回路疑似変換による順序回路テスト生成の一手法

大竹 哲史　　井上 智生　　藤原 秀雄

奈良先端科学技術大学院大学 情報科学研究科

〒 630-01 奈良県生駒市高山町 8916-5

e-mail: *satosi-o@is.aist-nara.ac.jp*

あらまし　組合せテスト生成複雑度でテスト生成可能な順序回路では、その順序回路のテスト生成問題は、フリップフロップを信号線に置き換えた (組合せ変換した) 組合せ回路におけるテスト生成問題に帰着できる。本論文では、この性質を一般の順序回路に拡張する。具体的には、与えられた順序回路において、平衡構造を有する部分回路を抽出し、その部分回路を組合せ変換した順序回路についてテスト生成を行い、得られたテスト系列を元の順序回路のテスト系列に戻すというテスト生成法を提案する。この手法では、組合せ変換によってフリップフロップ数が減少するので、元の順序回路よりフリップフロップ数が減少している分だけテスト生成時間の短縮が期待できる。提案する手法の有効性を ISCAS'89 ベンチマーク回路による実験によって評価する。

キーワード:　順序回路、テスト生成、無閉路構造、平衡構造

An Approach to Sequential Test Generation by Circuit Pseudo-Transformation

Satoshi OHTAKE, Tomoo INOUE, and Hideo FUJIWARA

Graduate School of Information Science,
Nara Institute of Science and Technology (NAIST)

8916-5, Takayama-cho, Ikoma, Nara, 630-01

e-mail: *satosi-o@is.aist-nara.ac.jp*

Abstract　The test generation problem for a sequential circuit *capable of generating tests with combinational test generation complexity* can be reduced to that for the combinational circuit formed by replacing each FF in the sequential circuit by a wire. In this paper, we consider an application of this approach to general sequential circuits. We propose a test generation method using *circuit pseudo-transformation* technique: given a sequential circuit, we extract a subcircuit with *balanced structure* which is capable of generating tests with combinational test generation complexity, replace each FF in the subcircuit by wire, generate test sequences for the transformed sequential circuit, and finally obtain test sequences for the original sequential circuit. We also estimate the effectiveness of the proposed method by experiment with ISCAS'89 benchmark circuits.

Key words:　Sequential circuits, test generation, acyclic structure, balanced structure

1. はじめに

近年のデバイス技術の進歩により、ディジタル集積回路の集積度が向上し、大規模なシステムを VLSI 上に実装することが可能となった。しかし、回路の大規模化に伴い、テストはますます困難となり、費用のかかる問題となっている。

増大するテスト費用を抑える 1 つの方法として、テスト生成時間を短縮する方法が考えられる。順序回路の場合、フリップフロップ (FF) が存在するため、テスト生成アルゴリズムの探索空間は FF 数を n とすると 2^n に依存する。この FF 数を減らすことができれば、テスト生成アルゴリズムの探索空間が縮小され、FF 数が減った分だけテスト生成時間が短縮されることが期待できる。FF 数を減らす方法として、リタイミングを用いて疑似的に回路中の FF を回路外へ移動するソフトウェア変換 [1] が提案されている。

組合せテスト生成複雑度でテスト生成可能な順序回路では、その順序回路のテスト生成問題は、FF を信号線に置き換えた組合せ回路におけるテスト生成問題に帰着できる。組合せテスト生成複雑度でテスト生成可能な順序回路には、平衡構造 [2] や内部平衡構造 [3] がある。本論文では、与えられた順序回路において、平衡構造を有する部分回路中の FF を疑似的に信号線に置き換えて (回路疑似変換して)FF 数を減らしてテスト生成を行い、得られたテスト系列を元の順序回路のテスト系列に戻すというテスト生成法を提案する。

本論文では、2.章で回路疑似変換を定義し、3.章では、回路疑似変換した回路でテスト生成を行って得られたテスト系列を用いて元の回路をテストするために、d-時刻保持変換と d-時刻保持テスト法を定義する。4.章で、組合せ回路疑似変換と d-時刻保持変換によるテスタビリティの保存性を考察し、テスト生成問題の帰着性について述べる。5.章では、回路疑似変換を用いたテスト生成法を提案し、6.章で提案手法の有効性を実験によって評価する。

2. 回路疑似変換

[定義 **1**] (平衡構造) 順序回路 S に閉路が存在しないとき、S は無閉路構造であるという。無閉路構造の順序回路 S_A の、任意の外部入力と任意の外部出力の対について、その 2 点間のどの経路の順序深度 (経路上の FF 数) も等しいとき、S_A は平衡構造であるという。

[定義 **2**] (回路疑似変換) 回路 S を疑似的に S とは別の回路に変換することを、回路疑似変換と呼ぶ。ここで、“疑似的” とはテスト生成を行うために、“仮に” 変換するという意味であり、物理的に回路を変換するのではない。また、S を回路疑似変換 T した回路を S^T と書く。

[定義 **3**] (核回路) 順序回路 S (図 1) 中の FF の集合を Q とする。Q の部分集合を Q_E として、S から Q_E を取り除いた回路 S_K を核回路と呼ぶ。このとき、Q_E の FF

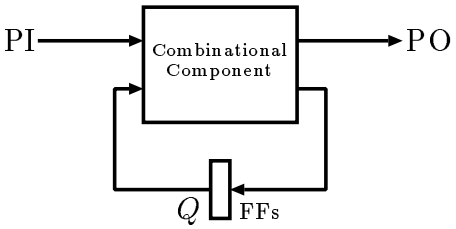

図 1: 順序回路 S

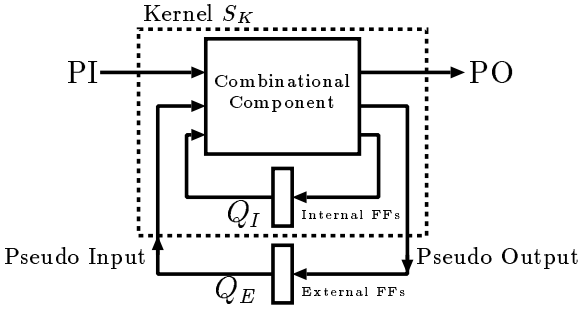

図 2: S を 核回路 S_K と外部 FF Q_E に分割

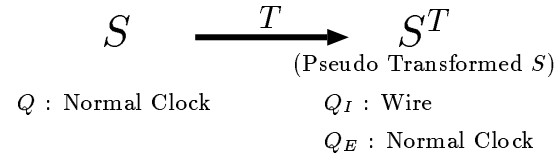

図 3: 組合せ回路疑似変換

を外部 **FF**、$Q_I = Q - Q_E$ の FF を内部 **FF** と呼ぶ。核回路の入力には、S の外部入力と Q_E の FF からの疑似入力があり、出力には、S の外部出力と Q_E の FF への疑似出力がある (図 2参照)。また、核回路が無閉路構造のとき、核回路の入力から出力への経路の順序深度で最大のものを d として、この d を核回路の順序深度と呼ぶ。

[定義 **4**] (組合せ回路疑似変換) 順序回路 S が核回路と外部 FF に分割されたとき、内部 FF を信号線に置き換える変換を組合せ回路疑似変換と呼ぶ (図 3,4参照)。

3. *d*-時刻保持テスト法

[定義 **5**] (***k*-クロック**) k を 0 以上の整数定数とする。通常のクロックの 1 クロックサイクルを $k+1$ に延ばしたクロックを***k*-クロック**と呼ぶ。

[定義 **6**] (***k*-時刻保持 FF**) k を 0 以上の整数定数とする。k 時刻値を保持した後、次の時刻で値を取り込む FF を***k*-時刻保持 1-時刻取り込み FF** と呼び、***k*-時刻保持 FF** と略記する。

通常の FF は 0-時刻保持 FF である。通常のクロックで動作する FF を図 5に示す。k-時刻保持 FF を実現する方法は、図 6(I) のように、k-クロックを用いる方法や、図 6(II) ように Hold/Load 信号による方法などが考えられる。

[定義 **7**] (***d*-時刻保持変換**) 順序回路 S (図 1) が核回路と外部 FF に分割されたとき (図 2)、核回路の順序深度を

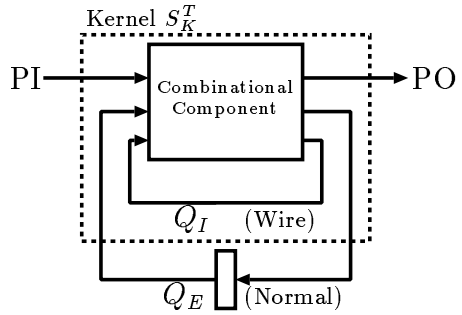

図 4: 順序回路 S^T

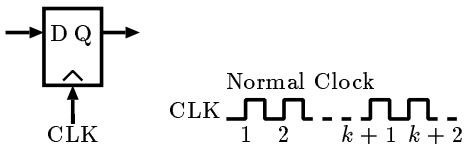

図 5: 通常 FF

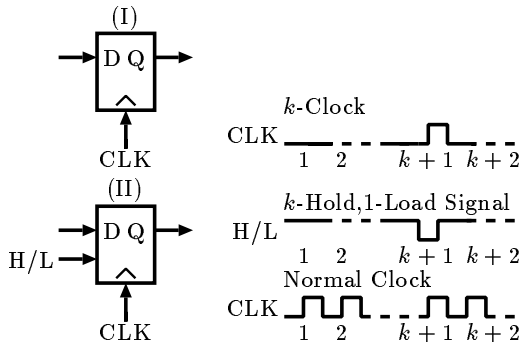

図 6: k-時刻保持 FF

d として、外部 FF を d-時刻保持 FF に置き換える変換を**d-時刻保持変換**と呼び、S を d-時刻保持変換 H した回路を S^H と書く (図 7,8参照)。

[定義 **8**] (**k-時刻保持系列**) 系列 t の系列長を n として、t の各ベクトルを $t_i(i = 1, 2, \ldots, n)$ とする。ただし、k を 0 以上の整数定数として、n を $k+1$ の倍数とする。$1 \leq m \leq n/(k+1)$ となる任意の整数 m について、$(m-1)(k+1) < i, j \leq m(k+1) \wedge i \neq j$ ならば $t_i = t_j$ となるとき、t を **k-時刻保持系列**と呼ぶ。

[定義 **9**] (**d-時刻保持テスト法**) 順序回路 S を 核回路と外部 FF に分割したとき、核回路の順序深度を d として d-時刻保持変換した回路を S^H とする。S^H に対して、テスト系列に d-時刻保持系列を用いて、外部 FF は d-クロックで動作させるテスト法を **d-時刻保持テスト法**と呼ぶ。

この d-時刻保持テスト法では、S^Tは次のような動作をする。

- 外部入力では、核回路全体に信号が伝搬する $d+1$ 時刻の間、d-時刻保持系列により入力値が保持される。
- 疑似入力では、核回路全体に信号が伝搬する $d+1$ 時刻の間、d-時刻保持 FF により値が保持される。
- 外部出力では、通常のテスト法と同様に出力値が観測される。

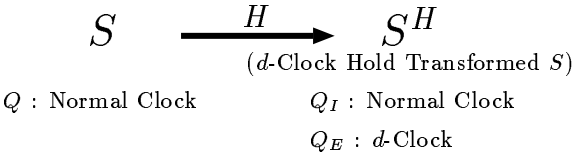

図 7: d-時刻保持変換

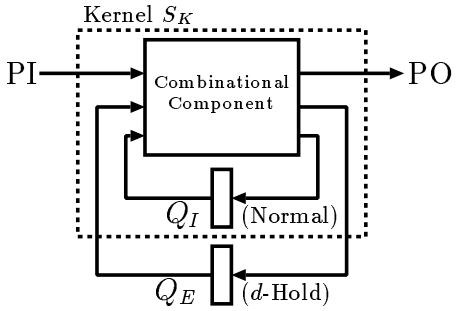

図 8: 順序回路 S^H

- 疑似出力では、d-時刻保持 FF により、核回路全体に信号が伝搬した $d+1$ 時刻目の値が FF に取り込まれる。核回路全体に信号が伝搬していない d 時刻の間の値は無視される。

4. テスト生成問題帰着性

以下では、回路が与えられると、その回路に対するテスト法も与えられるものとする。

[定義 **10**] (テスト可能) 回路 C の故障を f とし、C_f を f によって故障した回路とする。また、C の入力系列 t に対して、t を C に印加して得られた出力系列と、t を C_f に印加して得られた出力系列が異なる場合、t を f のテスト系列と呼び、f は t でテスト可能であるという。

[定義 **11**] (テスタビリティ保存性) 回路 C を変換 τ によって変換して得られた回路を C^τ とする。C、C^τ の全故障集合をそれぞれ F、F^τ とする。このとき、次のすべての条件を満たす変換 τ はテスタビリティを保存するという。

(i) F から F^τ への写像 φ_τ が存在する。

(ii) F の任意の故障 f に対して、f が C のテスト法でテスト可能ならば、$\varphi_\tau(f)$ は C^τ のテスト法でテスト可能である。

(iii) F の任意の故障 f に対して、f が C のテスト法でテスト不可能ならば、$\varphi_\tau(f)$ は C^τ のテスト法でテスト不可能である。

[定義 **12**] (テスト生成問題帰着性) 回路 C を変換 τ によって変換して得られた回路を C^τ とする。回路 C、C^τ の全故障集合をそれぞれ F、F^τ とする。このとき、C と C^τ について次の 2 つの条件を満たすならば、C のテスト生成問題は C^τ のテスト生成問題に帰着できるという。

(i) 変換 τ はテスタビリティを保存する。

(ii) 変換 τ における F から F^τ への写像をφ_τ とする。$\varphi_\tau(F)$ のうち C^τ のテスト法でテスト可能な故

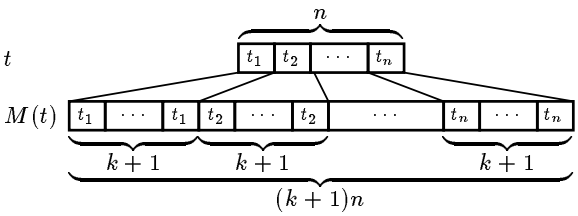

図 9: k-時刻保持系列変換

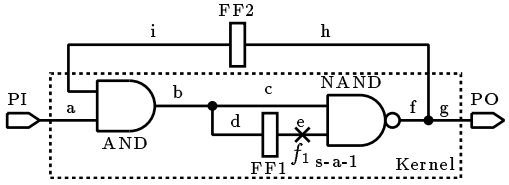

図 10: 順序回路 S_1

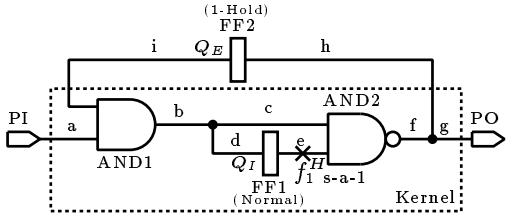

図 11: 順序回路 S_1^H

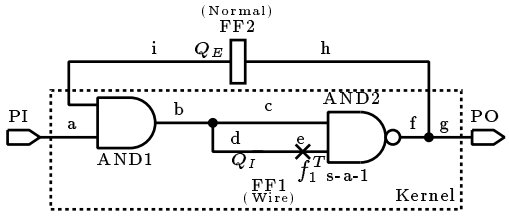

図 12: 順序回路 S_1^T

障の集合を F' とする。F' の任意の故障 f について、$\bigcup_{t\in\mathcal{T}(f)}\{\sigma(t)\}\subseteq\bigcap_{g\in\varphi_\tau^{-1}(f)}\mathcal{T}(g)$ となる写像 σ が存在する。ここで、$\mathcal{T}(f)$ は故障 f に対応する回路のテスト法で f をテスト可能なすべてのテスト系列の集合とする。

[定義 **13**] (***k***-時刻保持系列変換) 系列 t の系列長を n として、t の各ベクトルを $t_i(i=1,2,\ldots,n)$ とする。一方、k を 0 以上の整数定数として、系列長が $n(k+1)$ である k-時刻保持系列 t^k を考える。この t^k の各ベクトルを $t_l^k(l=1,2,\ldots,n(k+1))$ とする。$(i-1)(k+1)<j\leq i(k+1)$ となる任意の整数 i,j に対して $t_j^k=t_i$ となるような t から t^k への変換を ***k***-時刻保持系列変換と呼び、この変換を M と書く。

[例 **1**] (***k***-時刻保持系列変換) 系列長が n の系列 t が与えられたとき、t を k-時刻保持変換 M すると、図 9 のように k-時刻保持系列 $M(t)$ が得られる。

順序回路 S (図 1) を核回路 S_K と外部 FF Q_E に分割したとする (図 2)。内部 FF を Q_I、核回路の順序深度を d する。S を d-時刻保持変換した回路を S^H(図 8)、S を組合せ回路疑似変換した回路を S^T (図 4) とする。S^H のテスト法は d-時刻保持テスト法、S^T のテスト法は通常のテスト法である。ここで、S^H から S^T への変換 HT を考える。変換 HT は、S^H の通常 FF を信号線に、d-時刻保持 FF を通常 FF に置き換える変換である。この S^H と S^T の間のテスト生成問題帰着性に関して、定理 1、2 が成立する。

[定理 **1**] S_K が平衡構造のとき、S^H のテスト生成問題は S^T のテスト生成問題に帰着できる。

(証明) テスト生成問題帰着性の定義 12の条件 (i),(ii) を満たすことを示す。

(定義 12-(i)) 変換 HT がテスタビリティを保存することを示すために、定義 11の条件 (i),(ii),(iii) が成り立つことを以下順に示す。

(i) d-時刻保持変換と組合せ回路疑似変換の定義より、S^H と S^T は Q_I を除いて同形である (図 8,4 参照)。したがって、その部分回路中の故障は、S^H と S^T で 1 対 1 に対応する。一方、S^H における Q_I の FF の入力故障と出力故障は出力故障で代表でき、それは、S^T における対応する信号線の故障に 1 対 1 に対応させることができる。

(ii) S^H の任意の故障を f^H として、f^H に対応する S^T の故障を f^T とする。f^H が S^H のテスト法でテスト可能ならば、f^T は S^T のテスト法でテスト可能であることを示す。f^H がテスト可能ならば、テスト系列 t^H が存在する。S^H のテスト法は d-時刻保持テスト法なので、t^H は d-時刻保持系列である。ここで、t^H の系列長を $(d+1)n$ とする。S^H に入力系列 t^H を印加して得られる出力系列を $O^H=o_1^H,o_2^H,\ldots,o_{(d+1)n}^H$ とする。また、O^H から $d+1$ 周期でベクトルを抽出した系列を $O^{H\prime}=o_{d+1}^H,o_{(d+1)2}^H,\ldots,o_{(d+1)n}^H$ とする。d-時刻保持系列変換 M の逆変換を M^{-1} として、S^T に $M^{-1}(t^H)$ を印加して得られる出力系列を $O^T=o_1^T,o_2^T,\ldots,o_n^T$ とする。このとき、核回路が平衡構造なので $O^T=O^{H\prime}$ であり、f^H の影響は $O^{H\prime}$ に現れることから、f^T の影響は O^T に現れる。さらに、(i) で示した f^H と f^T の対応関係より、S^H において f^H が t^H でテスト可能ならば、S^T において f^T は $M^{-1}(t^H)$ でテスト可能である。

(iii) S^T の任意の故障を f^T として、f^T に対応する S^H の故障を f^H とする。f^T が S^T のテスト法でテスト可能ならば、f^H は S^H のテスト法でテスト可能であることを示す。f^T がテスト可能ならば、f^T のテスト系列 t^T が存在する。ここで、t^T の系列長を n とする。S^T に入力系列 t^T を印加して得られる出力系列を $O^T=o_1^T,o_2^T,\ldots,o_n^T$ とする。S^H のテスト法は d-時刻保持テスト法なので、f^H のテスト系列は d-時刻保持系列でなければならない。d-時刻保持系列変換を M として、S^H に $M(t^T)$ を印加して得られる出力系列を $O^H=o_1^H,o_2^H,\ldots,o_{(d+1)n}^H$ とする。また、O^H から $d+1$ 周期でベクトルを抽出した系列を $O^{H\prime}=o_{d+1}^H,o_{(d+1)2}^H,\ldots,o_{(d+1)n}^H$ とする。このとき、核回路が平衡構造なので $O^{H\prime}=O^T$ であり、f^T の影響は O^T に現れることから、f^H の影響は $O^{H\prime}$ に現れる。さらに、(i) で示した f^H と f^T の対応関係より、S^T に

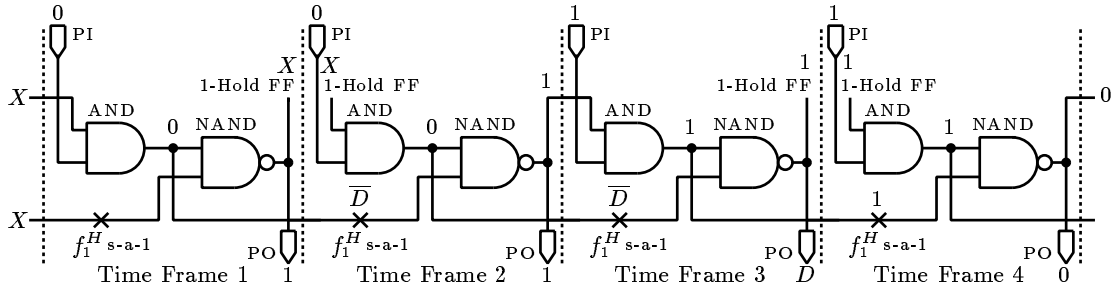

図 13: S_1^H の タイムフレーム展開

おいて f^T が t^T でテスト可能ならば、S^H において f^H は $M(t^T)$ でテスト可能である。

(定義12-(ii)) (定義12-(i))(i) より故障が1 対1 に対応し、(定義12-(i))(iii) より S^T において S^T のテスト法でテスト可能な任意の故障 f^T のテスト系列の集合 $\mathcal{T}(f^T)$ に対して、$\bigcup_{t\in\mathcal{T}(f^T)}\{M(t)\}$ は f^T に対応する S^H の故障 f^H の S^H のテスト法におけるテスト系列となっている。すなわち、$\bigcup_{t\in\mathcal{T}(f^T)}\{M(t)\}=\mathcal{T}(f^H)$ となる変換 M が存在する。ここで、$\mathcal{T}(f)$ は故障 f に対応する回路のテスト法で f をテスト可能なすべてのテスト系列の集合とする。

以上より、定理 1 は証明された。 □

[定理 **2**] S_K が無閉路構造のとき、S^H のテスト生成問題は、必ずしも S^T のテスト生成問題に帰着できない。

(証明) 変換 HT がテスタビリティを保存しない例を示す。図 10の順序回路 S_1 を考える。核回路が無閉路構造になるように FF2 を外部 FF に選び、S_1 から FF2 を取り除いた部分回路を核回路とする。核回路の順序深度は 1 である。S_1 を 1-時刻保持変換した回路を S_1^H (図 11)、S_1 を組合せ回路疑似変換した回路を S_1^T (図 12) とする。このとき、S_1 の信号線 e の 1-縮退故障 f_1 を考える。図 11 と図 12からわかるように、S_1^H と S_1^T にも f_1 に対応する故障 f_1^H と f_1^T がそれぞれ存在する。S_1^H におけるテスト法は 1-時刻保持テスト法で、f_1^H のテスト系列 t_1^H は、1-時刻保持系列 $t_1^H=\begin{bmatrix}0011\end{bmatrix}$ となり、3 時刻目で外部出力に誤りが観測できる (図 13参照)。これに対して、通常のテスト法の S_1^H において、信号線 c に 1、e に 0 を同時に設定できないので f_1^T はテスト不可能である (図 12 参照)。

以上より、定理 2は証明された。 □

順序回路 S (図 1) を核回路 S_K と外部 FF Q_E に分割したとする (図 2)。内部 FF を Q_I とする。S を組合せ回路疑似変換 T した回路を S^T とする。S、S^T のテスト法はどちらも通常のテスト法である。このとき、S と S^T の間のテスト生成問題帰着性に関して、定理 3、系 1 が成立する。

[定理 **3**] S_K が平衡構造のとき、S のテスト生成問題は、必ずしも S^T のテスト生成問題に帰着できない。

(証明) 組合せ回路疑似変換 T がテスタビリティを保存しない例を示す。

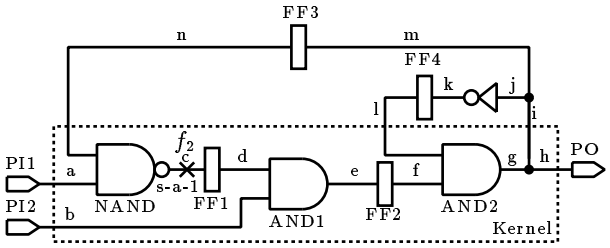

図 14: 順序回路 S_2

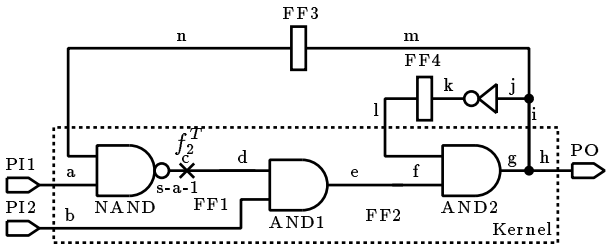

図 15: 順序回路 S_2^T

図 14の順序回路 S_2 を考える。核回路が平衡構造になるように FF3 と FF4 を外部 FF に選び、S_2 から FF3 と FF4 を取り除いた部分回路を核回路とする。S_2 を組合せ回路疑似変換した回路を S_2^T (図 15) とする。このとき、S_2 の信号線 c の 1-縮退故障 f_2 を考える。図 15からわかるように、S_2^T にも f_2 に対応する故障 f_2^T が存在する。S_2 における、f_2 のテスト系列 t_2 は、$t_2=\begin{bmatrix}0XX1XX\\0\ 1X0\ 1X\end{bmatrix}$ となる (図 16参照)。これに対して、S_2^T において FF3 に 1、FF4 に 1 を同時に設定できないので、f_2^T はテスト不可能である。(図 15参照)。

以上より、定理 3は証明された。 □

[系 **1**] S_K が無閉路構造のとき、S のテスト生成問題は、必ずしも S^T のテスト生成問題に帰着できない。

(証明) { 無閉路構造の順序回路 } $\supseteq$ { 平衡構造の順序回路 } なので、定理 3 より明らか。 □

5. 回路疑似変換によるテスト生成法

この章では、回路疑似変換によるテスト生成法を提案する。以下では、与えられた順序回路を S として、S を核回路 S_K と外部 FF Q_Eに分割したとする。また、内部 FF を Q_I、核回路の順序深度を d とする。S に対して、組合せ回路疑似変換を行った回路を S^T、d-時刻保持変換

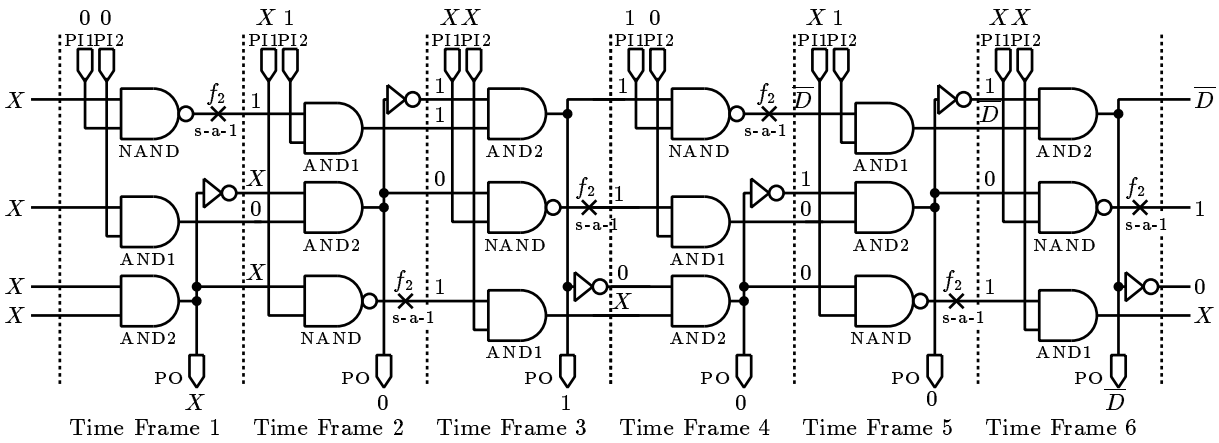

図 16: S_2 のタイムフレーム展開

を行った回路を S^H として説明する。

組合せ回路疑似変換によるテスト生成は次のように行う。

(Step 1) S (図1) を S_K と Q_E に分割 (図2) する。このとき、核回路の順序深度を d とする。

(Step 2) S を 組合せ回路疑似変換して S^T (図 4) を求める。

(Step 3) S^T に対して 順序回路テスト生成アルゴリズムを適用してテスト生成し、テスト系列 t^T を求める。

(Step 4) テスト系列 t^T を d-時刻保持系列変換して t^H を求める。

[例 **2**] (組合せ回路疑似変換によるテスト生成) 図 17の順序回路 S_3 について考える。ここで、C_1, C_2, C_3, C_4 は組合せ回路部分を表す。(Step 1) で図 17のように、S_3 を核回路 S_{3K} と外部 FF $Q_E = \{\text{FF5,FF6}\}$ に分割する。核回路の順序深度は 2 である。(Step 2) で内部 FF $Q_I = \{\text{FF1,FF2,FF3,FF4}\}$ を信号線に置き換えた回路を S_3^T とする。(Step 3) でテスト生成したテスト系列 t^T を (Step 4) で 2-時刻保持系列変換して、図 19のようにテスト系列 t^H を得る。

次に、S のテストは次のようなテスト法でテストを行う。

(Step 1) S を d-時刻保持変換して S^H (図 8) を求める。

(Step 2) 組合せ回路疑似変換によるテスト生成で得られた t^H で S^H を d-時刻保持テスト法でテストする。

[例 **3**] (テスト) 例2の回路 S_3 と得られたテスト系列 t^H について考える。(Step 1) では、S_3 を 2-時刻保持変換して図 20の S_3^H を得る。(Step 2) でこのテスト系列 t^H で S_3^H をテストする。

組合せ回路疑似変換によるテスト生成とテストの流れを図 21に示す。

ここで、核回路の回路構造について考える。3.章で述べたように、核回路が平衡構造のとき、S^H は S^T のテスト生成問題に帰着できる (定理 1)。そこで、本手法では平衡構造を核回路の回路構造に採用する。しかし、一般に S は S^T のテスト生成問題に帰着できない (定理 3)。帰着できない故障がどの程度存在するかについては、次

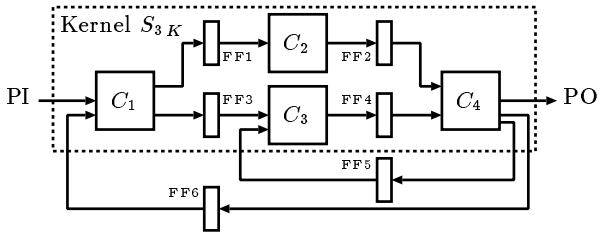

図 17: 順序回路 S_3

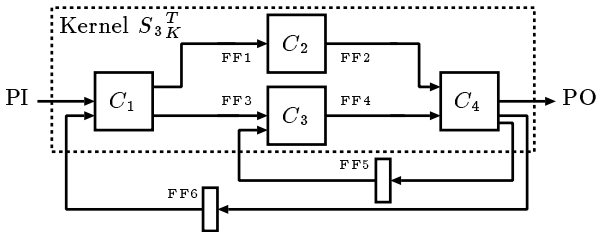

図 18: 順序回路 S_3^T

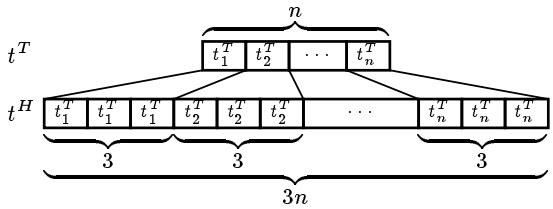

図 19: t^T の 2-時刻保持系列 t^H

章の実験によって評価する。

6. 実験結果

この章では、与えられた順序回路 S に対してテスト生成を行った結果と、S を組合せ回路疑似変換した順序回路 S^T でテスト生成を行った結果を示す。テスト生成ツールには、FASTEST[5] を用いた。使用した計算機は Sun-4/20 model 712 (Fujitsu) である。S には、ISCAS'89 ベンチマーク回路を用いた。ISCAS'89 ベンチマーク回路の回路特性を表 1に示す。circuit の欄は回路名を示し、#gate、#PI、#PO、#FF の欄はそれぞれゲート数、外部入力数、外部出力数、FF 数を表す。

組合せ回路疑似変換は、C 言語で作成したプログラム

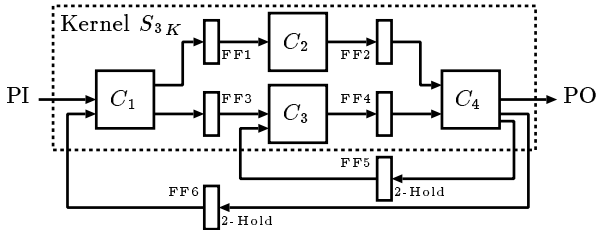

図 20: 順序回路 S_3^H

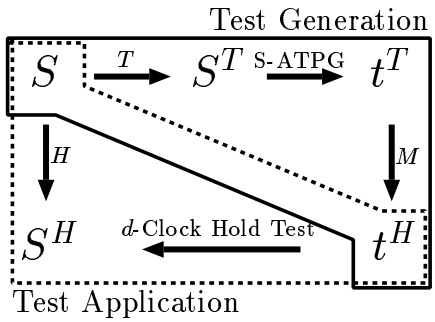

図 21: 組合せ回路疑似変換によるテスト生成とテスト

circuit	#gate	#PI	#PO	#FF
s382	99	3	6	21
s400	106	3	6	21
s444	119	3	6	21
s641	107	35	24	19
s713	139	35	23	19
s953	311	16	23	29
s1196	388	14	14	18
s1238	428	14	14	18
s1423	490	17	5	74
s5378	1004	35	49	179
s9234.1	2027	36	39	211
s9234	2027	19	22	228
s13207.1	2573	62	152	638
s13207	2573	31	121	669

表 1: 順序回路 S

によって行った。核回路の回路構造には平衡構造を用いている。組合せ回路疑似変換を行うプログラムは、次のような構成である。

(Step 1) 与えられた回路 S を、無閉路構造を有する部分回路 S_A と外部 FF Q_A に分割する。

(Step 2) S_A を平衡構造を有する部分回路 S_B と外部 FF Q_B に分割する。

(Step 3) S_B を核回路として、$Q_A \cup Q_B$ を外部 FF とし、内部 FF をバッファに置き換える。

(Step 1) では、MFVS(Minimum Feedback Vertex Set) [4] を求めることによって無閉路構造を有する部分回路を求めている。また、平衡構造は、無閉路に変換した回路に対して、次のような方法で求めた。

(Step 1) すべての外部入力とすべての疑似入力を始点として、始点の順序深度を 0 とする。始点から外部出力または疑似出力に到達するまで、(Step 2) から (Step 4) の操作を繰り返す。

circuit	#I-FF	#E-FF	d	cpu time (sec)
s382	6	15	1	**0.45**
s400	6	15	1	**0.44**
s444	6	15	1	**0.54**
s641	4	15	1	**1.71**
s713	4	15	1	**2.21**
s953	23	6	1	**2.08**
s1196	2	16	1	**2.07**
s1238	2	16	1	**2.01**
s1423	2	72	1	**4.70**
s5378	55	124	2	**44.65**
s9234.1	18	193	4	**162.00**
s9234	18	210	4	**161.50**
s13207.1	197	441	8	**429.57**
s13207	198	471	8	**418.67**

表 2: 順序回路 S^T

(Step 2) 始点から、はじめに到達する収斂点までのすべての経路の順序深度を計算する。

(Step 3) はじめに到達した収斂点において、その点に収斂するすべての経路の順序深度を比較して外部 FF を選択する。具体的には、順序深度 0 の経路が存在すれば、収斂点の直前の FF 全てを外部 FF として、その収斂点の順序深度を 0 とする。順序深度 0 の経路が存在しない場合は、最も多い順序深度をその収斂点の順序深度として、それ以外の順序深度を持つ経路は収斂点の直前の FF を外部 FF とする。

(Step 4) これらの収斂点を始点とする。

S を組合せ回路疑似変換した回路 S^T の回路特性を表 2 に示す。#I-FF、#E-FF の欄はそれぞれ内部 FF 数、外部 FF 数を表す。この#I-FF 数が信号線 (バッファ) に置き換わる FF 数で、減少する FF 数を表している。また、d の欄は核回路の順序深度を表し、cpu time (sec) の欄は組合せ回路疑似変換にかかった時間 (秒) を表している。

はじめにテスト生成した結果を示す。図 3 が FASTEST の結果である。表中の#fault、#vector、f.cov.(%)、cpu time (sec) の欄はそれぞれ全故障数、テストベクトル数、故障検出率 ([検出された故障数]/[全故障数])、実行時間 (秒) を示している。また、S、S^T の欄はそれぞれ S に対してテスト生成を行った結果、S^T に対してテスト生成を行った結果であることを示している。s382、s400、s444、s713、s1423 で故障検出率、テスト生成時間ともに S よりも S^T の方が良い結果が得られている。また、S と S^T で故障検出率が同じだった s1196、s1238、s9234.1、s9234 では S^T でテスト生成時間が短縮されていることがわかる。s13207.1、s13207 では、S^T で故障検出率は落ちているが、テスト生成時間が短縮されている。その他の回路では、大きな差は見られない。次に、この実験結果について S と S^T で検出された故障の違いを解析する。検出された故障の違いを表 4に示す。表中の$\#S_D$、$\#S_D^T$ の欄はそれぞれ S、S^T で検出された故障数を示す。$\#S_D \cap S_{UD}^T$、$\#S_{UD} \cap S_D^T$ の欄はそれぞれ S で検出され、かつ、S^T で検出されなかった故障数、S で検出されず、かつ、S^T で

circuit	#fault	S			S^T		
		#vector	f.cov.(%)	cpu time (sec)	#vector	f.cov.(%)	cpu time (sec)
s382	764	51	**55.63**	**630**	86	**82.98**	**399**
s400	800	51	**54.63**	**680**	70	**81.00**	**388**
s444	888	47	**18.92**	**1680**	87	**79.73**	**410**
s641	1278	146	**87.32**	**48**	159	**87.25**	**53**
s713	1426	137	**83.10**	**234**	141	**83.24**	**233**
s953	1906	12	**7.92**	**20**	7	**7.82**	**14**
s1196	2392	347	**99.87**	**210**	384	**99.87**	**204**
s1238	2476	368	**96.65**	**555**	334	**99.65**	**515**
s1423	2846	550	**87.16**	**7298**	477	**88.97**	**5627**
s5378	10590	856	**78.57**	**154397**	535	**77.75**	**168018**
s9234.1	18468	51	**10.02**	**236206**	49	**10.02**	**212121**
s9234	18468	4	**0.38**	**3170**	4	**0.38**	**3059**
s13207.1	26358	117	**11.64**	**226916**	57	**11.53**	**188256**
s13207	26358	161	**7.23**	**250668**	78	**6.84**	**153687**

表 3: FASTEST の結果

circuit	$\#S_D$	$\#S_D^T$	$\#S_D \cap S_{UD}^T$	$\#S_{UD} \cap S_D^T$
s382	425	634	**3**	**212**
s400	437	648	**3**	**214**
s444	168	708	**0**	**540**
s641	1116	1115	**1**	**0**
s713	1185	1187	**0**	**2**
s953	151	149	**3**	**1**
s1196	2389	2389	**0**	**0**
s1238	2393	2393	**0**	**0**
s1423	2481	2532	**14**	**65**
s5378	8821	8234	**182**	**95**
s9234.1	1851	1851	**0**	**0**
s9234	70	70	**0**	**0**
s13207.1	3069	3041	**40**	**12**
s13207	1906	1803	**103**	**0**

表 4: FASTEST で検出された故障の違い

検出された故障数を示している。

実験結果で、S^T で故障検出率、テスト生成時間ともに良くなっていた s382、s400、s444、s713、s1423 では、$\#S_{UD} \cap S_D^T$ が $\#S_D \cap S_{UD}^T$ を上回っている。特に、s444、s713 では、S で検出された故障はすべて S^T で検出され、さらに S で検出されなかった故障が S^T で検出されている。また、S と S^T で故障検出率が同じだった s1196、s1238、s9234.1、s9234 では、$\#S_D \cap S_{UD}^T$ と $\#S_{UD} \cap S_D^T$ ともに 0 であることから、完全に同じ故障が検出されていることがわかる。その他は $\#S_{UD} \cap S_D^T$ が $\#S_D \cap S_{UD}^T$ を下回っている。

検出された故障の違いを見ることによって、テスタビリティの保存性を評価することができる。S と S^T で完全に同じ故障を検出している回路に対しては、組合せ回路疑似変換はテスタビリティを保存するといえる。また、S^T において、S よりも多くの故障を検出している回路に対しては、組合せ回路疑似変換によってテスタビリティが向上したといえる。その他の回路では、S^T において、S よりも少ない故障しか検出できていないが、テスタビリティはそれほど悪くなっていない。

7. あとがき

本論文では、回路疑似変換によるテスト生成法を提案し、組合せ回路疑似変換によってテスト生成されたテスト系列でテストを行う d-時刻保持テスト法を提案した。組合せ回路疑似変換と d-時刻保持変換によるテスト生成問題帰着性を明らかにした。さらに、ベンチマーク回路を用いた実験により組合せ回路疑似変換によるテスト生成法の有効性を評価した。

謝辞

本研究に関し、多くの貴重な意見を頂いた本学の増澤利光助教授、井上美智子助手始め情報論理学講座の諸氏に感謝します。

参考文献

[1] A. Balakrishnan and S. T. Chakradhar, "Software Transformation for Sequential Test Generation," in *Proc. of IEEE 4th Asian Test Symposium*, pp.266-272, Nov. 1995.

[2] R. Gupta, R. Gupta and M. A. Breuer, "The BALLAST Methodology for Structured Partial Scan Design," *IEEE Transactions on Computers*, Vol.39, No.4, pp.538-544, April 1990.

[3] 藤原秀雄, "組合せテスト生成複雑度でテスト生成可能な順序回路とその応用," 信学技法, FTS95-65, pp.17-24, Dec. 1995.

[4] S. T. Chakradhar, A. Balakrishnan, and V. D. Agrawal, "An Exact Algorithm for Selecting Partial Scan Flip-Flops," in *Proc. of 31th ACM/IEEE Design Automation Conference*, pp.81-86, June, 1994.

[5] T. P. Kelsey, K. K. Saluja and S. Y. Lee, "An Efficient Algorithm for Sequential Circuit Test Generation," *IEEE Transactions on Computers*, Vol.42, No.11, pp.1361-1371, Nov. 1993.