

バス回路を含む順序回路の  
テストパターン自動生成について

佐藤 敬 山下孝男 伊申泰宣 植村昌俊

(沖電気工業株式会社)

近年、デジタルPCBの高機能化、高密度化に伴ってトライステート素子や双方向バスを有するバス回路が増大し、機能試験用テストパターン作成は、増々難しくなっている。

今回、バス回路も取り扱い可能とするため、村上の9値法にバス回路のモデル化及び論理値として新たにハイインピーダンス状態を導入することにより、バス回路を含む順序回路用テストパターン自動生成プログラムを開発し、その有効性を確認した。

本報告では、バス回路を中心としたテストパターン自動生成の手法とその実行結果について報告する。

Test Generation  
for Sequential Circuits  
with Buses and Tri-state Drivers  
(in Japanese)

by Takashi SATOH, Takao YAMASHITA, Yasunori IGUSHI and Masatoshi UEMURA  
(Computer Aided Design Department, Oki Electric Industry, Ltd., 4-10-3 Shibaura, Minatoku, Tokyo 108, Japan)

It has become more difficult to create a function test pattern due to the recent increase of bus drivers containing tri-state drivers and bi-directional busses with advanced-functions and high density digital PCB.

The test generation program for the sequential circuits including bus drivers has been developed by adopting bus drivers modelling and introducing a high impedance state as the logic value in the Murakami's nine-valued circuit model in order to enable bus drivers handling.

The validity of this program has already been confirmed.

The technique of test generation centers on bus drivers and the execution results are included in this report.

## 1. まえがき

近年、デジタル PCB の高機能化、高密度化に伴ってトライステート素子や双方向バスを有するバス回路が増大し、機能試験用テストパターン作成は、増々難しくなっている。

これまで、機能試験のテストパターンは、機能面に着目して作成した人手パターン及び乱数発生パターンに対して、故障シミュレーションを用いる方法が採られてきた。しかし、この方法では、バス回路上のデバイス・IC の論理値衝突を回避するために、人手作業が難解・煩雑になったり、機能検査の実施によって、デバイス・IC の劣化・破損が発生する場合があった。今回、バス回路も取り扱い可能とするため、9 値法<sup>(1)</sup>にバス回路のモデル化及び論理値として新たにハイインピーダンス状態を導入した、バス回路を含む順序回路用テストパターン自動生成プログラムを開発し、既存の故障シミュレータ<sup>(3)</sup>と組み合わせて、CATS-I (Computer Aided Test System-Integration) システム<sup>(4)(5)</sup>を構築した。

本報告では、バス回路を中心としたテストパターン自動生成の手法とその実行結果について報告する。

## 2. バス回路取り扱い上の基本的考え方

バス回路を含む順序回路のテストで問題となるのは、次の 2 点である。<sup>(6)</sup>

### (1) バス回路上の論理値衝突

バス回路を構成するデバイス・IC の劣化・破損を防ぐため、バス回路に出力する複数素子の論理値が、0 と 1、0 と 0 または、1 と 1 の組み合わせ（これを論理値衝突という）にならない様にする必要がある。

### (2) 一般素子入力線へのハイインピーダンス状態印加

バス回路を構成するデバイステクノロジー (MOS) によっては発生する恐れのあるデバイス・IC の劣化、電荷蓄積効果などによる誤動作を防ぐため、バス回路を入力する一般素子 (AND, FF, ...) の入力線の論理値が、ハイインピーダンス状態にならないようにする必要がある。

この問題を解決するため、次のような手法を用いたテストパターン自動生成方法を考案した。

問題点 (1) については、テストパターン自動生成において、トライステート素子、バス素子のモデル化及び論理値として新たにハイインピーダンス状態“Z”の導入で解決した。

問題点 (2) については、素子入力線の論理値が Z 状態になることを禁止する方法<sup>(7)</sup>もあるが、本手法では、論理設計段階で、バス回路にプルアップ抵抗又は、プルダウン抵抗を付けていることに着目し、バス回路に出力する全素子が Z 状態になっても、バス回路を入力する一般素子の入力線の論理値を 1 又は、0 として扱うことで解決した。

## 3. アルゴリズム

### 3.1 故障モデル

取り扱う故障モデルは、0 および 1 の単一縮退故障のみを対象とする。

### 3.2 対象論理値

Z 状態を導入しさらに、Muth の 9 値法<sup>(1)</sup>の長所である故障伝播時の値の要求を必要最小限とすることによる経路選択回数の減少とより遠くまでの故障伝播が可能な利点を得るため、対象論理値は、故障シミュレーションで用いる 0, 1, X (don't care), Z の論理値 4 値に対して、さらに Muth の 9 値法の考え方と同様に、正常回路と故障回路の組み合わせ (: 正常回路 / 故障回路) で表現した 16 値を用いている。

故障 \ 正常	0	1	X	Z
0	0/0	0/1	0/X	0/Z
1	1/0	1/1	1/X	1/Z
X	X/0	X/1	X/X	X/Z
Z	Z/0	Z/1	Z/X	Z/Z

故障伝播情報 (斜線) 内は 9 値法の値

但し、故障印入は 1/0 又は 0/1 論理値 : 正常回路 / 故障回路

表 1 16 値論理値表

表1の16値論理値表において、0及び1縮退故障は、1/0、0/1であり、故障情報の伝播に使用できる値は、1/0、0/1、Z/0、0/Z、Z/1、1/Zの6値であり、また、0/0=0、1/1=1、X/X=X、Z/Z=Zの意味で使用している。

### 3.3 素子モデル

対象とするモデルを、表2に示す。

表2 素子モデル

素子モデル	
外部端子	入力端子、出力端子、 双方向バス端子
基本ゲート	AND, NAND, OR, NOR, EOR, ENOR, NOT, BUFFER, 等
FF	Dタイプ, Tタイプ, SRタイプ, JKタイプ (スラッシュトリカール)
トライステート	TRI素子
バス	BUS素子
メモリ	ROM素子

本手法の特徴であるバス回路は、トライステート素子とバス素子を用いて図1に示すようにモデル化する。

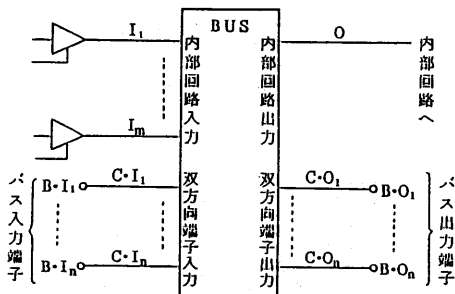


図1 バス回路の素子モデル

### (1) トライステート素子

トライステート素子は、制御入力線の論理値(入力値)が1または0の場合に、出力線の論理値(出力値)がそれぞれデータ入力と同値またはZ状態となる素子である。素子モデルを図2に、真理値表を表3に示す。

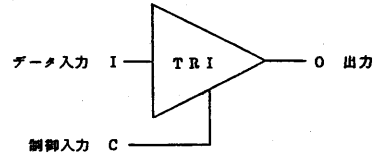


図2 トライステート素子の素子モデル

表3 トライステート素子の真理値表

入 力		出 力
C	I	O
1	1	1
1	0	0
0	X	Z

### (2) バス端子及びバス素子

双方向バス端子は、仮想的にバス入力端子とバス出力端子に分けて表現する。双方向バス端子が入力あるいは出力端子のいずれとして使われているかをモードで示す。このとき、各モードにおけるバス端子の真理値表を表4に示す。

バス素子は、多入力、多出力の仮想素子であり、双方向バス端子との接続用に双方向端子入力と双方向端子出力を持つ。

バス素子の真理値表は、表5のように、論理値衝突を回避するために、たかだか1つの入力を除き残りの入力の論理値を、全てZ状態になるように定義する。

なお、バス素子の故障伝播に使用する演算表(故障伝播演算表)についても、表6のように、論理値衝突を回避している。

表4 バス端子の真理値表

	入力モード	出力モード
バス入力端子	1 または 0	Z
バス出力端子	Z	1 または 0

表 5 バス素子の真値表

入 力				出 力		
$I_j$	$I_{1\sim m} \text{ 又 } I_j$	$C \cdot I_k$	$C \cdot I_{1\sim m} \text{ 又 } C \cdot I_k$	$C \cdot O_k$	$C \cdot O_{1\sim n} \text{ 又 } C \cdot O_k$	0
1	全て Z	Z	全て Z	1	全て 1	1
0	" Z	Z	" Z	0	" 0	0
Z	" Z	1	" Z	Z	" 1	1
Z	" Z	0	" Z	Z	" 0	0
Z	" Z	Z	" Z	Z	" Z	Z

表 6 バス素子の故障伝播演算表

入 力				出 力	
$I_j^*$	$I_1$	$I_{1\sim m} \text{ 又 } I_j, I_1$	$C \cdot I_{1\sim n}$	$C \cdot O_{1\sim n}$	0
1/0	Z/Z	全て Z/Z	全て Z/Z	全て 1/0	1/0
Z/0	1/Z	" Z/Z	" Z/Z	" 1/0	1/0
*1 Z/0	Z/Z	" Z/Z	" Z/Z	" 1/0	1/0
*2 Z/0	Z/Z	" Z/Z	" Z/Z	" Z/0	Z/0

\*: 故障情報を伝える入力

(A): " K Z を含まない場合

(B): " K Z を含む場合

\*1: (ブルアップ抵抗有)

\*2: ( " 無)

3. 4 処理手順概要

バス回路上の論理値衝突回避を実現したテストパターン自動生成手法の処理手順概要を図3を参照しながら説明する。

(1) まず、初期設定を行う。つまり、論理回路の内部状態をすべて、論理値 (X/X) で初期設定する (ステップ-1)。

(2) 次に、故障挿入を行う。つまり、故障位置 (論理回路中の故障を仮定した位置) に0縮退故障 (1/0) 又は1縮退故障 (0/1) を挿入する (ステップ-2)。

(3) 次に、16値用D操作を行う。つまり、故障位置の故障を出力外部端子で検出できるまで、16値を使用し、AND, OR等の基本ゲート、トライステート素子、バス素子等の各素子モデルごとに故障伝播させる操作を行う。トライステート素子、バス素子の故障伝播演算は、基本ゲートと同様に、正常回路の真値値と故障回路の真値値の交差により求めた演算表を参照することにより、出力側に故障を伝える。図4に、バス素子の故障伝播演算例を示す (ステップ-3)。

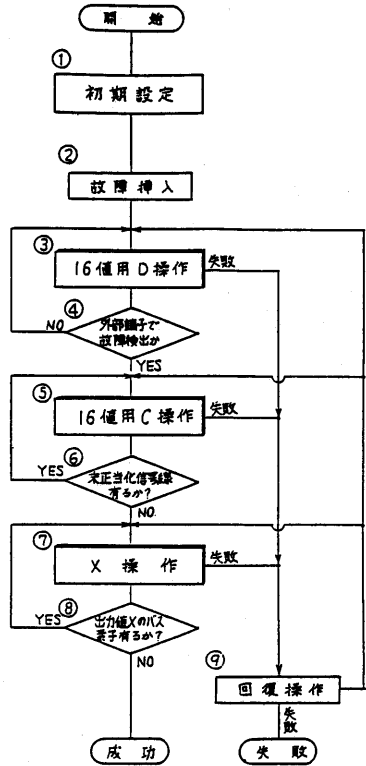


図 3 本手法の処理手順フロー

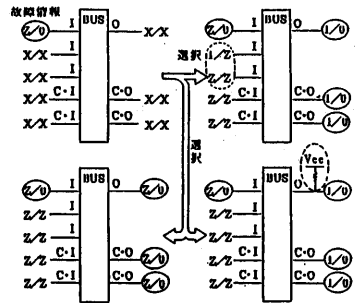


図 B 故障情報に Z を含む場合

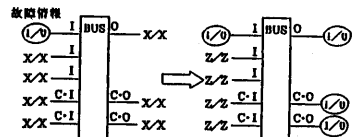


図 A 故障情報に Z を含まない場合

図 4 バス素子の故障伝播演算例

(4) 次に、外部端子での故障の検出の有無を判断する(ステップ4)。

その結果、故障が検出されない場合はステップ3に戻り、故障が検出された場合は次に進む。

(5) 次に、16値用C操作を行う。つまり、16値用D操作(ステップ3)で発生したすべての未正当化信号線(素子の出力値の正常回路又は故障回路の論理値が0, 1, Zに決定済みで入力値がXを持つ素子群)を、AND, OR等の基本ゲート、トライステート素子、バス素子等の各素子モデルごとに正当化させる操作を行う。トライステート素子、バス素子の正当化演算は、基本ゲートと同様に出力値を正常回路と故障回路に分けた出力値で、真理値表を参照することにより入力値を求める(ステップ5)。

(6) 次に、未正当化信号線の有無を判断する(ステップ6)。

この結果、未正当化信号線がある場合にはステップ5に戻り、未正当化信号線がない場合には、これにより、挿入した故障を、出力外部端子で故障検出できるテストパターンを生成できたことになる。

(7) 次に、X操作を行う。つまり、故障検出テストパターン生成後の正常回路又は故障回路の出力値がXのまま残っているすべてのバス素子に対し、その入力線で論理値0と1、0と0、1と1の衝突回避を保证するために、図5のように、バス素子の入力線の1つをXとし、残りをすべてZにし、決めたZのすべてに対し、素子モデルごとに正当化させる操作を行う(ステップ7)。

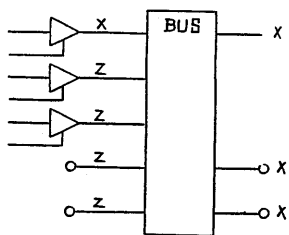


図5 バス素子のX操作例

(8) 次に、出力値Xのバス素子の有無を判断する。(ステップ8)

その結果、出力値Xのバス素子がある場合はステップ7に戻り、出力値Xのバス素子がない場合はテストパターンの生成に成功したことになる。

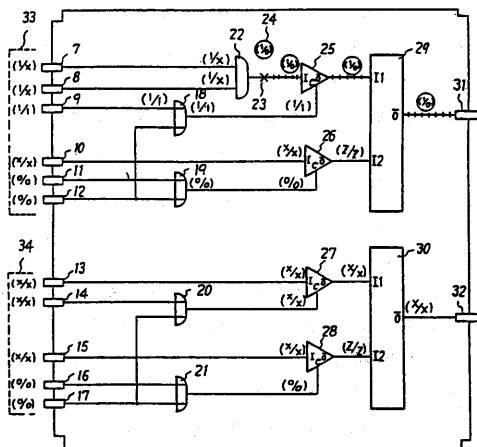
このようにして、16値用D操作(ステップ3)、16値用C操作(ステップ5)、X操作(ステップ7)の各操作がすべて成功した時、バス回路上の論理値衝突を回避したテストパターンが得られる。

(9) 16値用D操作(ステップ3)、16値用C操作(ステップ5)、X操作(ステップ7)の各操作で論理矛盾により失敗した場合、失敗直前の回路状態まで回復し、再度試行できるように準備する操作を行う(ステップ9)。この回復操作によってすべての試行が失敗した場合、テストパターン生成は失敗である。

### 3.5 適用例

図6を用いて本手法の適用例について説明する。

ここでは、2入力AND22の出力線の位置23が0縮退故障(1/0)した場合であり、この例について具体的に説明する。この場合、位置23に挿入した故障(1/0)をトライステート素子に伝えるために、トライステート素子の故障伝播演算表を参照



- |       |             |       |                   |
|-------|-------------|-------|-------------------|
| 23    | : 故障位置      | 29-30 | : バス素子            |
| 24    | : 0縮退故障     | 33    | : o操作より求めたテストパターン |
| 25-28 | : トライステート素子 | 34    | : *操作より求めたテストパターン |

図6 例題(デジタル回路図)

し、トリステート素子25のC入力線の論理値を(1/1)にする。次に、バス素子29に、故障(1/0)を伝えるために、バス素子の故障伝播演算表を参照し、バス素子29のI<sub>2</sub>入力線の論理値を(Z/Z)にする。この段階で、故障(1/0)が出力外部端子31に到達し、故障検出できる。ここまでの操作が前記した16値用D操作である。

ところで、2入力OR18やトリステート素子26のように、出力側の正常回路または故障回路の論理値が1やZに決定済みで、入力線がXである未正当化信号線が発生している。例えば、2入力OR18の出力線1/1の正常回路または故障回路の論理値1は2入力のOR18の真理値表を参照し、2入力OR18の入力線のどちらか一方を論理値1に設定すれば求まる。このように、16値用D操作で発生した未正当化信号線の出力値を正当化する外部入力端子の論理値を求める操作が、前記した16値用C操作である。この段階で、故障を検出する入力外部端子のテストパターン33を求めることができる。

次に、故障検出テストデータ生成後バス素子30のように、出力値の正常回路又は故障回路がXを持つバス素子が存在する場合がある。この場合、バス素子30の論理値衝突回避を保証するために、バス素子30のI<sub>1</sub>入力線に論理値(X/X)、I<sub>2</sub>入力線に(Z/Z)を設定し、トリステート素子28の出力値(Z/Z)を正当化する外部入力端子の論理値を求める。このように、故障伝播経路外のバス素子における論理値衝突回避を保証する入力外部端子の論理値を求めるのが、前記したX操作である。

この段階でバス回路上の論理値衝突を回避したテストパターン33と34とを得ることができる。

#### 4. 実行結果

本手法(ATG)と乱数発生法(乱数)による実行結果を表7に示す。本手法は、バス回路上の論理値衝突を回避したテストパターンを生成しているため、乱数発生法に比べ、2-8倍のCPU時間がかかっているが高い故障検出率が得られた。

#### 5. あとがき

本報告では、バス回路を含む順序回路のテストパターン自動生成の手法を開発し、その有効性を

表7 実行結果

回路	FF	TRI	双方向バス素子	論理値衝突(乱数)	故障検出率	故障検出率(n)		CPU時間K1(相対値)	
						乱数	ATG	乱数	ATG
A	x	o	x	153	288	96	108	1	2
B	x	o	o	660	764	40	92	1.1	2.2
C	o	o	o	558	834	42	94	2	1.7

\*1 乱数発生法によるCPU時間は、連続未検出パターン数=5パターンで打ち切った。

確認できた。本手法は、現在CATS-I:PCBテストシステムに組み込み実用に供してる。

終わりに、本手法を開発するにあたり、御協力戴いた島田章課長を始め関係者各位に感謝致します。

#### — 参考文献 —

- (1) 村上他 'LSI機能試験系列発生の一手法' (信学技法 SSD80-13 1980)
- (2) P. MUTH 'A Nine-valued circuit Model Test Generation' (IEEE TRAN S. Comput. C-25-1976)
- (3) 植野他 'トリステート・双方向バスを含む回路の平行フォールトシミュレータ' (情処学会第30回全国大会, 7H-9)
- (4) 佐藤他 'CATS-I:PCBテストシステム-システム構成-' (情処学会第32回全国大会, 4U-8)
- (5) 山下他 'CATS-I:PCBテストシステム-テストパターン自動生成-' (情処学会第32回全国大会, 4U-9)
- (6) M. A. Breuer 'Test Generation models for BUSES and TRI-STATE DRIVERS' (IEEE ATPG WORK SHOP PROCEEDINGS 1983)
- (7) 板崎他 'トリステートモジュールを含む回路のアルゴリズム的検査系列生成について' (信学技法 FTS85-22 1985)