

パス遅延故障を検出可能な順序回路を 構成するための状態割り当て法

佐久真 源太 島尻 寛之 吉田 たけお

琉球大学 工学部 情報工学科

E-mail: {gentas, shimajiri, tyoshida}@fts.ie.u-ryukyu.ac.jp

あらまし 本稿では、パス遅延故障のテストに用いられている2パターンテストの考えを応用した、パス遅延故障を検出可能な順序回路を構成するための状態割り当て法について述べる。パス遅延故障は順序回路のレジスタの値に影響を及ぼすため、レジスタの値を観測する必要がある。本稿では、レジスタの値を観測することによって、順序回路に発生したパス遅延故障を検出するための条件について検討する。また順序回路に対して、この条件を満たす状態遷移図や状態割り当ての方法を提案する。さらに、提案手法を用いたパス遅延故障を検出可能な順序回路の設計例を示す。ITC'99ベンチマーク回路に提案手法を適用した結果、回路の遅延時間がほとんど増加しないことを確認した。

キーワード：パス遅延故障，故障検出，配線遅延，状態割り当て

A State Assignment Method for Constructing Path Delay Faults Detectable Sequential Circuits

Genta SAKUMA, Hiroyuki SHIMAJIRI and Takeo YOSHIDA

Department of Information Engineering, Faculty of Engineering,

University of the Ryukyus

E-mail: {gentas, shimajiri, tyoshida}@fts.ie.u-ryukyu.ac.jp

Abstract In this paper, we propose a state assignment method for constructing sequential circuits which can detect path delay faults. Path delay faults affect a value of registers in sequential circuits. Therefore, we can detect path delay faults by observing a value of registers in sequential circuits. In this paper, we show detection conditions for path delay faults. We also show a method to satisfy the detection conditions and a design example of path delay faults detectable sequential circuits which are adopted the proposed method. ITC'99 benchmark circuits which are adopted the proposed method had little increase in a delay time.

Keywords : Path Delay Fault, Fault Detection, Wiring Delay, State Assignment

1 はじめに

近年、半導体微細加工技術の進歩によって、縮退故障ではモデル化できない新たな物理現象が顕在化してきた。そのような物理現象の一つに配線遅延が挙げられる[1]。この配線遅延に起因する故障のモデルは、既に多数提案されており、その中でも、近年、パス遅延故障モデルがよく扱われている。このパス遅延故障に関する研究は、テスト生成の分野では活発に行われているが、耐故障技術の分野では扱われていない。しかし、テストによって故障を完全に除去することは困難であるため、パス遅延故障に対する耐故障技術を早期に確立する必要があると考える。

故障に対する耐性を回路に持たせるためには、まず、その故障を検出できる必要がある。そこで本稿で

は、順序回路の状態遷移回路を対象とし、その動作時にパス遅延故障を検出する方法について検討する。

ところで、テスト生成の分野では、2パターンテストと呼ばれる手法を用いてパス遅延故障の検出を行っている。2パターンテストでは、2つのテストベクトルに対する出力の変化を観測し、要求した出力変化が得られなかった場合に、論理回路にパス遅延故障が発生したものと判別している[2, 3]。本稿では、この2パターンテストの考え方を用いる。

論理回路に発生したパス遅延故障を検出するためには、その回路の出力の変化を観測する必要がある。しかし、論理回路の出力の変化を観測するだけでは、その回路の出力が正しく変化したかどうかを判別できないため、時間的に連続する2つの出力値間に何ら

かとの関係を与えることを考える。本稿で対象とする状態遷移回路の出力は、順序回路の状態を表している。そのため、順序回路の状態割り当てを工夫することにより、時間的に連続する2つの出力値間に何らかの関係を与えることができ、パス遅延故障を検出できると考えられる。そこで本稿では、パス遅延故障を検出可能な順序回路を構成するための状態割り当て法について検討する。

以下2で、パス遅延故障の定義を示し、パス遅延故障の検出条件について述べる。続いて3では、パス遅延故障を検出可能な順序回路の設計方法について説明する。4で、設計例とその評価を示す。

2 パス遅延故障とその検出条件

2.1 パス遅延故障とその影響

ここでは、パス遅延故障の定義を示し、パス遅延故障が発生した場合の論理回路(順序回路)の出力が取り得る値について説明する。

定義 2.1 論理回路において、外部入力またはフリップフロップ(FF)を始点とし、外部出力またはFFを終点とする接続したゲートの集合をパスと呼ぶ。このとき、あるパスの始点における信号値の変化が、規定時間内にそのパスの終点に到達しないような故障をパス遅延故障という。□

次に、パス遅延故障の影響について検討する。そのために、以下の議論で用いる用語を定義しておく。

定義 2.2 n ビットのベクトル $X_1 = (x_{11}x_{12}\cdots x_{1n})$ と $X_2 = (x_{21}x_{22}\cdots x_{2n})$ に対して、

$$CB(X_1, X_2) = \{(c_1c_2\cdots c_n) \mid c_i \in \{x_{1i}, x_{2i}\}, \\ i = 1, 2, \dots, n\}$$

で表されるベクトル集合 $CB(X_1, X_2)$ を、 X_1 と X_2 で張られるキューブという。□

以下では、パス遅延故障が発生する可能性のある論理回路 L に対して、 L への入力および L からの出力は、どちらもレジスタを介しているものとする。このとき、パス遅延故障の影響を受けるのは、論理回路 L 内のレジスタ R が保持する値である。以下では、時刻 t において、レジスタ R が実際に保持している値およびレジスタ R が保持すべき正しい値をそれぞれ、 R^t および $[R^t]$ と表わすことにする。

ここで、 $[R^{t-1}] = (0, 0, 0)$ 、 $[R^t] = (0, 1, 1)$ となる場合について考える。まず、論理回路 L にパス遅延故障が発生していない場合は、 $R^t = (0, 1, 1)$ となる。またパス遅延故障により、すべての信号変化が遅延した場合は、 $R^t = (0, 0, 0)$ となる。さらに、1ビット目のみ遅延した場合、1ビット目と2ビット目が遅延した場合など、すべての可能性を調べると、 $R^t \in \{(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1)\}$ となる。

この例からわかるように、 n ビットのレジスタ R に対して、明らかに

$$R^t \in CB([R^{t-1}], [R^t])$$

が成り立つ。この式は、論理回路 L にパス遅延故障が発生しているか否かに関わらず成り立つ。

2.2 パス遅延故障の検出条件

次に、パス遅延故障を検出するための条件について検討する。ここで、論理回路 L にパス遅延故障が発生している場合、レジスタ R の値 R^t とパス遅延故障との間には、以下のような関係がある。

(1) $R^t = [R^t]$ の場合

パス遅延故障は、論理回路 L の動作に影響を及ぼさない。

(2) $R^t = [R^{t-1}]$ の場合

$[R^{t-1}] = [R^t]$ であれば、パス遅延故障は、論理回路 L の動作に影響を及ぼさない。一方、 $[R^{t-1}] \neq [R^t]$ であれば、パス遅延故障は、論理回路 L の動作に影響を及ぼす。

(3) 上記以外の場合

パス遅延故障は、論理回路 L の動作に影響を及ぼす。

論理回路 L が時刻 $t-1$ までに故障の影響を受けず正しく動作していた場合に、時刻 t で得られるレジスタ R に関する情報は、 $[R^{t-1}]$ と R^t のみである。これらの情報から、上記のどの場合に該当するかを判断できれば、(論理回路 L の動作に影響を及ぼす)パス遅延故障の発生の有無を判断できる。しかし、時刻 $t-1$ と時刻 t でレジスタ R が保持している値が同じ場合は、その値が、論理回路 L の機能によるものか、パス遅延故障の影響によるものかを判断できない。そのため以下では、 $[R^{t-1}] \neq [R^t]$ であると仮定する。すなわち、上記(2)に該当する場合、パス遅延故障が発生しているものと判断することにする。

ここで、 n ビットのベクトル X_1, X_2 のハミング距離を $d(X_1, X_2)$ とし、 $CB([R^{t-1}], [R^t])$ の要素の性質について考える。いま、 $d([R^{t-1}], [R^t]) = m$ とすると、 $\forall X \in CB([R^{t-1}], [R^t])$ に対して、 $d([R^{t-1}], X) \leq m$ なる関係が成り立つ。また、この式の等号は、 $X = [R^t]$ の場合のみ成り立つ。このことは、ハミング距離 $d([R^{t-1}], [R^t])$ の値が、 m となる場合は、論理回路 L にパス遅延故障は発生しておらず、 m とならない場合は、パス遅延故障が発生していることを意味する。すなわち、この関係を用いれば、論理回路 L にパス遅延故障が発生したか否かを判別できる。

ただし、この方法を用いるためには、以下の前提が必要となる。いま、 n ビットのレジスタ R に対して、 $d([R^{t-1}], [R^t]) = m$ とすると、 R への入力線のうち、

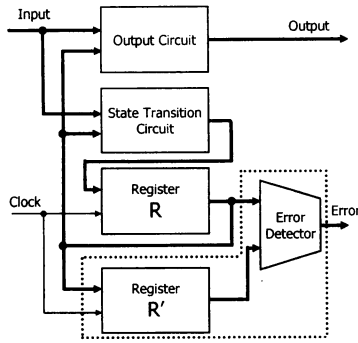


図1 パス遅延故障の検出機構を付加した順序回路

$n - m$ 本の信号線は、両時刻において値が等しくなる。これら $n - m$ 本の信号線の値が、ハザードなどの原因により変化すると、上記の方法を用いることができない。しかし、配線遅延が増加する状況下では、ハザードの影響を無視できないと考えられるため、この点についても考慮する必要がある。

以上のことを、以下に定理として示す。

定理 2.1 順序回路の状態遷移回路が以下の条件をすべて満たし、時刻 $t - 1$ まで正常に動作しているものとする。このとき、状態を保持する n ビットのレジスタ R の値が、 $d(R^{t-1}, R^t) = m$ であれば、パス遅延故障は発生しておらず、 $d(R^{t-1}, R^t) \neq m$ であれば、パス遅延故障が発生している、と判別できる [4].

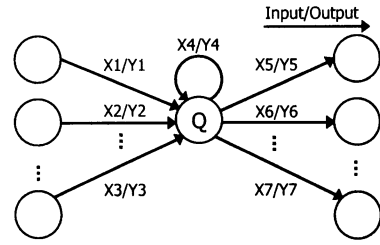
- (1) $[R^{t-1}] \neq [R^t]$
- (2) $d([R^{t-1}], [R^t])$ が既知の値 m
- (3) レジスタ R の入力線のうち、時刻 $t - 1$ と t において値が等しくなる $n - m$ 本の信号線が、ハザードの影響を受けない。 □

3 パス遅延故障を検出可能な順序回路の構成

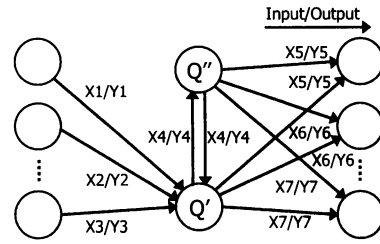
3.1 パス遅延故障検出機構

定理 2.1 を用いてパス遅延故障を検出するためには、まず、連続する2つの時刻におけるレジスタの値を比較する必要がある。そこで以下では、一般的な Mealy 型の順序回路に対して、図1に示す点線で囲まれた検出機構を付加した回路を用いることにする。また本稿では、図1において、組み合わせ回路の出力となる部分、すなわち、出力回路および誤り検出器に関しては、パス遅延故障の検出対象外とする。

図1において、レジスタ R' は、時刻 $t - 1$ におけるレジスタ R の値を保持する。また、誤り検出器は、レジスタ R, R' のそれぞれが保持している値のハミング距離が、ある規定値 m となっているか否かを判定する。図1の状態遷移回路が、**定理 2.1** を満たせば、以上で述べた検出機構を用いることにより、



(a) 自己ループが存在する状態



(b) 自己ループ除去後の状態

図2 状態の分割による自己ループの除去

パス遅延故障を検出できる。以下では、**定理 2.1** を満たす回路構成について述べる。

3.2 定理 2.1 (1) を満たす回路構成

図1の回路において、レジスタ R, R' は、順序回路の状態を保持するレジスタである。すなわち、**定理 2.1 (1)** の条件は、回路にクロックパルスが印加されるたびに、異なる状態に遷移しなければならないことを表している。状態割り当て時に各状態に対して割り当てる値はユニークであるため、この条件を満たすためには、自己ループの存在しない状態遷移図を用いて設計を行えばよい。

ここで、図2(a)に示すような、自己ループを持つ状態 Q に着目する。この自己ループを除去するには、図2(b)に示すように、元の状態を2つの状態 Q', Q'' に分割すればよい。

このように、与えられた状態遷移図の各自己ループを除去してから状態割り当てを行えば、**定理 2.1 (1)** を満たす回路を実現することができる。

3.3 定理 2.1 (2) を満たす回路構成

以下では、状態割り当ての際に各状態に割り当てる2値ベクトルを、その状態の状態値と呼ぶことにする。また、2つの状態に割り当てた状態値間のハミング距離を、単に状態間の(ハミング)距離と呼ぶ。

定理 2.1(2) を満たすためには、与えられた状態遷移図の隣接する2つの状態間に、ハミング距離が定数 m となるような状態値を割り当てる必要がある。ところで、**定理 2.1** のレジスタ R の値は、状態遷移回路の出力であると同時に、状態遷移回路の入力でも

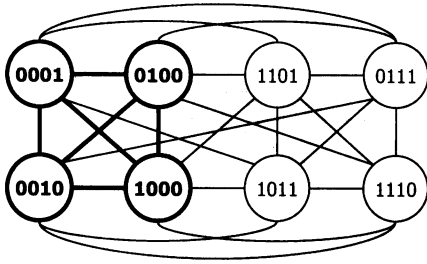


図3 4-状態値グラフ

ある。後に検討するハザードについて考慮した場合、レジスタ R の値が変化するとき、同時に変化するビット数は、少ない方が望ましい。そこで、隣接する状態間の距離を $m = 2$ とする。

次に与えられた状態遷移図に対して、任意の隣接する状態間の距離が2となるような状態割り当てについて検討する。いま、ビット長が b で(ハミング)重みが奇数である 2^{b-1} 個のベクトルをノードとし、距離が2となるノード(ベクトル)同士を無向辺で結んだ単純無向グラフを b -状態値グラフと呼ぶことにする。 b -状態値グラフでは、各ノードは ${}_bC_2$ 個のノードと隣接する。すなわち、 b -状態値グラフは、次数 ${}_bC_2$ の正則グラフとなる。また、 b -状態値グラフには、重み1のノードがちょうど b 個存在し、明らかに、それらは互いに隣接している。このため、 b -状態値グラフは、次数 b の完全グラフを部分グラフとして持つ。例として、4-状態値グラフを図3に示す。図3では、太線の部分が次数4の完全グラフとなっている。

また、状態遷移図(有向グラフ)において、各有向弧を無向辺に置き換え、さらに多重辺を除去して得られる単純無向グラフを、その状態遷移図の核グラフと呼ぶことにする。

このとき、**定理 2.1(2)** を満たす状態割り当ては、以下のように行えばよい。まず与えられた状態遷移図を、**3.2** で述べた方法に従って、自己ループを含まない状態遷移図に変換する。このときの状態数を s とする。次に、この状態遷移図の核グラフを部分グラフとして持つ b -状態値グラフを作成する。ここで、 s -状態値グラフには、次数 s の完全グラフが含まれるため、元の状態遷移図の核グラフを部分グラフとして持つ b -状態値グラフは、 $b \leq s$ で作成できる。このとき、元の状態遷移図の各状態に対応する b -状態値グラフのノードを用いて、状態割り当てを行う。これにより、任意の隣接する2状態間の距離が2となる。

3.4 定理 2.1 (3) を満たす回路構成

以下では、**定理 2.1 (3)** を満たす方法について検討する。そこでまず、ハザードについて述べる。

組み合わせ回路の各構成要素が持つ遅延のばらつ

きが原因で、その回路への入力に変化したときに、瞬間的に不正パルスを生じる可能性があるとき、その回路にはハザードが存在するという[5]。ハザードは、その存在要因によって、論理ハザードと関数ハザードに大別される。また、不正パルスの現れ方によって、静的ハザードと動的ハザードに大別される。論理ハザードは、回路構造を工夫することによって、除去できることが知られている。そこで以下では、論理ハザードは除去されているものとして、関数ハザードのみを考える。また、動的ハザードは、組み合わせ回路への入力に変化したときに、出力も変化する場合に影響が現れるハザードであるため、**定理 2.1 (3)** において考慮する必要はない。すなわち、**定理 2.1 (3)** は、状態遷移回路が静的関数ハザードの影響を受けないこと、と換言することができる。ところで、関数ハザードは、一般に除去できないことが知られている。そこで、関数ハザードに起因する不正パルスの対処について検討する。まず、組み合わせ回路に、静的関数ハザードが存在するための条件を示す。なお以下では、組み合わせ回路 C が実現している論理関数を $f_C(X)$ 、回路 C への入力が X_1 から X_2 に変化することを $X_1 \rightarrow X_2$ と表す。

定理 3.1 組み合わせ回路 C への入力変化 $X_1 \rightarrow X_2$ に対して静的関数ハザードが存在するための必要十分条件は、以下の通りである[5]。

- (1) $f_C(X_1) = f_C(X_2)$
- (2) $\exists X_3 \in CB(X_1, X_2)$ s.t. $f_C(X_1) \neq f_C(X_3)$ □

本稿で対象とする状態遷移回路は、時刻 t における順序回路への入力 X^t と、状態(レジスタ R の値) R^t から、時刻 $t+1$ の状態 R^{t+1} を求める組み合わせ回路である。すなわち、状態遷移回路が実現している論理関数、すなわち状態遷移関数 δ は、

$$R^{t+1} = \delta(X^t, R^t)$$

と表される。**定理 3.1** に示したように、静的関数ハザードは、回路への入力に変化した際に生じる。そこで、状態遷移回路への入力変化 ($[X^{t-1}, [R^{t-1}]] \rightarrow ([X^t, [R^t])$ について検討する。ここで、レジスタ R が保持する値は、**3.3** で述べたように、 b ビット長で重みが奇数のベクトルである。

いま、ハザードに起因する不正パルスが生じなければ、状態遷移回路の出力は、 $[R^t]$ から R^{t+1} に変化する。このとき、パス遅延故障の有無に関わらず、 $R^{t+1} \in CB([R^t], [R^{t+1}])$ が成り立つことは、既に述べた通りである。ところが、状態遷移回路にハザードが存在する場合は、この関係が成り立たなくなる可能性がある。まず、入力 X がハザードや故障の影響を受けないものとして、ハザードの影響とその対策

について検討する。ここで、

$$\begin{aligned} T_{X^t} &\in CB([X^{t-1}], [X^t]), \\ T_{R^t} &\in CB([R^{t-1}], [R^t]) \end{aligned}$$

と表すことにすると、状態遷移回路にハザードが存在する場合、入力変化 $([X^{t-1}], [R^{t-1}]) \rightarrow ([X^t], [R^t])$ に対して、状態遷移回路に (T_{X^t}, T_{R^t}) が、印加される可能性がある。ハザードの影響がない場合は、 $T_{R^t} = [R^t]$ となるので、 $\delta(T_{X^t}, T_{R^t}) = [R^{t+1}]$ の計算が行われる。しかし、ハザードの影響を受ける場合は、 $T_{R^t} \neq [R^t]$ となるため、この場合は、 $R^{t+1} \in CB([R^t], [R^{t+1}])$ が成り立たなくなる可能性がある。ところが幸い、 $T_{R^t} \neq [R^{t-1}]$ かつ $T_{R^t} \neq [R^t]$ である T_{R^t} は、重みが偶数であり、状態割り当てに用いられていない。なぜならば、重みが奇数であるベクトルから距離 2 となるベクトルの重みもまた奇数となるためである。 $[R^{t-1}]$ も $[R^t]$ も、重みが奇数であるので、これら以外の $CB([R^{t-1}], [R^t])$ の要素の重みは全て偶数となる。このことから、 $T_{R^t} \neq [R^t]$ となる場合の関数値を

$$\delta([X^t], T_{R^t}) = T_{R^t}$$

と定義することにより、図 1 に示した検出機構によって、異常を検出できる。このとき、 $T_{R^t} = [R^{t-1}]$ となっても、同様に、異常を検出できる。

次に、入力 X もハザードの影響を受ける場合について考える。入力 X が誤った値に変化した場合も、 $T_{R^t} \neq [R^t]$ となる場合は、上記の通り、異常を検出できる。しかし、入力 X だけが誤った値に変化し、レジスタの値は正常に $T_{R^t} = [R^t]$ となる場合は、誤った状態に遷移してしまう可能性がある。そのため、入力 X は、何らかの誤り検出符号に符号化されている必要がある。ここで、異なる 2 つの入力符号語を X_1, X_2 と表す。これまでの議論から、 $CB(X_1, X_2)$ の要素のうち、 X_1, X_2 以外の要素に符号語が含まれなければ、 $T_{X^t} \neq [X^t]$ かつ $T_{R^t} = [R^t]$ となる場合の関数値を

$$\delta(T_{X^t}, [R^t]) = [R^t]$$

と定義することにより、図 1 に示した検出機構によって、異常を検出できる。先の条件を満たす最も単純な符号は、等距離符号である。以下では、入力 X は、等距離符号に符号化されているものとする。

以上のことから、状態遷移関数 δ の関数値を

$$\delta(T_{X^t}, T_{R^t}) = \begin{cases} [R^{t+1}], & T_{X^t} = [X^t] \wedge T_{R^t} = [R^t] \\ T_{R^t}, & T_{R^t} \neq [R^t] \\ [R^t], & T_{X^t} \neq [X^t] \wedge T_{R^t} = [R^t] \end{cases}$$

と定義することにより、図 1 に示した検出機構によって、異常を検出できることがわかる。

4 ベンチマークによる構成例と評価

ここでは、提案手法を用いた順序回路の設計例および提案手法の評価を示す。まず、ITC'99 ベンチマーク回路 b05 に対して提案手法を適用した例を、図 4 に示す。

図 4(a) は、ITC'99 ベンチマーク回路 b05 の状態遷移図である。この状態遷移図を、3.2 で述べた方法に従って、自己ループを含まない状態遷移図に変換すると、図 4(b) に示す状態遷移図が得られる。次に、図 4(b) の状態遷移図の核グラフを部分グラフとして持つ δ -状態値グラフを探す。その結果、図 4(c) に示す 4-状態値グラフの太線の部分が、図 4(b) の状態遷移図の核グラフと同型となっていることがわかる。図 4(b) の状態遷移図に対して、図 4(c) の対応するノードのベクトルを割り当てると、図 4(d) の状態遷移図が得られる。これにより、任意の隣接する 2 状態間の距離が 2 となる。さらに、状態遷移関数を 3.4 で述べたように定義すれば、図 1 に示した検出機構によって、パス遅延故障あるいはハザードに起因する不正パルスの発生を検出できる。

次に、ITC'99 ベンチマークの各状態遷移回路を、提案手法を適用して、VHDL で記述し、Synopsys 社の Design Compiler (Ver.2003.06) を用いて論理合成を行った場合のそれぞれの回路面積と遅延時間を表 1 に示す。なお、表 1 において、ベンチマーク回路 b12 および b14 は、状態遷移回路、出力回路、レジスタの他に、カウンタ等の別の回路が含まれているため、今回は、適用不能として扱っている。

表 1 からわかるように、回路面積は平均で約 24% 増加、遅延時間は平均で約 1% 減少することがわかった。このことから、提案手法を適用しても、遅延時間がほとんど変化しないため、提案手法の適用によるパス遅延故障発生率の増加はないと考えられる。

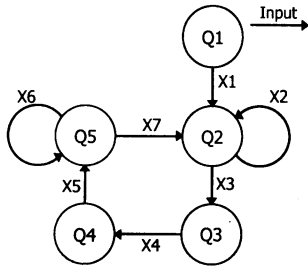
5 おわりに

本稿では、順序回路に発生したパス遅延故障を検出するための条件を明らかにした。また、この条件を満たす状態割り当て法を提案した。さらに、提案手法を用いて、ITC'99 ベンチマーク回路を設計し、その有効性について述べた。その結果、回路の遅延時間をほとんど変化させることなく、パス遅延故障を検出可能な順序回路を構成できることがわかった。

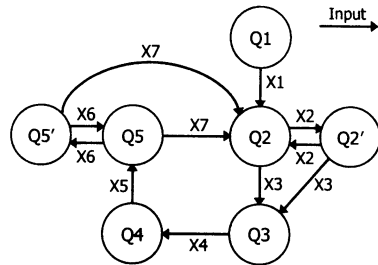
今後の課題として、状態遷移図の核グラフを部分グラフとして持つ δ -状態値グラフの効率的な探索法の検討や順序回路の出力に対するパス遅延故障の検出方法の検討などが挙げられる。

謝辞

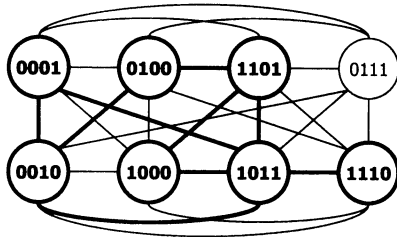
本研究は一部、文部科学省科学研究費補助金、若手研究 (B) 「遅延故障検出手法の開発と有効性の評価」



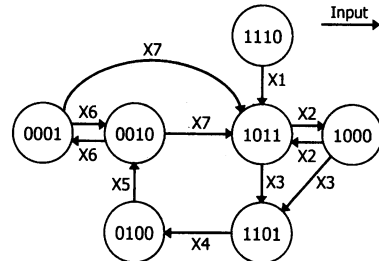
(a) ITC'99 ベンチマーク回路 b05 の状態遷移図



(b) 自己ループ除去後の状態遷移図



(c) 4-状態値グラフ内の部分グラフ



(d) 状態割り当て後の状態遷移図

図 4 ITC'99 ベンチマーク回路 b05 を提案手法を用いて設計した例

表 1 提案手法を用いて設計した ITC'99 ベンチマーク回路の回路面積と遅延時間

回路番号	回路面積 (cell)		クリティカルパス (μs)		回路増加率 (%)	遅延増加率 (%)
	適用前	適用後	適用前	適用後		
b01	88	175	1.39	1.39	199	100
b02	60	106	1.39	1.59	177	114
b03	388	408	1.39	1.39	105	100
b04	1039	1059	1.47	1.47	102	100
b05	937	977	34.24	33.80	104	99
b06	135	199	1.45	1.39	147	96
b07	798	890	1.39	1.39	112	100
b08	341	361	1.39	1.39	106	100
b09	383	448	1.45	1.39	117	96
b10	319	354	1.59	1.39	111	87
b11	753	801	1.39	1.39	106	100
b12	2034	—	1.54	—	—	—
b13	748	965	1.45	1.45	129	100
b14	9921	—	1.54	—	—	—
b15	12692	11743	1.54	1.54	93	100

(課題番号:16700073) による。

参考文献

- [1] G.L.Smith, "Model for Delay Faults Based upon Paths," ITC-85, pp342-349, 1985.
- [2] Virendra Singh, Michiko Inoue, Kewal K. Saluja and Hideo Fujiwara, "Instruction-Based Delay Fault Self-Testing of Processor Cores," Proc. International Conference on VLSI Design 2004, pp. 933-938, Jan. 2004
- [3] Yun Shao, Sudhakar M. Reddy, Seiji Kajihara, Irith Pomeranz, "An Efficient Method to Identify Untestable Path Delay Faults," ATS'01, pp233-238, Nov. 2001.
- [4] 佐久真 源太, 島尻 寛之, 吉田 たけお, "パス遅延故障を検出可能な順序回路の構成法," VLD-03, pp342-349, 2003.
- [5] 当麻 喜弘, 内藤 祥雄, 南谷 崇, "岩波講座 情報工学 13 順序回路", 岩波講座, 1993.