

## 組合せ回路における分岐指向型検査入力生成法†

藤原 秀雄\*\* 下野 武志\*\* 尾崎 弘\*\*

Dアルゴリズムは誤り訂正/変換回路の様な排他的論理和 (XOR) 素子の多い回路に対して効率が悪いことが指摘されており、これを改良するために最近 PODEM アルゴリズムが考案されている。本論文では、PODEM アルゴリズムよりさらにバックトラックの頻度の小さい高速のアルゴリズムを考案したので報告する。また、アルゴリズムの効率を高めるために新しい可検査性尺度を導入し、すでに知られている尺度とともに、アルゴリズムに尺度を適用した場合の有効性についても実験結果を示しながら考察する。ここで提案するアルゴリズムは 1,000 ゲート以上の大規模回路に対してとくに有効であることを示す。

### 1. まえがき

回路の高集積化が進むにつれて、検査入力生成の対象となる回路の規模がますます大きくなる傾向があり、それとともに検査入力生成に要する計算費用が膨大なものとなってきている。これを解決するためには、LSSD<sup>1),2)</sup>のような検査容易な設計技法で論理設計を行うとともに、さらに効率のよい検査入力生成法を考案することが必要であろう。LSSD のようなスキャンパス方式の論理設計が採用された論理回路に対しては、組合せ回路のレベルまで検査容易化が達成されている。したがって、組合せ回路の検査入力生成において効率のよい方法を考えれば十分である。

検査入力生成において、多くの故障は無作為に生成した検査入力で検出される確率が高いので、高い故障検出率を望まない場合は、乱数パターンを用いて故障シミュレーションを行う方式が計算時間も速く有効である。しかし高い故障検出率を望む場合には、無作為に生成した乱数パターンだけでは能率が悪く、次のような方法が必要となる。すなわち与えられた故障が検出可能であるか否かを判定でき、検出可能な場合には常に検査入力を生成することができる方法である。このような検査入力生成法を“完全である”という。完全な検査入力生成法としては、ブール微分法<sup>3),4)</sup>、Dアルゴリズム<sup>5)</sup>、PODEM アルゴリズム<sup>6)</sup>等が報告されている。とくに実用化されているものとしては後の二者がある。Dアルゴリズムは誤り訂正/変換回路のような排他的論理和 (XOR) 素子の多い回路に対しては効率が悪いことが指摘されており、これを改良する

ために最近新しく PODEM アルゴリズムが考案されている。

本論文では、PODEM アルゴリズムよりさらにバックトラックの頻度の小さい高速のアルゴリズムを考案したので報告する。これは、PODEM アルゴリズムが外部入力においてのみバックトラックを発生するように工夫されているのに対して、外部入力側の分岐点においてバックトラックが発生するように工夫されており、他に、Dフロンティア<sup>5)</sup>が唯一ならば分岐点までDドライブ<sup>5)</sup>する、含意操作 (implication)<sup>5),6)</sup>を前方だけでなく後方にも行う、という特徴を有する手法である。本手法は分岐点に着目した手法であるので分岐指向型検査入力生成法 (FANout oriented algorithm for test generation, 以下では略して、FANアルゴリズム) とよぶことにする。

これらの検査入力生成アルゴリズムにおいては、その各ステップにおいて複数個の自由度のある選択が存在し、その選択が失敗したときは元にもどって別な選択を行うというバックトラックの操作が含まれている。アルゴリズムの効率を高めるには、この選択操作を無作為に行うのではなく、高い成功率を保証するある種の尺度を用いて選択操作に優先度を設けるといふ発見的方策を採用することが望ましい。本論文では、新しい可検査性尺度を導入し、すでに知られている尺度<sup>7)</sup>とともに、アルゴリズムに尺度を適用した場合の有効性についても実験結果を示しながら考察する。

### 2. FAN アルゴリズム

本論文では、Dアルゴリズム<sup>5)</sup>、PODEM アルゴリズム<sup>6)</sup>はよく知られているという前提で、そこで使われている用語の定義は省略する。対象とする回路は、AND, OR, NAND, NOR, NOT から成る組合せ回路とする。以下では単一縮退故障を想定している

† A Fanout Oriented Test Generation Algorithm for Combinational Circuits by HIDEO FUJIWARA, TAKESHI SHIMONO and HIROSHI OZAKI (Department of Electronic Engineering, Faculty of Engineering, Osaka University).

\*\* 大阪大学工学部電子工学科

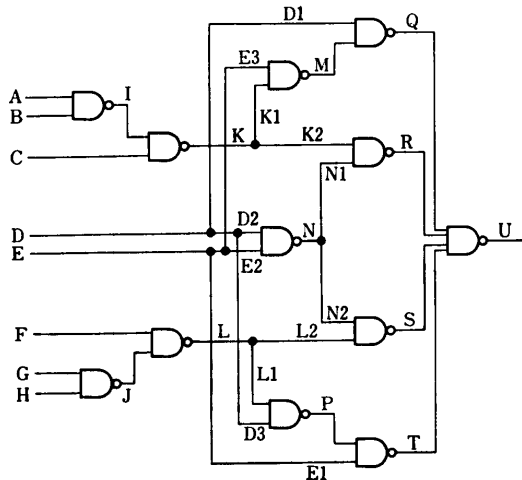


図 1 回路例

Fig. 1 Example of circuit.

が、固定故障であればより一般的な故障に対しても同様のアルゴリズムを考えることができる。

回路において、信号線が分岐している箇所を分岐点という。分岐点に入る信号線を分岐入力線、分岐点から出る信号線を分岐出力線とよぶ。分岐点から到達可能な信号線を束縛信号線とよび、これに隣接する非束縛信号線を先頭信号線とよぶ。

【例1】 図1の回路において、D、E、K、Lが先頭信号線である。この例から明らかなように、一般に先頭信号線数は外部入力数より少ない。

FAN アルゴリズムは PODEM アルゴリズムと比較して次の特徴を有するアルゴリズムである。

- (1) PODEM アルゴリズムでは、故障点に故障信号(Dまたは $\bar{D}$ )が発生するまで後方追跡(back trace<sup>6)</sup>)と含意操作の繰返しが行われるが、FAN アルゴリズムでは故障信号の挿入をまず行い、それにより生じる未正当化信号線(un-justified line<sup>5)</sup>)を、後方追跡と含意操作で正当化(line justification<sup>5)</sup>)する。
- (2) PODEM アルゴリズムでは、外部入力まで後方追跡を行うが、FAN アルゴリズムでは先頭信号線で後方追跡を停止する。したがって、バックトラックは先頭信号線でのみ発生するので PODEM アルゴリズムより少なくなる。
- (3) PODEM アルゴリズムでは、D伝搬は後方追跡と含意操作により行われ、Dドライブ操作は行われない。FAN アルゴリズムでは、D伝搬は後方追跡と含意操作により行われる以外に、Dフロンティアが唯一の場合には次の分岐点までD

ドライブを行う。これにより未正当化信号線が現れるので、正当化のための後方追跡も行う。

- (4) PODEM アルゴリズムでの含意操作は入力側から出力側への前方含意操作しか起こりえないが、FAN アルゴリズムでは前方だけでなく後方にも含意操作を行う。

以上の特徴を有する FAN アルゴリズムの概要フローチャートを図2に示す。

FAN アルゴリズムでは、与えられた回路を次の二つの部分に分けて考える。まず、先頭信号線より入力側の(非束縛信号線から成る)樹枝状部分回路。他の一つは、先頭信号線を外部入力とみなした束縛信号線から成る部分回路である。前者の部分回路は樹枝状回路であるので、D伝搬、信号線正当化の各操作の手数は  $O(m)$  である。ただし、 $m$  は回路の信号線数である。FAN アルゴリズムでは、後者の部分回路についてまず検査入力を求め、その後、前者の部分回路について信号線正当化を行い検査入力を完成する。

図2のフローチャートの各ブロックの説明を示そう。

#### (1) 故障信号挿入

故障信号線に故障信号(Dまたは $\bar{D}$ )を割り当て、検査対象素子の入出力の値が一意的に決まる場合、その値を割り当てる。図3に例を示す。入出力の値が一意的に決まる場合(図3(a)、(b))はDアルゴリズムにおける故障Dキューブに一致するが、一意的に決まらない場合(図3(c))は入力の値をXにしておく点はDアルゴリズムと異なる。分岐点の場合図3(d)のようにする。

#### (2) 含意操作

一意的に含意される値はすべて決める。入力側から出力側への前方操作と、出力側から入力側への後方操作の両方を行う。

#### (3) Dドライブ操作

Dフロンティアが唯一のときに限り次の分岐点が見られるまでDドライブを行う。すなわち、Dフロンティアから分岐点までの唯一の経路を活性化するために経路上の各素子入力に値を設定する。その際、発生する未正当化信号線は後で述べる後方追跡により正当化を行う。

#### (4) 正当化のための後方追跡

初期目標、目標=(目標信号線、目標値)は文献6)での initial objective, objective=(objective net, objective level)と同じ意味とする。正当化すべき信

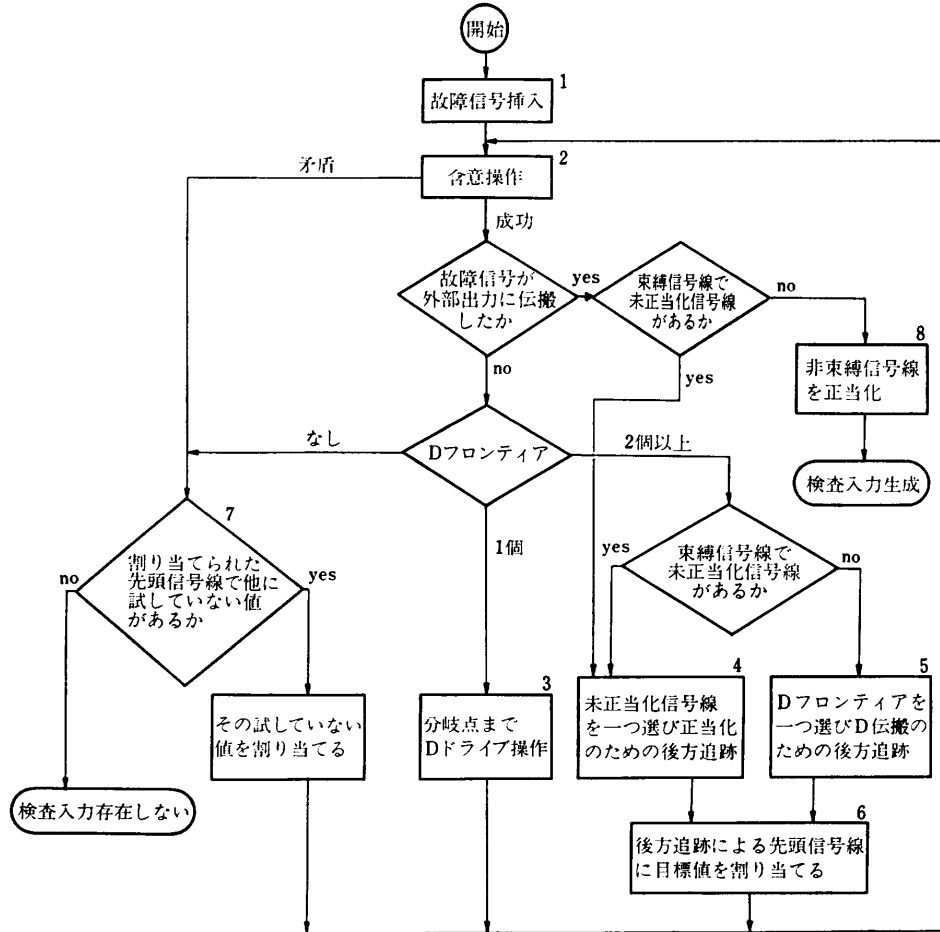
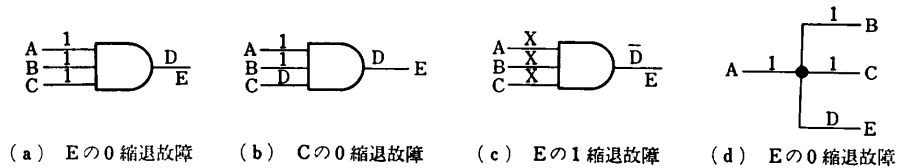


図 2 FAN アルゴリズムの概要フローチャート

Fig. 2 Flowchart for FAN algorithm.



(a) Eの0縮退故障 (b) Cの0縮退故障 (c) Eの1縮退故障 (d) Eの0縮退故障

図 3 故障信号挿入

Fig. 3 Insertion of fault signal.

号線を  $l$ , その値を  $v$  とすれば, 初期目標を  $(l, v)$  とおく. 追跡の仕方は PODEM アルゴリズムと同じである.

(5) D伝搬のための後方追跡

PODEM アルゴリズムの後方追跡と同じ.

(6) 先頭信号線の割当て

先頭信号線の後方追跡を停止し, そこを最終目標とし目標値を割り当てる. このとき, 判定木 (decision

tree<sup>6)</sup>) に対応する頂点を追加する.

判定木は PODEM アルゴリズムと同じ.

(7) バックトラック

含意操作で矛盾が生じるか, 外部出力に故障信号が伝搬しないうちにDフロンティアがなくなるとき, 判定木をバックトラックする.

(8) 非束縛信号線の正当化

この部分の信号線の正当化はDアルゴリズムにおけ

る正当化（一致操作）と同じ方法でバックトラックなしで行える。

【例2】 図1の回路において、信号線Nが1に縮退する故障の検査入力を FAN アルゴリズムで求めてみよう。

- (1) 故障信号挿入により、 $N = \bar{D}$ ,  $D2 = E2 = 1$  と割り当てる。
- (2) 含意操作により、 $D = 1, E = 1, D1 = 1, D3 = 1, E1 = 1, E3 = 1, N1 = \bar{D}, N2 = \bar{D}$ 。
- (5) RにDを伝搬するために初期目標を (R, 0) とする。(K2, 1)→(K, 1) と後方追跡。
- (6) 判定木に頂点 K=1, K=0 を追加する。K=1 を割り当てる。

- (2)  $K1 = 1, K2 = 1, R = D, M = 0, Q = 1$ 。
- (5) UにD伝搬するために初期目標を (U, 0) とする。(U, 0)→(S, 1)→(L2, 0)→(L, 0)。
- (6) 判定木に頂点 L=0, L=1 を追加する。L=0 を割り当てる。

- (2)  $L1 = 0, L2 = 0, S = 1, P = 1, T = 0, U = 1$  となりDフロンティアがなくなる。
- (7) バックトラックし、L=1 を割り当てる。
- (2)  $L1 = 1, L2 = 1, S = D, P = 0, T = 1, U = \bar{D}$  と含意された故障信号が外部出力に伝搬する。
- (8) 非束縛信号線  $K = 1, L = 1$  を正当化すると、 $C = 0, F = 0$  となる。

検査入力として、 $A = B = X, C = 0, D = E = 1, F = 0, G = H = X$  を得る。

以上の手順を判定木の上で見ると図4(a)のようになる。PODEM アルゴリズムでは図4(b)のようになる。バックトラック数、および通過する判定木の頂点数はいずれも FAN アルゴリズムのほうが少ない。

【例3】 図5の回路において、信号線Gの0縮退故障の検査入力を FAN アルゴリズムで求めてみよう。

- (1)  $G = D, K2 = 1, M = \bar{D}$ 。
- (2)  $K = 1, K1 = 1$ 。
- (3) Dフロンティアが唯一なのでDドライブする。 $N = 1, P = D$ 。
- (2)  $L = 0, A1 = J1 = 1, A = A2 = 1, J = J2 = 1, I = 0, H = C = 1, B = 0$  と含意される。
- (8)  $J = 1$  を正当化すると、 $E = 0$  となる。検査入力として、 $A = 1, B = 0, C = 1, E = 0, F = X, G = 1$  を得る。この場合、判定木には頂点が生成されず検査入力が生成されている。一方、

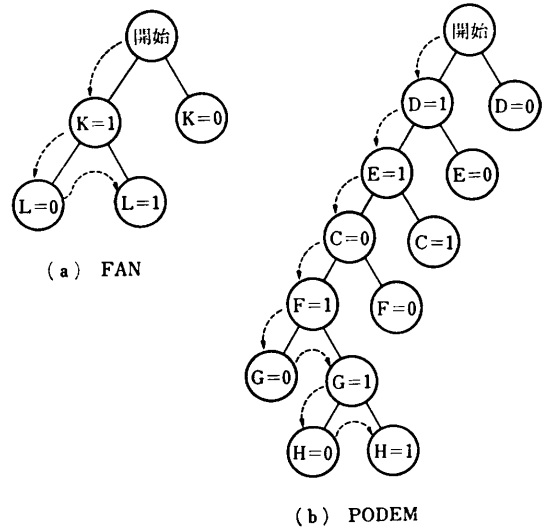


図4 判定木

Fig. 4 Decision tree.

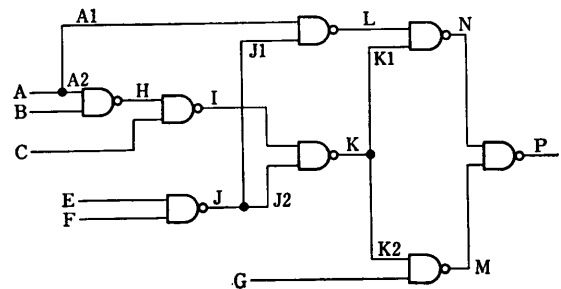


図5 回路例

Fig. 5 Example of circuit.

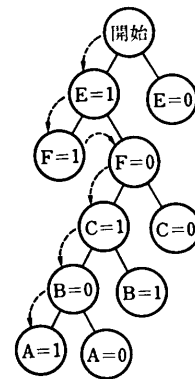


図6 判定木

Fig. 6 Decision tree.

PODEM アルゴリズムでは、図6のような判定木が生成され、バックトラックが発生している。

### 3. FAN アルゴリズムの完全性と計算複雑度

[定理 1] FAN アルゴリズムは完全である。

(証明) FAN アルゴリズムでは、回路を二つの部分に分けて検査入力を生成する。

- (1) 先頭信号線を入力とし束縛信号線から成る部分回路。
- (2) 非束縛信号線から成る樹枝状部分回路。

FAN アルゴリズムでは、(1)の部分回路の検査入力を生成後(図2の1-7)、(2)の回路の信号線の値を決め検査入力を求めている(図2の8)。(2)の回路は樹枝状回路であるので(1)の部分の検査入力生成に成功すれば常に検査入力を求めることができる。したがって、(1)の部分の検査入力生成アルゴリズムが完全であることを示せば十分である。

図2の概要フローチャートから明らかなように、FAN アルゴリズムと PODEM アルゴリズムの相違点は、FAN アルゴリズムには、1の故障信号挿入がある点、2の含意操作で後方操作がある点と、Dフロンティアが唯一のとき、3のDドライブ操作がある点である。これらの操作により決められる値は、その時点での割当てで検査入力が存在する場合はいずれ割り当てる必要のある値である。しかも、これらの操作により未正当化信号線が現れるが、これらは後方追跡と先頭信号線の目標値の割当てにより正当化の手続きが踏まれる。PODEM アルゴリズムが完全であること<sup>6)</sup>、以上のことより、FAN アルゴリズムにおいて検査入力が存在する場合は常に(1)の回路の入力となる先頭信号線にその検査入力に対応する値の組合せが現れその検査入力を求めることができる。すなわち、(1)の部分の検査入力生成アルゴリズムは完全である。(証明終)

次に、FAN アルゴリズムは PODEM アルゴリズムより効率のよいアルゴリズムであることを示そう。回路の外部入力数、先頭信号線数、全信号線数をおの、 $n, h, m$  とする。一般に  $m \geq n \geq h$  である。

[定理 2] FAN アルゴリズム、PODEM アルゴリズムの判定木において、アルゴリズムの実行中に通過する頂点数を  $N_F, N_P$ 、バックトラック数を  $B_F, B_P$  とすれば

$$N_F \leq N_P, B_F \leq B_P$$

である。また、FAN アルゴリズム、PODEM アルゴリズムの計算複雑度は、たかだか

$$O(2^h \cdot m), O(2^n \cdot m)$$

である。

(証明) PODEM と FAN の両アルゴリズムの相違点は2章で述べた4点がある。

第2番目の相違点により、 $n \geq h$  であるので判定木に生成される頂点数は、FAN アルゴリズムのほうが少ない。第1、第3、第4の相違点により、すなわち、故障信号の挿入、Dフロンティアが唯一のときのDドライブ操作、後方含意操作は、判定木に生成される頂点数を減らしこそすれ増やすことはない。また、通過する頂点数やバックトラック数についても同様である。したがって

$$N_F \leq N_P, B_F \leq B_P$$

である。明らかに

$$B_F \leq N_F \leq 2^h$$

$$B_P \leq N_P \leq 2^n$$

である。バックトラック間の各操作として、含意操作、Dドライブ、後方追跡等があるが、これらの計算複雑度は  $O(m)$  である。したがって、全体として、FAN アルゴリズム、PODEM アルゴリズムの計算複雑度はたかだか

$$O(2^h \cdot m), O(2^n \cdot m)$$

である。

(証明終)

### 4. 可検査性尺度

検査の難易度を表す可検査性尺度としては、可観測性と可制御性を表す尺度に分けて考えられる<sup>7)</sup>。Goldstein の尺度<sup>7)</sup> (以下、G 尺度とよぶ) は、信号線  $N$  の値を制御したり観測したりするために要する計算量の下限値を評価しているが、その最大の欠点は再収れん分岐を無視している点である。

ここでは、再収れん分岐を考慮した可制御性と可観測性を評価する尺度として可制御率  $PC^1, PC^0$  と可観測率  $PO$  を導入する (以下この尺度を F 尺度とよぶ)。

$PC^1(N)$ : 各信号線に値をたかだか1回割り当てるという条件で (換言すれば、バックトラックなしで)、信号線  $N$  に1を設定するのに成功する確率。

$PC^0(N)$ : 各信号線に値をたかだか1回割り当てるという条件で、信号線  $N$  に0を設定するのに成功する確率。

$PO(N)$ : 各信号線に値をたかだか1回割り当てるという条件で、信号線  $N$  の値を外部出力側に伝搬する (換言すれば、外部出力までの活性化経路を作る) のに成功する確率。

これらの確率を正確に計算するのは不可能に近い。また、尺度は検査入力生成の高速化に用いるのが目的であるから、尺度の計算に多くの時間を費すのは望ましくない。そこで、ここではF尺度を計算するのに次に示す簡単な計算で近似的な値を求めている。また、バックトラックが発生するのは再収れん分岐に起因することから、分岐点において成功確率が $k$ 倍 ( $0 < k < 1$ ) 減少するという仮定を設けている。

[PC<sup>1</sup>, PC<sup>0</sup> の計算]

PC<sup>1</sup>, PC<sup>0</sup> は回路の入力側から出力側へ次のようにして計算する。

(1) 外部入力  $I$  に対して

$$PC^1(I) = PC^0(I) = 1$$

(2)  $n$  入力  $X_1, X_2, \dots, X_n$  の AND ゲートの出力  $Y$  に対して

$$PC^1(Y) = \prod_{i=1}^n PC^1(X_i)$$

$$PC^0(Y) = \text{Max}_i \{PC^0(X_i)\}$$

(3)  $n$  入力  $X_1, X_2, \dots, X_n$  の OR ゲートの出力  $Y$  に対して

$$PC^1(Y) = \text{Max}_i \{PC^1(X_i)\}$$

$$PC^0(Y) = \prod_{i=1}^n PC^0(X_i)$$

(4) 入力  $X$  の NOT ゲートの出力  $Y$  に対して

$$PC^1(Y) = PC^0(X)$$

$$PC^0(Y) = PC^1(X)$$

(5)  $X_0$  から  $X_1, X_2, \dots, X_m$  への分岐に対して

$$PC^1(X_i) = k \cdot PC^1(X_0) \quad (1 \leq i \leq m)$$

$$PC^0(X_i) = k \cdot PC^0(X_0)$$

また、図7のような NOT を含む分岐の場合

$$\left. \begin{aligned} PC^1(X_i) &= k \cdot PC^1(X_0) \\ PC^0(X_i) &= k \cdot PC^0(X_0) \end{aligned} \right\} \quad (1 \leq i \leq m)$$

$$\left. \begin{aligned} PC^1(Y_j) &= k \cdot PC^0(X_0) \\ PC^0(Y_j) &= k \cdot PC^1(X_0) \end{aligned} \right\} \quad (1 \leq j \leq n)$$

[PO の計算]

PO は、PC<sup>1</sup>, PC<sup>0</sup> を用いて回路の出力側から入力側へ次のように計算する。

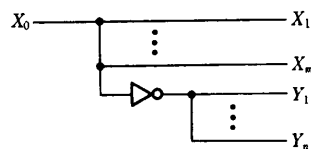


図7 NOT を含む分岐

Fig. 7 Fanout point with NOT gate.

(1) 外部出力  $U$  に対して

$$PO(U) = 1$$

(2) 入力  $X_1, X_2, \dots, X_n$ , 出力  $Y$  の AND ゲートの入力  $X_i$  に対して

$$PO(X_i) = PO(Y) \cdot \prod_{j \neq i} PC^1(X_j)$$

(3) 入力  $X_1, X_2, \dots, X_n$ , 出力  $Y$  の OR ゲートの入力  $X_i$  に対して

$$PO(X_i) = PO(Y) \cdot \prod_{j \neq i} PC^0(X_j)$$

(4) NOT ゲートに対して

$$PO(X) = PO(Y)$$

(5)  $X_0$  から  $X_1, X_2, \dots, X_m$  への分岐に対して

$$PO(X_0) = \text{Max}_i \{PO(X_i)\}$$

F尺度ではすべて $k$ のべき乗で表現されるので、その値の大小の比較は $k(0 < k < 1)$ の値に依存せず指数部の値の大小により決まる。したがって、後で述べる発見の方策で用いるF尺度では $k$ の値は $0 < k < 1$ のどの値でもよいことになる。

5. 発見の方策

アルゴリズムにおいて複数個の自由度のある選択が存在する場合、このアルゴリズムの効率を高めるためには、可検査性尺度を用いて各選択に優先順位を設けるという発見の方策が有効である。

PODEM アルゴリズムでは自由度のある選択は次の場合に生じる。

選択(1) 後方追跡の初期目標としてDフロンティアの一つを選ぶ場合。

選択(2) 後方追跡において、現在の目標が多入力ゲートの出力線上にあるとき、次の目標をどの入力線に伝えるかの選択。

この選択(1), (2)に対する発見の方策で、可制御性、可観測性尺度をどのように用いるかは文献6)と同じであるので省略する。

FAN アルゴリズムでは、上の二つの選択の他に次の場合にも自由度のある選択が生じる。

選択(3) 後方追跡の初期目標として未正当化信号線を選ぶ場合、どの信号線を選ぶかの選択。

この場合の発見の方策は次のように行う。未正当化信号線はすべて正当化する必要があるため、一つの信号線を正当化しても、いずれ残りの未正当化信号線を正当化しなければならない。このような場合には、失敗するものなら早い時期にその失敗を発見することがアルゴリズムの効率を高めるためにも望ましい。した

がって、未正当化信号線の値の可制御性の最も悪いものから順に選択する。

## 6. 実行結果

PODEM, FAN の両アルゴリズムのプログラム化を行い、700~2,000 ゲート程度から成る回路に適用して検査入力の生成を行った結果を表 1~3 に示す。

表 4 には回路データの特性を示す。使用計算機は大阪大学大型計算機 ACOS システム 1000 (15 MIPS) で、使用言語は FORTRAN である。検査入力生成には、各代表故障ごとに一つの検査入力を生成するという手続を踏んだ。表には参考のために、単一経路活性化法に基づく EALG (Eアルゴリズム)<sup>8)</sup> の実行結果も示している。

各アルゴリズムには発見的手法が適用されており、その方策を決める尺度として G 尺度および F 尺度を使用した。

表 1, 表 2, 表 3 には、これら六つのアルゴリズムに対する計算時間、検出率、平均バックトラック数が示されている。検出率は検査入力生成に成功した故障と代表故障数の比率である。( )内は、途中で打ち切られた故障の比率で、打切時間は 1 故障当り 2 秒である。平均バックトラック数とは、1 故障当りのバックトラック数である。

まず表 3 に示す平均バックトラック数については、EALG においてのみ G 尺度を用いたほうが F 尺度を用いたほうより効率がよいが、PODEM, FAN の両アルゴリズムにおいては、G 尺度より F 尺度のほうが効率がよい。アルゴリズム間では、明らかに、FAN が最もバックトラック数の少ないアルゴリズムであることがわかる。表 2 の検出率においても、EALG や PODEM より FAN のほうが優れており、打切故障も非常に少ない。とくに #3 の回路では EALG が極端に悪く、F 尺度を適用した FAN アルゴリズムが最も優れている。表 1 の計算時間においては、すべての回路について、EALG, PODEM, FAN の順に速くなっており、全体的に F 尺度を適用した FAN が最も高速である。

以上のことからアルゴリズムでは EALG, PODEM, FAN の順に効率がよくなり、FAN が最も効率のよいアルゴリズムであることがわかる。可検査性尺度に関しては、あまり大きな差が現れていないが、全体的に G 尺度より F 尺度のほうが若干優れている。ここでは示さなかったが、500 ゲート以下の小規模な回路に

表 1 計算時間 (秒)

Table 1 Computation time (second).

回路番号	EALG		PODEM		FAN	
	G 尺度	F 尺度	G 尺度	F 尺度	G 尺度	F 尺度
#1	164.7	171.6	39.2	35.2	28.1	27.6
#2	370.2	415.8	259.5	278.3	122.0	139.3
#3	551.0	584.0	497.3	430.0	301.4	197.4
#4	304.2	322.4	140.9	136.0	83.2	81.7

表 2 検出率 (%)

Table 2 Test coverage (%).

回路番号	EALG		PODEM		FAN	
	G 尺度	F 尺度	G 尺度	F 尺度	G 尺度	F 尺度
#1	99.0 (0.5)	98.8 (0.7)	99.2 (0.5)	99.3 (0.4)	99.5 (0.1)	99.5 (0.1)
#2	91.5 (4.4)	90.8 (5.0)	94.2 (3.6)	94.7 (3.2)	95.5 (1.4)	95.7 (1.1)
#3	81.0 (3.1)	81.0 (3.1)	93.0 (5.1)	93.6 (4.5)	94.0 (2.6)	95.1 (1.5)
#4	98.8 (0.1)	98.8 (0.1)	98.7 (0.4)	98.6 (0.4)	98.9 (0.1)	98.9 (0.1)

( )内は途中で打ち切られた故障の比率。  
打切時間は 1 故障当り 2 秒。

表 3 平均バックトラック数

Table 3 Average number of backtracks.

回路番号	EALG		PODEM		FAN	
	G 尺度	F 尺度	G 尺度	F 尺度	G 尺度	F 尺度
#1	40.1	43.7	6.6	4.7	2.5	2.4
#2	57.1	61.8	44.5	44.2	36.7	42.9
#3	51.0	61.6	90.9	71.7	38.1	21.9
#4	1.4	1.6	3.3	2.8	0.8	1.2

表 4 回路の特性

Table 4 Characteristic of circuits.

回路番号	ゲート数	信号線数	入力数	出力数	分岐点数	代表故障数
#1	718	1925	33	25	381	1871
#2	1003	2782	233	140	454	2748
#3	1537	4437	138	22	596	3674
#4	2002	5429	178	123	806	5350

についても実験したが、各アルゴリズム間の有効性に大きな差は現れなかった。ここで示したように、1,000 ゲートを越す大規模回路に対して、各アルゴリズム間に大きな差が現れ、先に述べたように F 尺度を用いた FAN アルゴリズムが最も有効なアルゴリズムであることがわかる。この傾向はより大規模な回路に対してとくに大きく現れるものと思われる。

## 7. あとがき

分岐点に着目した新しい検査入力生成法 (FAN アルゴリズム)

ルゴリズム)を提案した。バックトラックの回数において FAN アルゴリズムは PODEM アルゴリズムより効率のよいアルゴリズムであることを示した。さらに新しい可検査性尺度を導入しその有効性を示した。1,000 ゲートを越す実際の回路に対する実行結果では、F 尺度を用いた FAN アルゴリズムが最も効率のよいアルゴリズムであるという結果を得た。

### 参 考 文 献

- 1) Eichelberger, E. B. and Williams, T. W.: A Logic Design Structure for LSI Testability, Proc. 14th Design Automation Conf., pp. 462-468 (1977).
- 2) Yamada, A. et al.: Automatic System Level Test Generation and Fault Location for Large Digital Systems, Proc. 15th Design Automation Conf., pp. 347-352 (1978).
- 3) Sellers, F. F., Hsiao, M. Y. and Bearnson, L. W.: Analyzing Errors with the Boolean Difference, *IEEE Trans. Comput.*, Vol. C-17, No. 7, pp. 676-683 (1968).
- 4) Kinoshita, K., Takamatsu, Y. and Shibata, M.: Test Generation for Combinational Circuits by Structure Description Functions, Proc. 10th Int. Symp. on Fault Tolerant Computing, pp. 152-154 (1980).
- 5) Roth, J. P., Bouricious, W. G and Schneider, P. R.: Programmed Algorithms to Compute Tests to Detect and Distinguish between Failures in Logic Circuits, *IEEE Trans. Comput.*, Vol. EC-16, No. 10, pp. 567-580 (1967).
- 6) 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).
- 7) Goldstein, L. H.: Controllability/Observability Analysis of Digital Circuits, *IEEE Trans. Circuits Syst.*, Vol. CAS-26, No. 9, pp. 685-693 (1979).
- 8) 高松雄三, 藤原秀雄, 樹下行三: 単一経路活性化法に基づく検査系列生成プログラム (E アルゴリズム), 情報処理学会電子装置設計技術研究会資料 9-2 (1981).
- 9) 藤原秀雄, 尾崎 弘: 発見的手法によるテスト生成のための新しい尺度, 信学技報, EC 80-38 (1980).
- 10) 藤原秀雄, 下野武志, 尾崎 弘: 組合せ回路における分岐指向型検査入力生成法, 信学技報, EC 81-50 (1981).
- 11) 下野武志, 藤原秀雄, 尾崎 弘: 可検査性尺度に基づく検査入力生成効率の改善法, 情報処理学会電子装置設計技術研究会資料 13-4 (1982).

(昭和 57 年 3 月 15 日受付)

(昭和 57 年 7 月 12 日採録)