), \\ \tilde{A} &= \bigcup_{i=0\sim n-1} \tilde{A}_i. \end{aligned}$$

[定義 6]（UIO（unique input/output）系列） [5] 次式を満たす最短な入力系列 $\tilde{u}_i$ を状態 i の UIO 系列という．

$$\tilde{g}(\tilde{u}_i, i) \neq \tilde{g}(\tilde{u}_i, s), \quad \forall s \in S_v, \quad s \neq i$$

状態 i に対する UIO 系列 $\tilde{u}_i$ の中で，到達状態が状態 j であるものを $\tilde{u}_{ij}$ と表す．一般に $\tilde{u}_i$ や $\tilde{u}_{ij}$ は複数個存在する．

状態 i の UIO 系列の集合を以下のように表す．

$$\begin{aligned} \tilde{U}_{ij} &= \{\tilde{u}_{ij}\}, \\ \tilde{U}_i &= \bigcup_{j=0\sim n-1} \tilde{U}_{ij} \ (n : \text{状態数}), \\ \tilde{U} &= \bigcup_{i=0\sim n-1} \tilde{U}_i. \end{aligned}$$

UIO 系列は存在しないこともある．そのような場合は長さ 0 の系列であると考える．

2.2 テスト系列の定義

現時点までに生成されている系列の最後尾へ，基本生成系列と呼ぶ局部的に最適な部分系列を連結していくことで，テスト系列を生成する．

[定義 7]（基本生成系列） 活性化ベクタ $v(k-1) \in V$ を N へ印加（M の状態を $s = s_{k-1}$ として x_{k-1} を印加）した後，次に $v(k)$ を N へ印加するまでに加える入力系列

$\tilde{w}(k) = \tilde{u}_{ij}\tilde{a}_{js_k}x_k$, $i = f(x_{k-1}, s_{k-1})$ を基本生成系列と言う.

状態 s_k のときに一時入力 x_k を M へ印加することを, とくに誤解を与えない限り, $v(k)$ を印加するとも言う. $\tilde{u}_{ij}$ は $v(k-1)$ の印加に対する UIO 系列であり, $v(k)$ 印加のための系列ではないことに注意を要する. 添字 k は, テスト系列を作成していく上で k 番目 ($k = 0, 1, \ldots, m-1$) の基本生成系列であることを意味し, $\tilde{w}(k)$ を作成する上で用いられた V の要素を改めて $v(k)$ と表現していることに注意を要する.

[定義 8] (単一テスト系列) $\tilde{w}(k) = \tilde{u}_{i_1j_1}\tilde{a}_{j_1s_k}x_k$, $\tilde{w}(k+1) = \tilde{u}_{i_2j_2}\tilde{a}_{j_2s_{k+1}}x_{k+1}$ を連結して得られる部分系列の一部

$$\tilde{t}(k) = \tilde{a}_{j_1s_k}x_k\tilde{u}_{i_2j_2}$$

を単一テスト系列と言う. ここで, $i_2 = f(x_k, s_k)$ であり, $\tilde{u}_{i_1j_1}, \tilde{u}_{i_2j_2}$ は各々M へ $v(k-1), v(k)$ を印加した後の状態 i_1, i_2のUIO 系列である.

$\tilde{t}(k)$ は, 入力 $\tilde{a}_{j_1s_k}$ によって M の状態を s_k に設定し, M に x_k を印加することで $v(k) = (x_k, s_k) \in V$ を N に加えている. このとき, 出力 Z に異常が出れば故障は容易に検出できる. また, 状態 $i_2 = f(x_k, s_k)$ に異常が出れば, $\tilde{u}_{i_2j_2}$ の印加によって故障があったことが検出できる. すなわち, $v(k) \in V$ の印加によって検出される N 内の故障は, $\tilde{t}(k)$ の印加によって検出できることになる. 但し, $\tilde{w}(k)$ や $\tilde{t}(k)$ で用いる UIO 系列と状態制御系列は, 故障の無い M において求めた系列であり, 故障がある場合において必ず有効であるとは限らない. しかし, 高い確率で有効であることが示されている [2][3].

このように, 本稿のテスト系列は, 部分的に単一テスト系列を印加することにより故障を検出することを基本としている. 但し, テスト系列の生成単位は, 定義 7 に述べた $\tilde{w}(k)$ としている点に注意を要する.

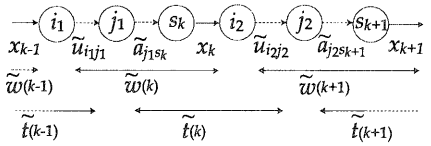

図 2　基本生成系列と単一テスト系列との関係

2.3 UIO 系列生成に使用する定義

提案手法で利用する外部出力線 extra output と UIO 系列生成に関係する用語について説明する.

[定義 10](状態グループ) ある系列を印加された時に同じ出力値をとる状態同士をグループとして扱い, これを状態グループとする.

[定義 11](Extra Output(EO)) 本手法では検査用外部出力線を 1 本付加する. m 本の一次出力線に対して付加する外部出力線を z_e とし, Extra Output(EO) と呼ぶ. EO の論理値は

$$z_e = g_e(x, i)$$

と表される. 状態 i の UIO 系列となるように $g_e(x, i)$ の論理値を決定する. また, UIO 系列に関係のない場合 $g_e(x, i) = *$ であり論理値はドントケアとなる.

[定義 12](トリビアル (tribial)) 状態グループが要素として 1 つの状態だけ含むこと. (例 : (A), (B))

[定義 13](ホモジーニアス (homogeneous)) 状態グループが要素として 1 つだの状態だけを含むか, 同じ状態のみを含むこと. (例 : (A), (BB))

3 UIO 系列

ここではまず UIO 系列生成について示し, その後提案手法を示す.

3.1 UIO 系列生成

一般的な UIO 系列生成 (外部出力線を付加していない場合) について説明する. UIO 系列を求めるには UIO 木を生成する.
以下に UIO 系列生成アルゴリズムを示す.

Step1: 状態グループ $G = \{s_1, \ldots, s_k\}$ とする.

Step2: 状態グループ G に対し入力を印加したときの出力値でグループ分けする.

Step3: ノードの状態グループの内容に応じて以下の処理を行う.

条件 1: ノードにおいて, ある状態グループがトリビアルである. これによりグループに所属する状態の初期状態に対する UIO 系列が見つかったので, このグループを部分終端とする.

条件 2: ノードにおいて, ある状態グループがトリビアルでなくホモジーニアスである. このグループを部分終端とする. このとき, UIO 系列は見つかっていない.

条件 3: ノードにおけるホモジーニアスでない状態グループと同じ状態を要素として持つ状態グループが先行ノードですでに出現している. このとき, このグループを部分終端とする.

表 1　FSM の例

現状態	次状態，出力	
	0	1
A	C,1	D,0
B	D,0	B,1
C	B,0	C,1
D	C,0	A,0

条件 4: まだ，終端していないグループがあるならば，そのグループを G とし **Step2** へ戻る．そうでなければ終了．

以下，UIO 系列探索アルゴリズムを具体的に説明する．探索する UIO 木の例を図 3 に示す．この UIO 木は表 1 の FSM の UIO 木であり，図中の楕円は部分終端を表している．この図の頂点で 0 を印加する．このとき出力によって状態をグループ分けすると初期状態は (B，C，D) と (A) というグループに分割でき，この初期状態から遷移した現状態グループは (D，B，C) と (C) となる．このとき系列 0 は初期状態 (A) の UIO 系列となる．さらにこの状態グループに 0 を印加すると，初期状態 (B，C，D) と (A) というグループに分割できる．しかし，この状態グループは前のノードで出現しており条件 3 により部分終端となるのでノードを 1 つバックトラックする．次に 1 を印加する．このとき初期状態 (B)(C，D)(A) という具合にグループ分割でき，初期状態 (B) の UIO 系列が求まったことになる．以下アルゴリズムに従って同様の処理をしていくと，状態 A の UIO 系列は 0（到達状態 C)，10(C)，状態 B の UIO 系列は 01(A)，101(A)，状態 C の UIO 系列は 0101(A)，101(B)，状態 D の UIO 系列は 0101(B)，10(C) となる．

この例からもわかるように UIO 系列は 1 つの状態に対して 1 つとは限らない．また，この例では全ての状態に対しての UIO 系列が存在したが，どの回路のどの状態に対しても UIO 系列が存在するという保証はなく，UIO 系列が存在しない場合には，UIO 系列を持たない状態を外部から観測できなくなることが問題となる．

3.2　提案手法

UIO 系列についてはその存在が必ずしも保証されているものではなく，また，探索空間の大きさから全ての入力系列について探索することは難しい．一方，これまでの研究により単一テスト系列を印加することによるテスト手法では UIO 系列を持たない場合に故障検出率が低下してしまうという問題がわかっている．そこで，UIO 系列を持たない状態に UIO 系列を持たせることを目的とし，

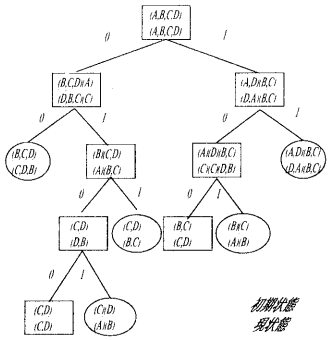

図 3　UIO 木

有限状態機械（FSM：Finite State Machine) に対する出力線 z_e (EO:Extra Output) を付加したテスト容易化合成を行うことによる UIO 系列生成手法を提案する．提案手法は UIO 木を深さ優先探索で空間探索し状態のグループ分けを行う．そのグループの内容により EO の値を決定し UIO 系列生成を容易にする手法であり，面積オーバヘッドを考慮しドントケアを積極的に EO に割り当てている．以下に EO の論理値を決定し，UIO 系列生成を行うアルゴリズムを示す．

Step1: 有効状態集合を S_v とする

Step2: 深さ優先探索に従い入力パタンを印加する．その出力値によってグループ分けをする．そのグループの内容によって以下の処理を行う．

条件 1: ノードにおいて，ある状態グループがトリビアルである．これによりグループに所属する状態の初期状態に対する UIO 系列が見つかったので，このグループを部分終端とする．この状態を S_v から取り除き z_e にドントケアを割り当てる．

条件 2: ノードにおいて，ある状態グループがトリビアルでなくホモジーニアスである．このグループを部分終端とする．このとき，UIO 系列は見つかっていない．

条件 3: ノードにおけるホモジーニアスでない状態グループと同じ状態を要素として持つ状態グループが先行ノードですでに出現している．このとき，このグループを部分終端とする．

条件 4: まだ，終端していないグループがあり，そのグループ内の状態数が 2 つならば，それぞれの z_e に 1 と 0 をそれぞれ割り当て S_v から取り除く．

Step3: 全ての状態に対しUIO系列が生成された時点で終了．そうでなければ次へ

Step4: あらかじめ発見的手法で求めておいた深さまで全数探索を行った結果，UIO系列が生成されていない状態がある．そのとき最後に生成したグループに対し以下の処理を行う．

G_p に属する状態数を $N_G(p)$，G_p に属する q 番目の状態を $G_p(q)(0 \leq p \leq l, 0 \leq q \leq N_G(p)-1$，但し，$l \leq n$　n：状態数) と表す．

loop $p = 0 \sim l$

case $N_G(p) > 1$

$g_e(x, G_p(0)) \leftarrow 1$

loop $r = 1 \sim N_G(p) - 1$

$g_e(x, G_p(r)) \leftarrow 0$

loop end

loop end

表1のFSMを例に提案手法について説明する．初期状態グループに対し0を印加する．これにより(A)というグループと(B,C,D)というグループに状態がグループ分けされる．まず条件1に基づき(A)に対するUIO系列生成は終了となる．次にこのノードに対し0を印加すると条件3により部分終端となる．深さ優先探索法により1つ前のノードにバックトラックし，今度は入力1を印加する．これにより(B)というグループと(C,D)というグループに分割される．図4に示すように，条件1により(B)に対するUIO系列生成は終了する．また(C,D)も条件4の処理によりUIO系列生成が終了となり，全ての状態に対しUIO系列が生成できたことになる．

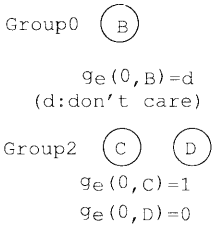

図4　EOによる状態のグループ分け

表2に使用するベンチマークFSMの概要と提案手法と従来手法によるUIO系列生成数，EO付加による面積オーバヘッドを示す．#STは有効状態数，#FFはFF数，#INは入力線数，#OUTは出力線数，#GAはゲート数，#nonEOはEOを用いないUIO系列生成の結果，#H_EOは長谷川らの手法[4]によるUIO系列生成の結果，#P_EOは提案手法によるUIO系列生成の結果，#H_areaは長谷川らの手法による面積オーバヘッド，#P_areaは提案手法による面積オーバヘッドをそれぞれ表している．長谷川らの手法のdk16，提案手法のex4の面積オーバヘッドがマイナスとなっているのは論理合成する際に論理圧縮が行われた結果EO付加前の回路よりも小さな回路となったためである．このように，EOを利用しない手法，長谷川らの手法よりも，多くのUIO系列を生成する事ができた．

4　評価

提案手法によるUIO生成手法を順序回路の単一縮退故障テストに適用し，シミュレーション実験により提案手法の評価を行う．また，よく利用されている時間展開手法を用いたテスト生成ツールのHITEC[17]および長谷川らの手法との比較検討を行なう．

4.1　実験環境

提案手法を用いて生成したテスト系列の故障検出率を確認するための故障シミュレーションには，HITECの結果と比較することを考慮してPROOFS[17]を用いた．シミュレーション実験は，128MBのメモリを搭載したSUN SPARCstation 5を使用して行なった．

MCNCベンチマークFSMを用いて行ったシミュレーション実験の結果をHITEC，単一テスト系列手法との比較として表3に示す．評価項目として，テスト系列長(test length)，故障検出率(coverage)，テスト生成時間(generation time)の3項目を使用した．また，FSMの論理合成はsynopsys社のDesign Compiler[19]を使用した．

4.2　結果と考察

シミュレーション実験の結果，テスト系列長の大幅な増加を招くことなく，多くの回路で従来手法に比べ高い故障検出率を得ることができた．また，テスト生成時間に関しては，大きな回路ではHITECに比べ短くなる傾向が見られるが，入力線数や状態数の多い回路ではテスト生成時間が長くなる傾向が見られる．これは以下の理由による．HITECは順序回路のテスト生成複雑度で直接テスト系列を生成しているため，ゲート数増加による影響を受けやすい．しかし，提案手法ではUIO系列の探索がテスト生成時間の大きな部分を占めている．そのためゲート数の増加よりも探索空間の大きさを決定する要素である入力線数と状態数が多い回路でUIO系列生成に時間がかかり，テスト生成時間全体に影響を与え長くなる

表 2　ベンチマーク FSM の概要と UIO 系列生成数（生成成功数／全有効状態数）

circuits	#ST	#FF	#IN	#OUT	#GA	#nonEO	#H_EO	#P_EO	#H_area[%]	#P_area[%]
bbtas	6	3	3	2	28	6/6	4/6	6/6	35.71	9.68
cse	16	4	8	7	190	14/16	16/16	16/16	24.12	18.81
dk14	7	3	4	5	97	3/7	7/7	7/7	11.65	25.29
dk15	4	2	4	5	67	3/4	4/4	4/4	2.99	8.33
dk16	27	5	3	3	275	20/27	9/27	24/27	-0.36	1.85
dk27	7	3	2	2	25	4/7	4/7	7/7	13.79	7.69
dk512	14	4	2	3	71	10/14	5/14	13/14	12.12	7.20
ex1	20	5	9	19	226	17/20	20/20	20/20	0.94	0.58
ex4	13	4	6	9	76	11/13	13/13	13/13	21.05	-0.14
ex6	7	3	6	8	135	7/7	7/7	7/7	5.56	9.74
keyb	19	5	8	2	225	7/19	18/19	19/19	63.56	56.84
opus	9	4	6	6	111	6/9	9/9	9/9	6.17	2.55
planet	48	6	8	19	504	46/48	48/48	48/48	6.17	5.72
pma	24	5	7	8	182	24/24	24/24	24/24	0.00	0.00
sand	32	5	12	9	432	15/32	32/32	32/32	19.56	15.00
sse	13	4	8	7	145	10/13	13/13	13/13	23.31	14.43
styr	30	5	10	10	487	15/30	30/30	30/30	20.00	13.75
tav	4	2	3	5	19	4/4	3/4	4/4	65.38	17.78

ためである．

また，長谷川らの手法と比較した場合，UIO 系列を多く発見したことにより順序回路の故障検出率も上がっている．これは，従来手法では FF に伝搬してしまい検出できなくなっていた故障を UIO 系列をより多く持つことで観測できるようになったことと，出力線付加により回路内部の可観測性が向上したためと考えられる．

5　おわりに

本稿では，出力回路付加による UIO 系列生成手法の提案を行い，それを順序回路テストに適用した．評価のためのシミュレーション実験を行い，提案手法を先行研究である HITEC，長谷川らの手法との比較を行った．その結果，大幅なテスト系列長の増加をすることなく高い故障検出率を得ることができた．

以上の結果よりテスト系列長・テスト生成時間のオーバヘッドが大きく増加することなく，高い故障検出率を得ることができることが示された．これにより，本手法は順序回路のテスト容易化合成手法として有効な手法であると言うことができる．

今後の課題は，

- テスト生成時間の短縮
- 多重故障に対する本手法の適用の検討
- 大規模回路への適用

などが挙げられる．

参考文献

[1] S.Bommu, S.T.Chakradhar and K.Doreswamy, "Static compaction using overlapped restoration and segment pruning", IEEE/ACM Int'l. Conf. on CAD, pp. 140-146, Nov. 1998.

[2] 長谷川，三浦，大豆生田，伊藤，"状態遷移図と組合せ回路部テストを利用した順序回路のテスト生成"，信学論 B-I 投稿中

[3] 長谷川, 三浦，大豆生田，伊藤, "組合せ回路部テストと状態遷移を利用した順序回路のテスト生成", 信学技報 IDC98-47, FTS98-47, pp. 85-92, May 1998.

[4] 長谷川，大豆生田，伊藤, "テスト容易化合成による順序回路 UIO 系列生成時間の短縮化", 信学秋季ソ大会情報・システム論文集, p.110, Oct. 1998.

[5] D.Lee and M.yannakakis, "Testing Finite-State Machines:State Identification and Verification", IEEE Trans. on Comput. , vol.43, No.3, pp.306-320, Mar.1994.

[6] Frank F.Hsu, Janak H.Patel, "A Distance Reducution Approach to Design for Testability", IEEE 13th VLSI Test symp. , pp.158-163, 1995.

[7] T.Niermann, J.Patel "HITEC : A Test Generation Package for Sequential Circuits" , The European Design Automation Conference , pp.214-218, 1991.

[8] A. Ghosh, S. Devadas and A.R. Newton, "Test Generation and Verification for Highly Sequential Circuits",

表 3　MCNC ベンチマーク FSM を用いた HITEC との比較結果

circuits	test length			coverage[%]			generation time[s]		
	HITEC	Hase.	Prop.	HITEC	Hase.	Prop.	HITEC	Hase.	Prop.
bbtas	102	89	73	98.81	98.81	100.00	0.12	4.28	0.57
cse	272	251	343	100.00	100.00	100.00	2.68	6.10	8.91
dk14	83	53	86	100.00	98.83	100.00	0.12	0.25	0.67
dk15	55	59	69	100.00	98.84	100.00	0.02	0.10	0.28
dk16	550	301	201	99.85	98.91	99.60	11.45	10.65	12.52
dk27	42	23	57	97.56	97.56	100.00	0.03	0.00	0.85
dk512	108	96	101	99.43	98.82	99.78	0.78	0.13	0.85
ex1	240	221	243	99.64	98.58	100.00	5.60	10.62	5.71
ex4	76	72	138	99.54	97.22	100.00	0.47	6.32	0.48
ex6	137	104	93	100.00	100.00	100.00	0.60	0.52	1.17
keyb	523	365	522	100.00	98.38	100.00	6.07	7.32	35.00
opus	117	97	89	99.30	95.83	98.56	1.03	0.37	0.57
planet	519	464	383	99.50	98.92	100.00	83.38	14.62	52.50
pma	205	141	239	99.78	97.84	98.48	12.13	5.80	4.85
sand	572	279	359	99.62	98.26	100.00	68.47	11.05	64.19
sse	154	138	197	99.69	99.08	99.74	2.02	5.82	4.95
styr	697	473	662	99.82	99.20	100.00	40.15	61.90	80.75
tav	23	65	61	94.82	94.87	100.00	0.07	7.75	0.51

IEEE Trans. On Computer-Aided Design, Vol.10, No.5, pp.652-667, 1991

[9] P.Geol, "An implicit enumeration algorithm to generate test for combinational logiccircuit" , IEEE Trans. Computers , vol.C-30, No.3, pp. 215-222, 1981.

[10] T.M.Niermann, R.K.Roy, J.H.Patel, J.A.Abraham, "Test compaction for sequential circuits" , IEEE Trans. on CAD, vol.11, No.2, pp. 260-267, 1992.

[11] D.K.Das, S.Ohtake, H.Fujiwara, "New DFT Techniques of Non-Scan Sequential Circuits with Complete Fault Efficiency" ,IEEE 8th proceeding the Asian Test symp., pp. 263-268, Nov. 1999.

[12] Parag K.Lala, "Fault Tolerant Fault Testable Hardware Design" ,Prentice/Hall Int'l, 1985.

[13] Kwang-Ting Cheng and Jing-Yang Jou, "Functional Test Generation For Finite State Machines" ,Int'l Test Conf., pp. 162-168, 1990.

[14] 小谷, 樋上, 樹下, "リセット状態を用いたテスト系列の静的圧縮について" , 第 38 回 FTC 研究会資料, 1998

[15] 大竹, 和田, 増澤, 藤原, "完全故障検出率を保証するレジスタ転送レベルでの非スキャンテスト容易化設計技法", 信学技報 FTS99-53-64, pp.47-54, 1999.

[16] Parag K.Lala, "Digital circuit Testing and Testability" , Academic Press, 1997.

[17] HITEC/PROOFS, "A Sequentail Circuit Test Generation and Faultsimulation Package", http://www.crhc.uiuc.dec/IGATE/hitecsoftware.html

[18] Davie Lee and Mihalis Yannakakis, "Testing Finite-State Machines:State Identification and Verification", IEEE Trans. Comput., Vol.43, No.3, pp. 306-320, Mar. 1994.

[19] Synopsys, "Design COMPILER REFERENCE MANUAL:FUNDAMENTALS", synopsys,Inc., 1997.