

パス遅延故障を検出するための等距離遷移回路の提案

趙 顯秀 吉田 たけお

琉球大学 工学部 情報工学科

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

あらまし：本稿では、パス遅延故障を検出するための等距離遷移回路を提案する。パス遅延故障は、同期式順序回路中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、同期式順序回路中の組み合わせ回路の入力と出力を符号化し、レジスタの値が符号語か否かを観測することにより、パス遅延故障を検出する方法について述べる。また、そのために符号が満たすべき条件を示し、この条件を満たす組み合わせ回路として等距離遷移回路を提案する。さらに、この等距離遷移回路が、パス遅延故障に対してフォールトセキュアとなることを示す。最後に、ITC'99 ベンチマーク回路を等距離遷移回路として設計し、等距離遷移回路の評価を行う。

キーワード：パス遅延故障，故障検出，配線遅延，等距離遷移回路，等距離符号

A Equidistant Transition Circuit for Detecting Path Delay Faults

Hyonsu CHO and Takeo YOSHIDA

Department of Information Engineering, Faculty of Engineering, University of the Ryukyus

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

Abstract : In this paper, we propose an equidistant transition circuit for detecting path delay faults. A value of each register in sequential circuits is necessary to be checked because of path delay faults affect it. In this paper, we describe a method for detecting path delay faults by observing whether a value of register is a codeword or a non-codeword, after both the input and the output of combinational circuits in sequential circuits are encoded. In addition, we show that a proposed equidistant transition circuit is fault secure for path delay faults. We also show that the evaluation of ITC'99 benchmark circuits which are adopted the proposed method.

Keywords : Path Delay Fault, Fault Detection, Wiring Delay, Equidistant Transition Circuit, Equidistant Code

1 はじめに

近年、半導体微細加工技術の進歩により、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。

パス遅延故障とは、パスの始点における信号値の変化が、規定時間内に、パスの終点に到達しないような故障のことである。ここでパスとは、外部入力またはフリップフロップを始点とし、外部出力またはフリップフロップを終点とする接続したゲートの集合のことをいう。

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 [2, 3]。しかし、耐故障技術の分野ではあまり扱われていない。また、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確立する必要があると考える。

我々の研究グループでは、順序回路の状態遷移回路を対象に、それに発生したパス遅延故障を検出する方法を既に提案している [4, 5]。そこで本稿では、同期式順序回路内の任意の組み合わせ回路を対象に、それに発生したパス遅延故障を検出する条件について検討する。また、そのパス遅延故障の検出条件を満たす方法として、等距離遷移回路を提案する。

以下 2 で、パス遅延故障の影響とその検出条件について述べる。続いて 3 では、パス遅延故障の検出条件を満たす検出方法について検討し、その検出方法を実現する一つの例として等距離遷移回路を提案する。4 で、提案した等距離遷移回路の評価を示す。

2 パス遅延故障が及ぼす影響と検出条件

2.1 パス遅延故障が及ぼす影響

ここでは、同期式順序回路に発生したパス遅延故障が、その回路に及ぼす影響について検討する。まず、以下の

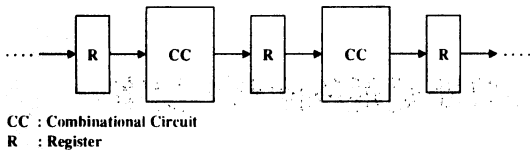


図1 同期式順序回路の構成

議論で用いる用語を定義しておく。

定義 2.1 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。□

定義 2.2 ベクトル X_1 と X_2 で張られるキューブ $CB(X_1, X_2)$ から X_1, X_2 を除いたベクトル集合を $TR(X_1, X_2)$ と表し、 X_1 と X_2 の間の過渡ベクトル集合という。すなわち、

$$TR(X_1, X_2) = CB(X_1, X_2) - \{X_1, X_2\}$$

とする。また、 $TR(X_1, X_2)$ の任意の要素を、 X_1 と X_2 の間の過渡ベクトルという。□

同期式順序回路は、図1に示すように、レジスタと組み合わせ回路が交互に接続された構造になる。以下では、同期式順序回路中の組み合わせ回路の一つを C と表す。また、回路 C の入力および出力に接続されているレジスタを、それぞれ I_C, O_C と表す。さらに、同期式順序回路中のレジスタ R が時刻 t において実際に保持している値を R^t 、レジスタ R が時刻 t において保持すべき正しい値を $\lceil R^t \rceil$ と表すことにする。ここで、同期式順序回路が正常に動作している場合は、 $R^t = \lceil R^t \rceil$ となる。

いま、組み合わせ回路 C に対して、 $[O_C^t] = (000)$ 、 $[O_C^{t+1}] = (011)$ となるものとする。このとき、回路 C にバス遅延故障が発生していない場合は、 $O_C^{t+1} = [O_C^{t+1}] = (011)$ となる。また、バス遅延故障により、すべての信号変化が遅延した場合は、 $O_C^{t+1} = [O_C^t] = (000)$ 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、 $O_C^{t+1} \in \{(000), (001), (010), (011)\}$ となる。この例からわかるように、同期式順序回路が、時刻 t まで正常に動作していた場合、 n ビットのレジスタ O_C に対して、明らかに

$$O_C^{t+1} \in CB([O_C^t], [O_C^{t+1}]) \quad (1)$$

が成り立つ。ここで、時刻 $t+1$ においてレジスタ O_C が保持する値 O_C^{t+1} とバス遅延故障との関係は以下のようになる。

(P1) $O_C^{t+1} = [O_C^{t+1}]$ の場合

バス遅延故障は、回路 C の動作に影響を及ぼさない。

(P2) $O_C^{t+1} = [O_C^t]$ の場合

$[O_C^{t+1}] = [O_C^t]$ であれば、バス遅延故障は、回路 C の動作に影響を及ぼさない。一方、 $[O_C^{t+1}] \neq [O_C^t]$ であれば、バス遅延故障は、回路 C の動作に影響を及ぼす。

(P3) $O_C^{t+1} \in TR([O_C^t], [O_C^{t+1}])$ の場合

バス遅延故障は、回路 C の動作に影響を及ぼす。

2.2 ハザードが及ぼす影響

配線遅延が支配的となる状況下では、ハザードの影響を無視できないと考えられる。そこで以下では、同期式順序回路にハザードが存在する場合の影響について検討する。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[6]。なお、このような不正パルスが生じることを、ハザードが顕在化するということにする。

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ため、以下では、関数ハザードのみについて考える。さらに動的ハザードは、回路への入力に変化したときに、出力も変化する場合に顕在化するハザードであるため、その論理的な影響は、バス遅延故障が生じた場合の影響と等価であると考えられる。そこで以下では、静的関数ハザードの影響についてのみ検討する。

まず、組み合わせ回路 C 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、回路 C が実現している論理関数を f_C 、回路への入力が X_1 から X_2 に変化することを $X_1 \rightarrow X_2$ と表す。

定理 2.1 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2つが成り立つことである[6]。

$$(1) f_C(X_1) = f_C(X_2)$$

$$(2) \exists X_3 \in CB(X_1, X_2) \text{ s.t. } f_C(X_1) \neq f_C(X_3) \quad \square$$

定理 2.1 より、回路 C に静的関数ハザードが存在する場合、入力変化 $X_1 \rightarrow X_2$ において、一時的に X_3 が印加された状態にならなくても、 $f_C(X_3)$ の値が出力される可能性があることがわかる。ここで、2.1 で説明した記号を用いると、同期式順序回路内の組み合わせ回路 C が実現している論理関数 f_C は、

$$[O_C^{t+1}] = f_C([I_C^t]) \quad (2)$$

と表すことができる。また、定理 2.1 に示したように、静的関数ハザードの影響は、回路 C への入力に変化した際に現れる。そこで、回路 C への入力変化 $[I_C^{t-1}] \rightarrow [I_C^t]$ について考える。ここで、

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

と表すと、回路 C にバス遅延故障が発生していない場合は、

$$O_C^{t+1} = f_C(V)$$

となる。ここで、時刻 t においてレジスタ I_C が保持する値 I_C^t と V との関係は、以下ようになる。

(H1) $V = [I_C^t]$ の場合

ハザードは、回路 C の動作に影響を及ぼさない。

(H2) $V \neq [I_C^t]$ の場合

ハザードは、回路 C の動作に影響を及ぼす。

2.3 バス遅延故障の検出条件

一般に、論理回路に発生した故障を検出するためには、その回路の出力に何らかの冗長性を持たせる必要がある。本稿では、同期式順序回路内のハザードが存在する組み合わせ回路 C の出力を何らかの符号 W で符号化し、回路 C の出力が符号 W の符号語であるか否かによって、バス遅延故障の有無を判別することにする。このとき、回路 C は、ハザードが顕在化しているか否かに関わらず、バス遅延故障に対してフォールトセキュアでなければならない。以下では、回路 C をフォールトセキュアとするために、符号 W が満たすべき条件について検討する。なお一般に、回路 C の出力は、レジスタに保持された後、次段の組み合わせ回路への入力となるため、回路 C の入力も同じ符号 W で符号化するものとして、以下の議論を進める。

2.1 および 2.2 で述べたように、バス遅延故障およびハザードの影響は、それぞれ、3通りと2通りの場合に分けて考えることができるため、それらの組み合わせによって、表 1 に示すように、合計6通りの場合について考える必要がある。それらのうち、2.1 の (P1) となる場合については、バス遅延故障およびハザードの顕在化の有無に関わらず、回路 C の出力が正しい符号語であると

表 1. バス遅延故障とハザードによる影響

ハザード の影響	バス遅延故障の影響		
	(P1)	(P2)	(P3)
(H1)	正	誤	誤
(H2)	正	誤	誤

判断できる。すなわち、これ以外の4通りの場合について、回路 C の出力が誤っていると判断できる必要がある。

まず、ハザードが顕在化していない状態、すなわち (H1) の場合について考える。このとき、(P2) の $O_C^{t+1} = [O_C^t]$ となる場合、 O_C^{t+1} は符号 W の符号語となるが、この場合、 O_C^{t+1} が正しい符号語か不正な符号語かを判別できない。これらを区別するためには、 $[O_C^{t+1}] (= f_C([I_C^t])) \neq [O_C^t] (= f_C([I_C^{t-1}]))$ とする必要がある。すなわち、回路 C をフォールトセキュアとするためには、時刻 t ($t \geq 0$) における W の符号語と時刻 $t+1$ における W の符号語が異なる必要がある。これを条件 1 とする。以下では、時刻 t における W の符号語の集合を W^t と表すことにする。この表現を用いると、

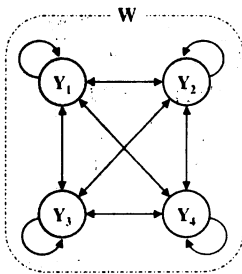
$$W = \bigcup_{t=0}^{\infty} W^t \quad (3)$$

となる。なお、条件 1 が満たされているときに、 $O_C^{t+1} = [O_C^t] (\in W^t)$ となった場合は、 O_C^{t+1} を非符号語とみなし、バス遅延故障が発生したと判断すればよい。

また、(P3) の $O_C^{t+1} \in TR([O_C^t], [O_C^{t+1}])$ となる場合、 $TR([O_C^t], [O_C^{t+1}])$ に W^{t+1} の符号語が含まれていないければ、バス遅延故障を検出できる。すなわち、 W^t の符号語と W^{t+1} の符号語との間の過渡ベクトル集合に、 W^{t+1} の符号語が含まれていない必要がある。これを条件 2 とする。

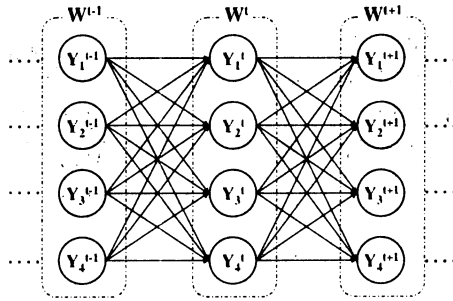
次に、ハザードが顕在化している状態、すなわち (H2) の場合について考える。この場合、2.1 で示した式 (1) が成り立たなくなる可能性がある。いま、上記 2 つの必要条件を満たしている場合、回路 C でハザードが顕在化することは、回路 C に非符号語が印加されることと等価である。この場合、例えば、回路 C がコードディスジョイントであれば、顕在化しているハザードを検出できる。

ただし、ハザードが顕在化した状態で、回路 C にさらにバス遅延故障も発生する、すなわち、(P2) または (P3) となる可能性がある。このような場合でも、回路 C は非符号語を出力しなければならない。そのためには、ハザードは顕在化しているがバス遅延故障は発生していない状態での回路 C の出力 (非符号語) と前時刻の出力の間の過渡ベクトル集合に、現時刻での符号語が存在しなければよい。これを条件 3 とする。なお、条件 3 が満たされ



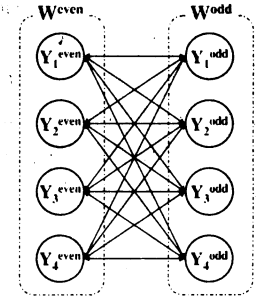
メッセージ	符号語
M_1	Y_1
M_2	Y_2
M_3	Y_3
M_4	Y_4

(a) 通常の符号語を用いた場合



メッセージ	符号語
M_1	$\dots, Y_1^{t-1}, Y_1^t, Y_1^{t+1}, \dots$
M_2	$\dots, Y_2^{t-1}, Y_2^t, Y_2^{t+1}, \dots$
M_3	$\dots, Y_3^{t-1}, Y_3^t, Y_3^{t+1}, \dots$
M_4	$\dots, Y_4^{t-1}, Y_4^t, Y_4^{t+1}, \dots$

(b) 時刻毎に符号語が異なる場合



メッセージ	符号語
M_1	Y_1^{even}, Y_1^{odd}
M_2	Y_2^{even}, Y_2^{odd}
M_3	Y_3^{even}, Y_3^{odd}
M_4	Y_4^{even}, Y_4^{odd}

(c) 偶数時刻と奇数時刻で符号語が異なる場合

図2 符号語数4 (メッセージ数4) の遷移グラフ

る場合、明らかに、回路 C はコードディスジョイントになる。

以上の各条件を定理としてまとめておく。

定理 2.2 組み合わせ回路 C の入出力が、符号 W で符号化されているものとする。このとき、符号 W が以下の3つの条件を満足すれば、回路 C は、バス遅延故障に対してフォールトセキュアとなる。ただし、 $t > 0$ とする。

- (1) $W^{t-1} \cap W^t = \phi$
- (2) $\forall X_1 \in W^{t-1}, \forall X_2 \in W^t,$
 $TR(X_1, X_2) \cap W^t = \phi$
- (3) $\forall X_1 \in W^{t-1}, \forall X_2 \in W^t, \forall V \in TR(X_1, X_2),$
 $TR(fc(X_1), fc(V)) \cap W^t = \phi$ □

3 等距離遷移回路

3.1 バス遅延故障を検出可能な符号

入出力が符号 W で符号化された組み合わせ回路 C に発生したバス遅延故障を検出するためには、符号 W が定理 2.2 を満たせばよい。そこで以下では、定理 2.2 の各条件を満たす符号について検討する。

3.1.1 定理 2.2 (1) を満たす符号

まず、定理 2.2 (1) の条件を満たす符号について考える。いま、出力が符号化された組み合わせ回路 C において、回路 C から出力される可能性のある符号語をノードとし、符号語 X_1 の次に符号語 X_2 が出力される可能性のあることを、ノード X_1 からノード X_2 への有向弧で

表した有向グラフを考える。以下では、このような有向グラフを遷移グラフと呼ぶことにする。

一般に、回路 C では、各符号語が任意の順番で出力されるため、任意の二つのノード間に有向弧が存在するような遷移グラフになると考えられる。例えば、符号語数が4の場合、図 2 (a) のような遷移グラフになると考えられる。なお、図 2 の各遷移グラフの下にある表は、元のメッセージ $M_i, i = 1, 2, 3, 4$ と符号語 Y_i との対応表である。

ここで、定理 2.2 (1) の条件は、遷移グラフに自己ループが存在しないことを意味している。この条件を満たすには、例えば、1つのメッセージに対して、時刻毎に異なる符号語を割り当てればよい。このような割り当てに基づいて、図 2 (a) の自己ループを展開すると、図 2 (b) のようになる。

ここでさらに、定理 2.2 (1) の条件を満たすには、連続する時刻において、符号語が異なりさえすればよいので、例えば、偶数時刻での符号語と奇数時刻での符号語が異なるだけでもよい。この考えに基づいて遷移グラフを構成すると、図 2 (c) のようになる。

以上のことから、定理 2.2 の条件を満たすには、定理 2.2 (2), (3) を満たす符号 W を二つに分割し、一方を偶数時刻での符号語、他方を奇数時刻での符号語として扱えばよいことがわかる。

3.1.2 定理 2.2 (2) を満たす符号

次に、定理 2.2 (2) の条件を満たす符号について考える。ここで、3.1.1 で述べたように、符号 W を二つに分

割する方法を採用する場合、定理 2.2 (2) の条件は、

$$\forall X_1 \in W^{t-1}, \forall X_2 \in W^t, \\ TR(X_1, X_2) \cap W = \phi$$

と等価になる。この条件は、偶数時刻の符号語と奇数時刻の符号語の間の過渡ベクトル集合に、 W の符号語が含まれないことを意味している。

いま、ベクトル X_1 と X_2 の (ハミング) 距離を $d(X_1, X_2)$ 、 X_1 と X_2 の間の任意の過渡ベクトルを V と表すと、

$$d(X_1, V) < d(X_1, X_2), \\ d(X_2, V) < d(X_1, X_2)$$

が成り立つ。ここで、符号 W の任意の異なる符号語間の最小距離を d_{\min} 、最大距離を d_{\max} と表した場合、

$$2d_{\min} > d_{\max} \quad (4)$$

が成り立てば、 V は W の非符号語となるので、定理 2.2 (2) の条件を満足することがわかる。

この条件を満たす最も単純な符号は、等距離符号であると考えられる。等距離符号では、 $d_{\min} = d_{\max}$ が成り立つので、明らかに、定理 2.2 (2) の条件を満足する。そこで本稿では、符号 W として、等距離符号を採用することにする。

3.1.3 定理 2.2 (3) を満たす符号

最後に、定理 2.2 (3) の条件を満たす符号について考える。ここで、3.1.1 で述べたように、符号 W を二つに分割する方法を採用する場合、定理 2.2 (3) の条件は、

$$\forall X_1 \in W^{t-1}, \forall X_2 \in W^t, \forall V \in TR(X_1, X_2), \\ TR(f_C(X_1), f_C(V)) \cap W = \phi$$

と等価になる。さらに、3.1.2 で述べたように、符号 W として等距離符号を採用した場合、上記の $f_C(X_1)$ は、 W^t の符号語となり、 V は、 W の非符号語となる。

ここで、符号 W の任意の符号語と $f_C(V)$ との間の距離の最小値を d'_{\min} 、最大値を d'_{\max} と表した場合、

$$2d'_{\min} > d'_{\max} \quad (5)$$

が成り立てば、 $TR(f_C(X_1), f_C(V))$ には、 W の符号語が含まれないので、定理 2.2 (3) の条件を満足することがわかる。

この条件を満たすには、いくつかの方法が考えられる。例えば、 W のどの符号語からの距離も等しいような非符号語 S が存在する場合、 $f_C(V) = S$ となるように、論

表 2 入出力の各桁に割り当てるベクトルの例

元の値	偶数時刻		奇数時刻	
	入力	出力	入力	出力
0	0001	0010	0010	0001
1	0100	1000	1000	0100
-	others	0000	others	0000

理関数の割り当てを行えばよい。また、 W として等距離符号を採用する場合、その符号語の 1 つ S' を取り除き、 $f_C(V) = S'$ となるように、割り当てを行ってもよい。この場合、 S' を除いた残りの符号語で、偶数時刻の符号語と奇数時刻の符号語を構成することになる。

3.2 等距離遷移回路とその設計方法

本稿で提案する等距離遷移回路とは、3.1 で述べた方法によって、定理 2.2 を満たすように設計された組み合わせ回路のことをいう。以下では、その実現方法の一例を示す。

等距離符号には、いくつかの符号が存在するので、等距離遷移回路を設計するためには、まず、使用する等距離符号を選択する必要がある。本稿では、その一例として、重みが 1 のベクトルを符号語とする重み一定で最小距離が 2 の等距離符号を採用する。なお以下では、このような重み一定の等距離符号を one-hot 符号と呼ぶことにする。

one-hot 符号では、全零のベクトルは、非符号語であり、かつ、任意の符号語からの距離が 1 となる。すなわち、回路 C が実現する論理関数 f_C に対して、非符号語が印加されたときの出力が、全零のベクトルとなるように割り当てれば、定理 2.2 (3) の条件が満たされる。

ただし、 n 入力 m 出力の回路 C の入出力を one-hot 符号で符号化すると、 2^n 入力 2^m 出力になってしまう。また、等距離遷移回路を実現するためには、さらに、one-hot 符号を偶数時刻の符号語と奇数時刻の符号語に分割する必要がある。この場合、 2^{n+1} 入力 2^{m+1} 出力になってしまう。このように、one-hot 符号をそのまま用いると、入出力線数が、劇的に増加してしまう。

そこで、元の入出力の各桁毎に短い one-hot 符号で符号化することによって、この問題を回避する。これについても、いくつかの方法が考えられるが、本稿では、表 2 に示す割り当てを採用する。

表 2 では、組み合わせ回路の各桁に対して、時刻毎に異なる 4 ビットの符号化を行っている。この場合、元の回路が n 入力 m 出力であれば、符号化後の回路は、 $4n$ 入力 $4m$ 出力の等距離遷移回路となる。また、表 2 からわかるように、この等距離遷移回路は、ハザードが存在する状況下でも、パス遅延故障に対してフォールトセ

表 3 等距離遷移回路として設計した 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	258	1104	0.21	0.24	323	14
b02	151	490	0.17	0.21	223	24
b03	879	3573	0.24	0.29	306	21
b04	2842	24387	0.44	0.82	758	86
b05	—	—	—	—	—	—
b06	336	1881	0.17	0.31	460	82
b07	2392	15321	0.39	0.60	541	54
b08	819	4506	0.27	0.31	450	15
b09	995	6444	0.22	0.36	548	64
b10	956	4640	0.22	0.31	385	41
b11	2167	26699	0.45	0.57	1132	27
b12	—	—	—	—	—	—
b13	1803	13131	0.22	0.38	628	73

キュアとなっている。

4 評価

ここでは、提案した等距離遷移回路の評価を行う。ITC'99 ベンチマーク回路を等距離遷移回路として設計し、遅延時間と回路面積について比較した結果を、表 3 に示す。なお論理合成には、Synopsys 社の Design Compiler Ver.2003.06、TSMC 社の 90nm Standard Cell Library を使用した。表 3 からわかるように、回路面積は平均で約 524%増加、遅延時間は平均で約 46%増加した。なお、ベンチマーク回路 {b05,b12} については、適用しなかった。

回路面積と遅延時間の両方に平均値以上の増加が見られるのは、回路 {b04,b07,b09,b13} である。このうち、回路 {b04,b07,b13} では、多数の算術演算回路が含まれていた。ただし、算術演算回路が内部に含まれているものの、遅延時間の増加があまり見られなかった回路 {b11} に関しては、比較演算回路が多数含まれていた。また、回路 {b09} に関しては、入出力を用いて回路内部変数の値を決定しており、この内部変数の処理するビット数が入出力ビット数に対して増加していた。なお、遅延時間の増加率が低かったものの、回路面積の増加率が高かった回路 {b08} には、ROM が含まれていた。また、その他のベンチマーク回路については、適用前の回路が小さく、処理も少ないため、回路面積と遅延時間の増加があまり見られなかった。

本稿で提案した等距離遷移回路は、同期式順序回路内のすべての組み合わせ回路に対して適用できる。しかし一方で、すべての値が出力されることを想定しているため、等距離遷移回路を適用すると、さらに冗長な構成になると考えられる。

5 おわりに

本稿では、同期式順序回路の組み合わせ回路に発生したバス遅延故障を検出するための条件を明らかにした。また、この条件を満たす方法の一つとして、等距離遷移回路の構成法を提案した。さらに、提案した等距離遷移回路を用いて、ITC'99 ベンチマーク回路を設計し、その評価結果を示した。

今後の課題としては、評価結果において回路面積と遅延時間の増加率が高かった算術演算回路などの演算回路に適用する場合の新たな手法の検討や等距離遷移回路の自動設計手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 (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] Kwang-Ting Cheng and Vishwani D. AGRAWAL, "A Partial Scan Method for Sequential Circuits with Feedback," IEEE Trans on Computers, VOL-39, NO.4, April, 1990.
- [3] Yun Shao, Sudhakar M. Reddy, Seiji Kajihara, Irith Pomeranz, "An Efficient Method to Identify Unstable Path Delay Faults," ATS'01, pp233-238, Nov. 2001.
- [4] 佐久真 源太, 島尻 寛之, 吉田 たけお, "バ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] 佐久真 源太, 島尻 寛之, 吉田 たけお, "バ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," VLD2004-76, pp91-96, 2004.
- [6] 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.