

## シヨートノート

## 樹枝状に分割可能な組合せ回路の故障検出入力生成法†

橋 爪 正 樹\*\* 山 本 博 資\*\*\*  
 為 貞 建 臣\*\*\*\* 高 橋 一 磨\*\*\*\*

回路モジュール内に分岐再収れんを含んでもよいが、回路モジュール分割により樹枝状回路となる組合せ回路において、回路モジュールごとに求めた故障検出入力から回路全体の単一縮退故障の検出入力を求める方法について述べる。本論文では、(1)本手法により回路モジュールごとに求めた故障検出入力から回路全体の故障検出入力を求めることができること、また(2)回路はすでに分割されていると考え、故障検出入力生成時間のみを考慮すると、回路モジュールに分割せずに故障検出入力を求めるよりも、本手法の方が高速にかつより少ない故障検出入力を求めることができることを示す。

## 1. ま え が き

近年、回路規模の増大に伴い、回路の故障検出入力生成が困難となっている。過去にもさまざまな組合せ回路の故障検出入力生成手法が提案されているが、それらの手法は素子数の増加に対し指数関数的に故障検出入力生成時間が増加し、大規模回路への適用が困難である。しかし、検査対象回路を回路モジュールに分割し、回路モジュールごとに故障検出入力を求め、それらから回路全体の故障検出入力が求められれば、一度に扱う回路規模が小さくなるため、高速に故障検出入力を求められることが期待できる。

そこで、我々はまず手始めとして  $N$  入力 1 出力の回路モジュールから構成される樹枝状回路に分割できる回路の故障検出入力生成法の開発とその評価を行った。本稿では、2章で故障検出入力生成法について述べ、その評価結果を3章で述べる。

## 2. 故障検出入力生成法

本アルゴリズムは、組合せ回路の単一縮退故障を検出するためのもので、図1のような  $N$  入力 1 出力の回路モジュールから構成される樹枝状回路の故障検出

入力を求めることができる。具体的には次の手続きで故障検出入力を生成する。なお、本手法は付録に述べるように、冗長故障以外の故障をすべて検出することができる完全なアルゴリズムである。

## 【故障検出入力生成手続き】

(ステップ1) 回路を図1のように樹枝状回路となるよう分割する。なお、各回路モジュール内の回路は樹枝状回路でなくてもよく、分岐再収れんを含んでもよい。

(ステップ2) 各回路モジュールごとに完全な故障検出法により故障検出入力を求める。故障検出入力を求める順序は任意で、また最小でなくてもよい。各モジュールの故障検出入力が求めれば、各モジュールの入力信号線名・信号値だけでなく、その故障検出入力で得られる出力値、およびその故障検出入力で検出できる入力信号線の縮退故障も同時に記憶しておく。

(ステップ3) 次の故障検出入力生成アルゴリズムで回路全体の故障検出入力を生成する。

## 【故障検出入力生成アルゴリズム】

- ①: 出力側から最も離れた回路モジュールを選択する。
- ②: 選択した回路モジュールの故障検出入力の1つを選択し、その故障検出入力のフラグをたてる。なお、選択時にはフラグの立っていない故障検出入力を優先的に選択する。
- ③: 選択した回路モジュールの出力側回路モジュールの故障検出入力を選択し、その故障検出入力のフラグを立てる。ここでは、②で選んだ故障検出入力の出力値が0(1)なら、入力信号線で1(0)縮退故

† Test Generation Algorithm for Combinational Circuits Divided into Fanout-free Module Circuits by MASAKI HASHIZUME (Department of Electronic Engineering, Technical College of Tokushima University), HIROSUKE YAMAMOTO (Department of System Communication, The University of Electronic-Communications), TAKEOMI TAMESADA and KAZUMA TAKAHASHI (Department of Electronic Engineering, Faculty of Engineering, Tokushima University).

\*\* 徳島大学工業短期大学部電子工学科

\*\*\* 電気通信大学電子情報工学科

\*\*\*\* 徳島大学工学部電子工学科

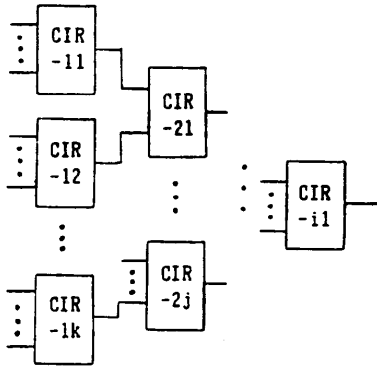


図1 検査対象回路(樹枝状回路)  
Fig. 1 Circuits to be diagnosed (fan out free).

障を調べられる故障検出入力を選択する。

- ④: ③で選んだ故障検出入力が得られるよう入力側回路モジュールの故障検出入力を1つ選択する。ここで③で選んだ故障検出入力で出力側回路モジュールの他の入力信号線の故障を検出できるならば、選択した故障検出入力のフラグを立てる。
- ④の操作を外部入力信号線の値が定まるまで繰り返す。
- ⑤: ③, ④の操作を繰り返し、外部入出力信号線の値を求め、故障検出入力とする。
- ⑥: ②で選んだ故障検出入力以外の故障検出入力に対して、③~⑤の操作を行う。ただし②で選んだ故障検出入力と同じ出力値が得られる故障検出入力に対しては③~⑤の操作で求めた故障検出入力と組み合わせて故障検出入力を生成する。
- ⑦: ②~⑥までの操作を各回路モジュールの故障検出入力フラグが立っていない故障検出入力がなくなるまで繰り返す。

### 3. 実験結果とその考察

冗長故障を含まない回路(回路A, C)と含む回路

表1 回路モジュールに分割しない方法との性能比較  
Table 1 Comparison with the algorithm not divided into module circuits.

回路	ゲート数	モジュール数	入力生成時間(sec)		検査入力数		使用メモリ量(kバイト)	
			Alg #1	Alg #2	Alg #1	Alg #2	Alg #1	Alg #2
A	33	3	56	145	23	40	218	233
B	30	3	90	1,223	17	29	210	226
C	187	17	372	10,760	117	240	292	269

Alg #1: 回路モジュールに分割して求める方法  
Alg #2: 回路モジュールに分割せずに求める方法

(回路B)について故障検出入力を生成した結果を表1に示す。なお、各回路モジュールの故障検出入力生成法として、冗長故障以外の故障を検出可能な手法であればよく、Dアルゴリズム<sup>2)</sup>やFANアルゴリズム<sup>3)</sup>などの手法を用いてもよいが、本稿ではPODEM法<sup>1)</sup>を用いた。さらに以下では、回路はすでに分割されていると仮定し評価する。

表1より、本アルゴリズム(Alg #1)を使用すれば、回路モジュール分割せずに回路全体の故障検出入力を求める方法(Alg #2)に比べ、入力生成時間を3~30分の1に短縮でき、故障検出入力も約50%少なくて済む。特に回路内に冗長故障が含まれる場合、PODEM法ではすべての検査入力を試すため、時間がかかるが、本手法では回路規模の小さい回路モジ

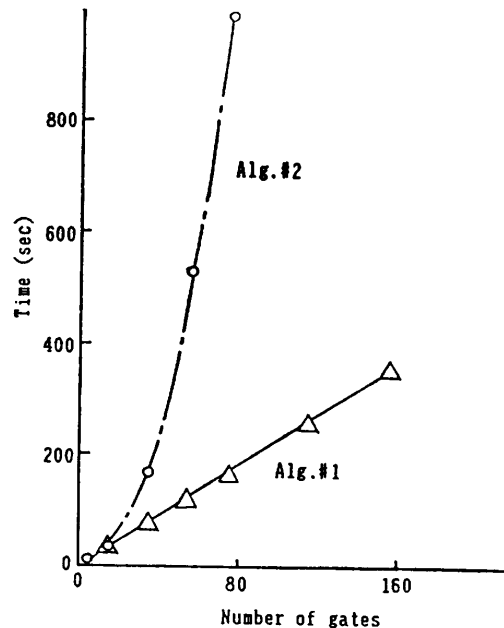


図2 素子数に対する本手法の評価  
Fig. 2 Effectiveness of this algorithm for number of gates.

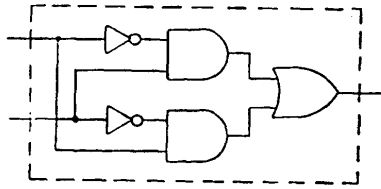


図 3 検査対象回路  
Fig. 3 A diagnosed circuit.

ジュールで冗長故障の存在を検出するため、高速に故障検出入力を求めることができる。また、必要なメモリ量はほぼ同程度にすんでいる。それは、(1) 樹枝状に分割可能な組合せ回路で単一故障を仮定しているため、各回路モジュールの入力信号線の縮退故障のすべての組合せに対して入出力間の故障の伝搬関係を記憶する必要がないこと、さらに(2) 検査入力生成アルゴリズムでは回路全体の接続関係を記憶する必要がなく、それより非常に単純なモジュール間の接続関係だけを記憶すればよいとめと考えられる。

図 2 に冗長故障を含まない図 3 の回路を多段に接続した時の素子数と故障検出入力生成時間の関係を示す。本評価では使用した計算機のメモリ容量の制限からゲート数が 155 個の回路までしか求められなかったが、回路分割をせずに故障検出入力を求めた場合に比べ高速に求められることがわかる。なお、ゲート数が数千以上になり、かつモジュール数が増加すると、本手法では各回路モジュールの故障検出入力数が増加し、必要なメモリ容量が増加する。また必要な検査入力生成時間が指数関数的に増加すると考えられる。

本実験では PODEM 法における一次入力信号線の値は出力信号線から最も離れている入力信号から順に一樣乱数を用いて発生した。また、表 1 および図 2 の各結果は NEC パーソナルコンピュータ PC-98 XA 上の LISP での実行結果である。

#### 4. む す び

本稿では、 $N$  入力 1 出力の回路モジュールから構成される樹枝状回路に分割できる組合せ回路に対し、前もって求めた各回路モジュールの故障検出入力から、回路全体の故障検出入力を求める手法について述べた。本稿で述べたアルゴリズムは冗長故障以外の故障をすべて検出することができる。また、回路はすでに分割されているとし、故障検出入力生成時間のみを考慮すると、回路モジュールに分割せずに回路全体の故障検出入力を求める方法に比べ、高速でかつ、よ

り少ない故障検出入力を求めることができることが判明した。今後の課題は、本手法が適用できるような回路の設計手法の開発、および回路分割により樹枝状にならない回路に関しても回路モジュールの故障検出入力から回路全体の故障検出入力を求められるように、故障検出入力生成手法を改良することである。

#### 参 考 文 献

- 1) Goel, P.: An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits, *IEEE Trans. Comput.*, Vol. C-30, No. 3, pp. 215-222 (1981).
- 2) 樹下, 藤原: デジタル回路の故障診断 (上), pp. 65-79, 工学図書, 東京 (1983).
- 3) Fujiwara, H. and Shimono, T.: On the Acceleration of Test Generation Algorithms, *IEEE Trans. Comput.*, Vol. C-32, No. 12, pp. 1137-1144 (1983).

#### 付 録

[定理] 本手法は完全なアルゴリズムである。

(証明)

組合せ回路が 1 つの回路モジュールから構成されていると仮定する。その場合、回路モジュールの故障検出入力生成アルゴリズムが完全であるため、明らかに本手法は完全である。

$n$  個の回路モジュールから構成されている回路に対して本手法が完全であると仮定する。その時、 $n$  個の回路モジュールからなる回路 CIR # $n$  の入力の 1 つに回路モジュール CIR #1 が付け加わった図 A.1 の回路を考える。CIR #1 の出力には CIR #1 内の故障が現れる。CIR #1 の出力が 0 で CIR #1 内の故障で出力信号線  $I_1$  の 1 縮退故障およびそれと等価な故障が検出できる。本手法では 2 章のアルゴリズム ③ で CIR # $n$  に対し、入力信号線  $I_1$  の 1 縮退故障を検出できる故障検出入力を割り当てるため、単一故障の仮

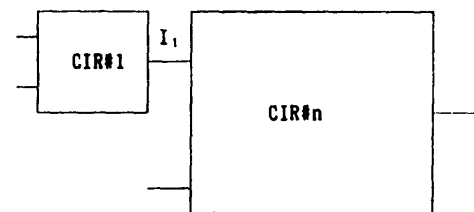


図 A.1  $n+1$  個の回路モジュールから構成される組合せ回路

Fig. A.1 A combinational circuit consisting of  $n+1$  module circuits.

定により, CIR #1 の故障を CIR # $n$  の入力信号線  $I_1$  の1縮退故障と等価な故障にすることができる. また同様に CIR #1 の出力1に対し, CIR # $n$  の入力信号線  $I_1$  の0縮退故障と等価な故障にすることができる. したがって,  $n+1$  個の回路モジュールからなる回路に対して本手法で故障検出入力生成が可能である.

(証明終)

(昭和62年11月6日受付)

(昭和63年4月14日採録)



**橋爪 正樹 (正会員)**

昭和32年生. 昭和54年徳島大学工学部電気工学科卒業. 昭和56年同大学院修士課程修了. 同年日本電信電話公社入社, 昭和58年徳島大学工業短期大学部助手, 現在に至る.

回路の論理設計用 CAD に関する研究に従事. 電子情報通信学会, IEEE 各会員.



**山本 博資**

昭和27年生. 昭和50年静岡大学工学部電気工学科卒業. 昭和55年東京大学大学院工学系研究科博士課程電気工学専攻修了. 工学博士. 同年徳島大学工学部助手. 同講師, 助

教授を経て, 昭和62年電気通信大学電気通信学部助教授. おもに情報理論の研究に従事. IEEE, 電子情報通信学会, 情報理論とその応用学会各会員.



**為貞 建臣**

昭和15年生. 昭和38年徳島大学工学部電気工学科卒業. 昭和41年同大学院修士課程修了. 同年同大工学部助手. 現在, 同大工学部電気電子工学科教授. 京都大学工学博士.

その間, 準安定状態をもつ順序回路, 論理ゲートを用いたマルチバイブレータ, 高速演算方式, 電子回路のCADなどの研究に従事. 著書「電子回路I」(朝倉書店)など5冊. 電子情報通信学会, IEEE 各会員.



**高橋 一彦**

昭和40年生. 昭和62年徳島大学工学部電子工学科卒業. 同年富士通徳島システムエンジニアリング入社. 在学中, 論理回路の故障診断に関する研究に従事.