

# 検査入力生成アルゴリズムの高速化について

下野武志 藤原秀雄 尾崎 弘

(大阪大学 工学部)

## 1. まえがき

論理回路のLSI/VLSI化に伴い故障検査の問題がますます困難となり、この問題の解決が重要な課題となっている。特に、どれだけ高い検出率の検査入力集合をいかに速く求めるかという、検査入力生成アルゴリズムの高速化の問題が重要な課題であると思われる。これに関しては、否定素子を含まない単調な組合せ回路でさえも一般にNP完全であることが示されており[1]、大規模な回路になるほど非常に難しいことが分かる。

通常、組合せ回路の検査入力生成には、指定された故障の検査入力を解析的に生成する検査入力生成法と、生成された検査入力ごとの様な故障を検出できるかを調べる故障シミュレーション法を組合せた方法がとられている。検査入力生成法には、ブール微分法[2,3]、Dアルゴリズム[4]、9値アルゴリズム[5~7]、PODEMアルゴリズム[8]、FANアルゴリズム[10~12]等が発表されている。

著者らは、先にこれらのDアルゴリズムやPODEMアルゴリズムより効率の良い高速のアルゴリズムとしてFANアルゴリズムを報告した。検査入力生成アルゴリズムの高速化には、バックトラック回数を減らすことと、バックトラック間の処理を高速化することが必要である。本論文では、この高速化のための種々の手法を考察し、先に提案したFANアルゴリズムの改良版となるアルゴリズムを紹介し、その有効性を評価する。また、同時故障シミュレーション[16]と組み合わせることにより、かなり高速の自動検査入力生成

プログラムを作成することができたので、その結果も報告する。

## 2. アルゴリズムの高速化

本論文では、AND, OR, NAND, NOR, NOTから成る組合せ回路を対象とし、故障は単一縮退故障を想定する。

ここでは、アルゴリズムの高速化に着目して、Dアルゴリズム及びPODEMアルゴリズムにおける欠点をいくつか指摘しながら、その欠点を除去できるいくつかの有効な技法を紹介する。

検査入力生成アルゴリズムの高速化には、

- (1) バックトラックの発生回数を減らすこと。
- (2) バックトラック間の処理を高速化すること。

が必要である。特に高速に高検出率を達成するためには(1)の効果が重要である。分枝限定法(branch and bound)によるアルゴリズムでは、判定木において現在の節点から下位には解が存在しないことがわかった場合、すぐ上位の節点に戻り別の枝を選択する(バックトラック操作)。このバックトラックの発生回数を減らすためには、

- (1) ある時点でいくつかの選択が存在する場合、解の存在する確率すなわち成功率の高い方を選ぶ。

このために、種々の発見的手法を採用することが出来る[13]。

- (2) 判定木のある節点で、それより下位にいくら進んでも解が存在しない場合、それを早

期に発見する。もしくは、その様な節点が判定木に入らない様にする。

このためには、各時点で選択の余地なしに一意的に決まる値はできるだけ多く決めるべきである。この点に関してPODEMアルゴリズムはあまり考慮されておらず、不必要なバックトラックを多く行うことがある。

## 2.1 故障信号挿入の効果

図1の回路の信号線Lの1縮退故障を考えてみよう。故障信号DをLに挿入し、Lの素子の入力J, K, Eに1を割当てた後、含意操作を行えば一意的に検査入力が必要(図1(a))。一方PODEMでは、初期目標(L, 0)を設定し、図1(b)の様な後方追跡を行いBに0を割当てると、含意操作によりL=1となるので失敗となりバックトラックが発生する。この例で分かる様に故障信号挿入による一意的な値の割当ては、絶対必要な条件であり多くの不必要な選択を生じない様に行うことができる。

## 2.2 一意活性化の効果

「Dフロンティアが唯一の場合、そのDフロンティアから外部出力に至るどの経路もある部分経路を必ず通る時、その部分経路を活性化するために経路上の各素子入力に値を割当てる。」これを一意活性化という。

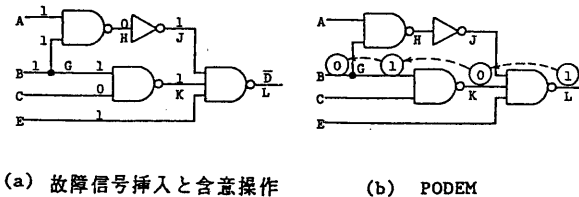


図1. 故障信号挿入による効果

図2の回路を考えてみよう。DフロンティアはゲートG2だけとする。G2から外部出力へ至るどの経路もF-HとK-Mの部分経路を必ず通る。したがってこの時点で検査入力が存在すれば、これらの経路を活性化せざるを得ない。そこで、C, G, J, Lに1を割当てる。この後含意操作を行えば図2(a)の様になる。一方PODEMでは、DフロンティアG2を活性化するために初期目標を(F, 0)として後方追跡を行うが、図2(b)の様に後方追跡を行いAに0を割当てると、含意操作によりJ=0, K=1となる。この割当てに解がないのは明らかであり、いずれ必ずバックトラックすることになる。

一意活性化は、多重経路の同時活性化に対しても有効である。例えば図3の回路の信号線Gの1縮退故障を考えてみよう。故障信号線から外部出力へ至る2つの経路G-J-MとG-K-Mに共通のMを活性化するためにMの入力のIとLに1を割当てる。この後

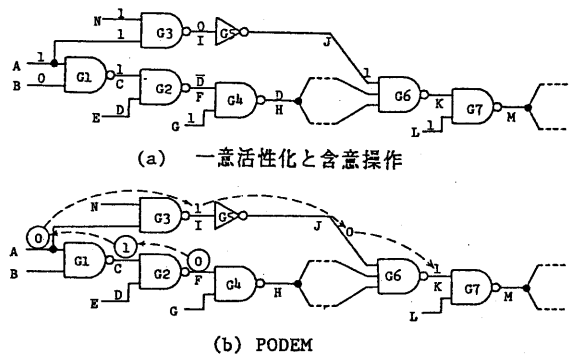


図2. 一意活性化の効果

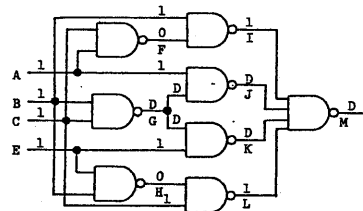


図3. 一意活性化による多重経路の活性化

含意操作を行えば、2つの経路が必然的に同時活性化され、一意的に検査入力が必要。

これらの例からも分かる様に、各時点で値が一意的に決まる信号線を一つでも多く見つけて値を割当てることはバックトラックの減少につながる。それだけ可能な選択の余地が少なくなるからである。

### 2.3 先頭信号線の効果

回路において、いずれかの分岐点から到達可能な信号線を**束縛信号線**と呼び、これに隣接する非束縛信号線を**先頭信号線**と呼ぶ。

図4(a)の回路を考えよう。信号線K, L, Mは束縛信号線で、それに隣接する信号線J, Hが先頭信号線である。今PODEMで、最初は(J, 0)の目標が設定されるが、J=1にならない場合は検査入力が存在しない場合を考えよう。PODEMでは、外部入力に達するまで後方追跡を行う結果、A=1, B=0を割当てて初めてJ=0となる。ここで解の不在が判明したとしてもJ=1となるまでには図4(b)に示す様にバックトラックを行う。これに対して、先頭信号線Jで後方追跡を止め、先頭信号線に値を割当てると、

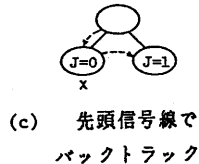
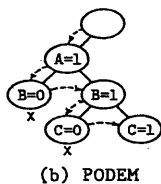
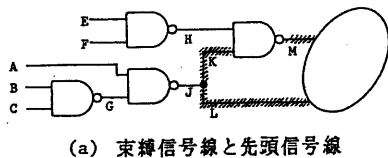


図4. 先頭信号線の効果

図4(c)の様にバックトラックの回数が減る。先頭信号線の値を満足する様に外部入力値を設定するのは常にバックトラックなしに行えるから、後方追跡は外部入力まで行わず、先頭信号線までで十分である。

一意活性化を行う時値を割当てるべき素子入力線を識別できる必要があり、後方追跡の時終点である先頭信号線を識別できる必要がある。これらは、検査入力生成前に回路テーブルなどと共に前処理で求めておけば、検査入力生成時に高速に処理できる。

### 2.4 多重後方追跡の効果

検査入力生成のために多数の先頭信号線に値を割当てなければならない時、多重後方追跡は単一後方追跡に比べて、値を割当てるべき信号線の集合を効率良く高速に求めることができる。図5の部分回路において、外部入力は同時に先頭信号線であるとする。Cに0を設定する目標に対して、単一後方追跡を用いると、Cに0が割当てられるまでにC-B-A, C-F-Eの2つの経路に沿って3回ずつ後方追跡が繰返される。多重後方追跡ではいずれの経路も一度ずつしか通らず無駄が少ない。

故障信号挿入や一意活性化などでは回路の内部の信号線に値を直接割当てるので、出力値が決まっているが入力値が決っていない未正当化信号線が一般に複数個生じる。Dアルゴリズムでは、この未正当化信号線を正当化するのに一致操作を行い、未正当化信号線の入

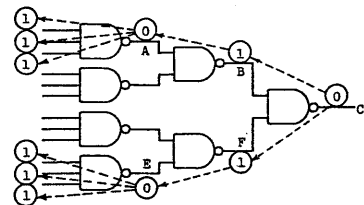


図5.

力に値を直接割当て、一つずつ正当化していく。これに対し多重後方追跡では、すべての未正当化信号線を初期目標として後方追跡を行うので、すべての未正当化信号線を正当化するために値を割当てるべき先頭信号線と値の組を一度に求めることができる。

以上の観点から、複数個の初期目標を複数個の経路に沿って同時に後方追跡する多重後方追跡の手法が有効であると推定される。次に、今回採用した多重後方追跡の手法について具体的に説明する。

多重後方追跡は、一般に複数個の初期目標から始める。これを初期目標群という。初期目標群から始めて途中の段階で現われる目標の集合を、その時点での現在目標群といい、最終的に先頭信号線で得られる目標の集合を先頭目標群という。又、分岐点における目標の集合を分岐目標群という。

多重後方追跡における目標は、

$$\text{目標} = (S, n_0(S), n_1(S))$$

$S$  : 目標信号線

$n_0(S)$  :  $S$  に目標値 0 が要求される回数

$n_1(S)$  :  $S$  に目標値 1 が要求される回数

で定義する。信号線  $S$  に値 0, 1 を設

定するという初期目標は、それぞれ、 $(S, 1, 0)$ ,  $(S, 0, 1)$  とする。これらの初期目標群から始めて、出力側から入力側へ各素子を通過する際に表 1 の規則に従って目標を定めていく(図 6 参照)。

多重後方追跡のフローチャートは図 7 の様になる。分岐点に達した目標はそこで一度止め置かれる。分岐点以外の目標がなくなると、次に最も出力側にある分岐目標を取出す。取出した目標が、(1) 故障信号線より入力側にあり、かつ、(2) 0 と 1 の両方の目標値の要求がある時、その分岐目標を最終目標として後方追跡を終了する。そして、要求度の大きい方の値を割当てて含意操作を行う。この理由は、分岐点において 0 と 1 の両方の目標値が要求されるということは、これまでの後方追跡のどこかに不都合が生じている可能性が大きく、この時点で検査入力の解が存在しない確率が高いと考えられるからである。したがって、先頭信号線まで後方追跡をしてから値を割当てるより、要求が矛盾する分岐点のところで先に値を割当てた方が「不一致の早期発見」につながり、バックトラック数を減らすことができると考えられる。条件(1)は、割当てる値が 0 と 1 だけであることを保証するためである。

表 1

素子	入力 $X$	$n_0(X)$	$n_0(X_i)$	$n_1(X)$	$n_1(X_i)$
AND	0 - 制御費最小	$n_0(Y)$	0	$n_1(Y)$	$n_1(Y)$
OR	1 - 制御費最小	$n_0(Y)$	$n_0(Y)$	$n_1(Y)$	0
NAND	0 - 制御費最小	$n_1(Y)$	0	$n_0(Y)$	$n_0(Y)$
NOR	1 - 制御費最小	$n_1(Y)$	$n_1(Y)$	$n_0(Y)$	0
NOT		$n_1(Y)$		$n_0(Y)$	
分岐点		$\sum_{i=1}^k n_0(X_i)$		$\sum_{i=1}^k n_1(X_i)$	

### 3. 改良版 FAN アルゴリズム

本節では、前節で述べたアルゴリズムの高速化の技法をすべて採用したアルゴリズムを述べる。これは、FAN アルゴリズム [10~12] の改良版であり、FAN-Ⅲアルゴリズムと呼ぶことにする。FAN-Ⅲの概要フローチャートは図 8 の様になる。

(1) 故障信号挿入

図 9 に例を示す。図 9 (b) では、故

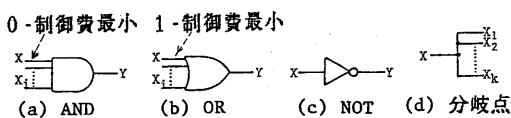


図 6

障信号線は未正当化信号線となる。

(2) 後方追跡初期化フラグ

多重後方追跡には、次の2つのモードがある。

(i) 初期目標群から始める。

(ii) 先に行った多重後方追跡の続きを分岐目標から行う。

初期化フラグが立っている時(i)のモードで、倒れている時(ii)のモードである。

(3) 含意操作

一意的に含意される値はすべて決める。前方含意(出力側へ)と後方含意(入力側へ)の両方を行う。

(4) 回路の状態の変化

多重後方追跡の初期目標がDフロンティアで、含意操作の結果そのDフロンティアが変化した場合、および、初期目標群とした未正当化信号線が含意操作の結果すべて正当化された場合、以前に行った多重後方追跡の結果を続けて利用するのは適当でないので、次の多重後方追跡は初期目標からやりなおす。そのために初期化フラグを立てる。

(5) 一意活性化

Dフロンティアが唯一の場合、前節で述べた一意活性化を行う。

(6) 値を割り当てる目標の探索

このフローチャートを図10に示す。この処理が終了後もすべての目標群はその

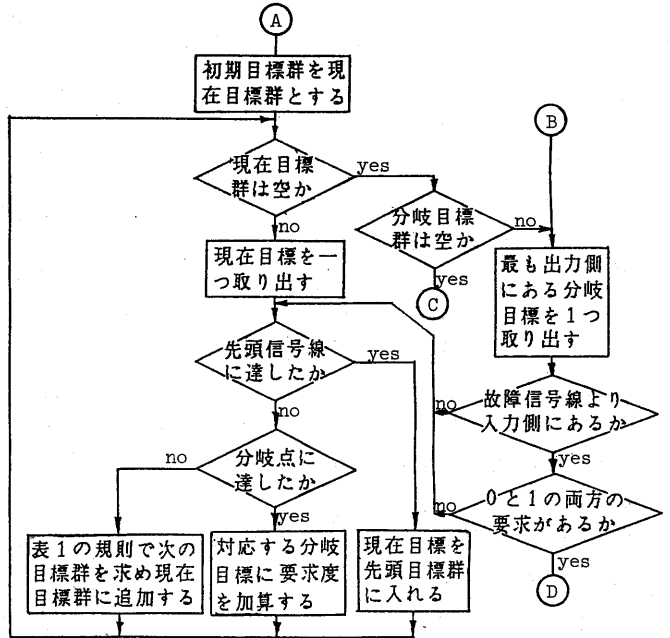


図7. 多重後方追跡のフローチャート

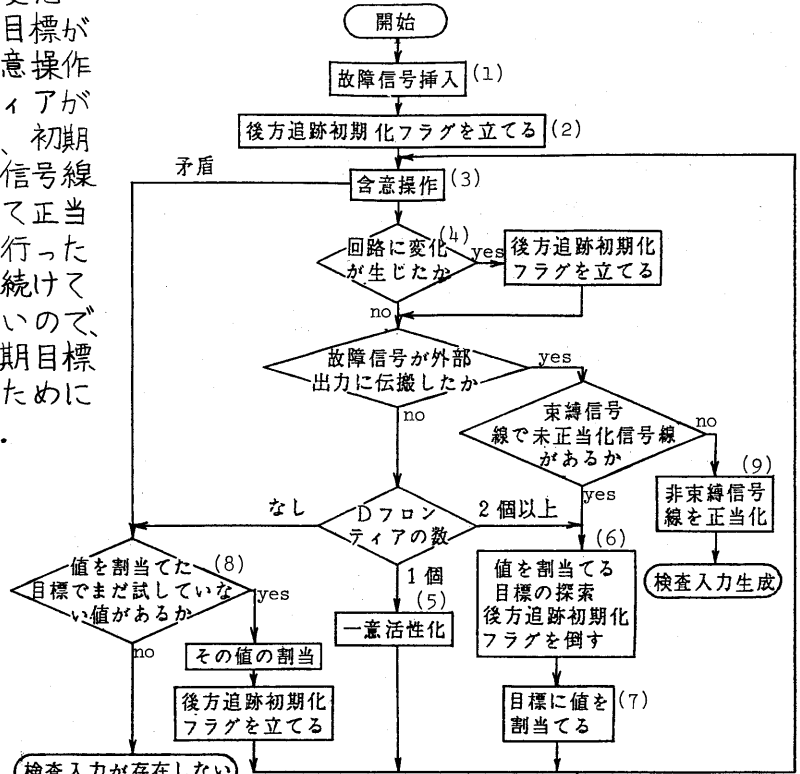


図8 FAN-III アルゴリズムの概要フローチャート

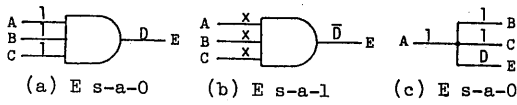


図9. 故障信号挿入

ままの状態で作保存されている。したがって、分岐目標がなくなると、回路に変化が生じて再び初期化フラグが立つまでは、後方追跡を行わず、先頭目標群から順次目標が取り出され値が割当てられていく。(7) 値の割当て

目標である先頭信号線または分岐点に目標値を割当てる。この時、この割当てに対応する節点を判定木に追加する。

(8) バックトラック

含意操作で矛盾が生じるか、Dフロンティアがなくなりかつどの外部出力にも故障信号が伝搬していない時に判定木をバックトラックする。

(9) 非束縛信号線の正当化

束縛信号線から成る部分回路の検査入力生成が終了すると、先頭信号線より入力側の非束縛信号線から成る樹木状回路の部分の値を決める。この部分の信号線の正当化はDアルゴリズムの正当化(一致操作)と同じ方法でバックトラックなしで行える。

4. 実行結果

FAN-Ⅲアルゴリズムのプログラム化を行い、EALG[9], PODEM[8], FAN[10]の各アルゴリズムと比較するため、数百~数千ゲート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2~4に示す。使用計算機は大阪大学大型計算機ACOS1000(処理速度15MIPS)で、使用言語はFORTRANである。

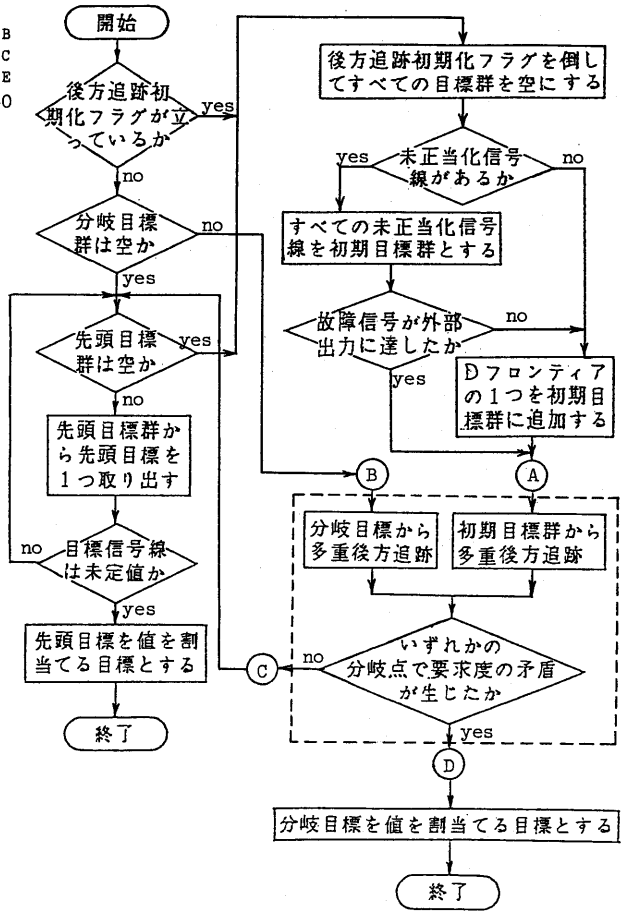


図10. 値を割当てる目標を求める手順

表2には回路の特性を、表3には各アルゴリズムの計算時間と平均バックトラック数、表4には検出率を示した。検査入力の生成は、各代表故障毎に一つの検査入力を生成するという方式で行った。計算時間の制限から、1故障当り100回を越えるバックトラックをする故障については計算を打ち切り、打ち切り故障とした。各アルゴリズムには発見的的手法[13]が採用されており、その方策を決める尺度としてGoldstein[14]の尺度を使用した。

表3から、FAN-Ⅲは打ち切り故障が少なく検出率の高い効率の良いアルゴリズムであることがわかる。また、PODEMに比べて冗長故障に対する打

切がきわめて少なく、解の不在の早期発見のための種々の手法の効果が現われている。実際FAN-Ⅲでは、冗長故障のほとんどはバックトラックなしに解の不在が判明している。

ここでは、故障シミュレーションとして同時シミュレーション[16]を採用し、先に示したFAN-Ⅲアルゴリズムと組合せて検査入力生成システムを作成したので、その性能を紹介する。方式としては、FAN-Ⅲアルゴリズムにより生成した検査入力に対して、

### 5. 同時シミュレーションとの結合

表2. 回路の特性

回路番号	型	ゲート数	信号線数	入力数	出力数	分岐点数	代表故障数
# 1	SELECTER	449	1407	130	106	215	1522
# 2	ECC	718	1925	33	25	381	1871
# 3	ALU	1003	2782	233	140	454	2747
# 4	ALU	1456	3572	50	22	579	3428
# 5	ALU	1537	4437	138	22	596	3674
# 6	64bit Adder	2002	5429	178	123	806	5350
# 7	ALU	2037	5920	220	145	902	5888
# 8	Carry Save Adder Tree	2238	6141	163	243	1081	6342
# 9	ALU	2982	7618	207	108	1300	7550

表3. 計算時間と平均バックトラック数

回路番号	計算時間(秒)				平均バックトラック数			
	EALG	PODEM	FAN	FAN-III	EALG	PODEM	FAN	FAN-III
# 1	8.5	4.2	4.9	4.5	0.16	0.05	0.01	0.01
# 2	57.2	21.5	22.3	23.9	5.06	0.51	0.16	0.25
# 3	103.3	57.1	33.0	30.9	6.54	6.57	3.24	1.73
# 4	311.5	129.9	98.5	84.7	32.09	8.64	2.62	0.56
# 5	269.5	118.4	102.8	81.0	11.27	8.12	3.94	0.58
# 6	229.7	90.4	73.0	72.6	0.49	1.78	0.27	0.21
# 7	381.3	174.2	121.6	138.6	4.80	3.29	0.58	1.72
# 8	350.4	105.7	103.6	103.6	5.39	1.03	0.64	1.00
# 9	872.7	526.8	300.0	281.8	4.78	7.78	4.94	3.46

表4. 検出率(%)

回路番号	EALG			PODEM			FAN			FAN-III		
	検出故障	打切故障		検出故障	打切故障	冗長故障	検出故障	打切故障	冗長故障	検出故障	打切故障	冗長故障
# 1	99.15	0.13		99.28	0	0.72	99.28	0	0.72	99.28	0	0.72
# 2	94.87	4.65		99.20	0.48	0.32	99.47	0.16	0.37	99.52	0.11	0.37
# 3	90.32	5.68		93.05	5.79	1.16	94.36	2.95	2.67	95.27	1.60	3.13
# 4	63.36	21.09		90.52	7.82	1.66	93.79	2.39	3.82	95.89	0.12	4.00
# 5	79.86	5.33		91.40	7.35	1.25	93.33	3.21	3.46	96.16	0.11	3.73
# 6	98.65	0.19		98.45	0.58	0.97	98.77	0.13	1.10	98.90	0	1.10
# 7	90.32	3.94		93.44	1.27	5.28	94.00	0.14	5.86	93.64	0.51	5.86
# 8	95.40	3.91		98.85	0.76	0.39	99.01	0.54	0.49	98.85	0.66	0.49
# 9	94.49	3.77		93.72	5.59	0.69	95.10	4.01	0.89	96.57	2.46	0.97

表5. FAN-IIIと同時シミュレーションとの結合

回路番号	計算時間(秒)			検出率(%)	打切故障率(%)
	FAN-III	同時シミュレーション	合計		
# 1	0.2	0.7	0.9	99.28	0
# 2	2.5	4.5	7.0	99.52	0.11
# 3	6.3	3.6	9.9	95.74	1.13
# 4	6.5	9.2	15.7	96.00	0
# 5	8.8	11.3	20.1	96.27	0
# 6	3.4	9.8	13.2	98.90	0
# 7	7.4	14.0	21.4	94.14	0
# 8	1.5	4.1	5.6	99.45	0.06
# 9	22.7	19.2	41.9	98.20	0.83

その未定入力値に乱数により値を設定して故障シミュレーションを行う方式を採用した。表2の回路に対する実行結果を表5に示す。

## 6. あとがき

組合せ回路の検査入力生成を高速化するための種々の手法について述べた。また、それらの手法を採用したFAN-IIIアルゴリズムを作成し、少ないバックトラック数で高検出率が得られることを示した。さらに同時シミュレーションと組合せることにより高性能の検査入力生成システムが作成できた。

## 文献

- [1] H.Fujiwara and S.Toida, "The complexity of fault detection: An approach to design for testability", Proc. FTCS-12, pp.101-108, June 1982.
- [2] F.F.Sellers, M.Y.Hsiao and L.W.Bearnson, "Analyzing errors with the Boolean difference", IEEE Trans. Comp., C-17, pp.676-683, July 1968.
- [3] K.Kinoshita, Y.Takamatsu and M.Shibata, "Test generation for combinational circuits by structure description functions", Proc. FTCS-10, Oct. 1980.
- [4] J.P.Roth, W.G.Bouricious and P.R.Schneider, "Programmed algorithms to compute tests to detect and distinguish between failures in logic circuits", IEEE Trans. Elec. Comp., EC-16, pp.567-580, Oct. 1967.
- [5] P.Muth, "A nine-valued circuit model for test generation", IEEE Trans. Comp., C-25, pp.630-636, June 1976.
- [6] C.W.Cha, W.E.Donath and F.Ozguner, "9-V algorithm for test pattern generation of combinational digital circuits", IEEE Trans. Comp., C-27, pp.193-200, Mar. 1978.
- [7] 村上, 平川, 菊地原, "LSI機能試験系列発生の一手法," 信学技報 半導体トランジスタ研究会, SSD80-13, 1980.
- [8] P.Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits", IEEE Trans. Comp., C-30, pp.215-222, Mar. 1981.
- [9] 高松, 藤原, 樹下, "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.
- [10] 藤原, 下野, 尾崎, "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 EC81-50, 1981.
- [11] 藤原, 下野, 尾崎, "組合せ回路における分岐指向型検査入力生成法," 情報処理, vol.24, No.1, 1983. (掲載予定)
- [12] 藤原, 下野, 磯部, 尾崎, "組合せ回路における分岐指向型検査入力生成法(2) - 多重後方追跡の適用 -, " 信学技報 EC82-6, 1982.
- [13] 下野, 藤原, 尾崎, "可検査性尺度に基づく検査入力生成効率改善法," 情報処理学会 電子装置設計技術研究会資料 13-4, 1982.
- [14] L.H.Goldstein, "Controllability / observability analysis of digital circuit", IEEE Trans. Circuits and Systems, CAS-26, pp.685-693, Sep. 1979.
- [15] 藤原, 尾崎, "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 EC80-38, 1980.
- [16] E.G.Ulrich and T.Baker, "The concurrent simulation of nearly identical digital networks", Proc. 10th Design automation Workshops, pp.145-150, June 1973.