

# TESTA: 順序回路のテスト系列自動生成システム

山田 輝彦 梶所 幹幸 加須屋 義弘  
(日電東芝情報システム(株) 研究所)

## 1. はじめに

近年の集積回路技術の進歩に伴う論理回路の大規模、複雑化は回路の故障検査を増々困難にしている。回路の高信頼性を実現するためには高品質の故障検査が不可欠であり、これまでに多数のテスト系列生成システムが開発された。これらのシステムでは、回路の各信号線の論理縮退故障について論理素子の接続関係を逐一追跡してテスト系列を生成する方法<sup>(1)(2)</sup> (ハードウェア的アプローチ)が採用されている。この方法は、論理素子の数および接続の複雑さ(例えば、フィードバック・ループの数)の増大に伴い計算の負担が急増するので、複雑で規模の大きな順序回路のテスト系列生成には適さない。

順序回路のテスト系列を生成する別の方法として回路の各状態と状態間の各遷移を逐一確認する方法<sup>(3)</sup>がある。この方法では100%の故障検出率をもつテスト系列が生成できるが、複雑で規模の大きな順序回路に対しては状態遷移表の作成が必ずしも容易ではなく、またテスト系列が非常に長くなることが実用上問題となっている。

筆者らは、先に順序機械の全ての状態遷移を尽すための入力系列が非常に高い故障検出率をもつことを示し、“状態遷移追跡法”と呼ぶテスト系列の生成法を提案した。<sup>(4)</sup>この方法はテスト系列の生成が容易でテスト系列長が比較的短いことにおいて上記の方法<sup>(3)</sup>より実用的ではあるが、複雑で規模の大きな順序回路に対しては依然として状態遷移表の作成とテスト系列長が問題となる。そこで、計算機の処理能力に応じた回路のモデル化と実用的長さのテスト系列の生成を可能にするために次のアプローチを採って実用化をはかった。<sup>(5)(6)</sup>

- ・被検査回路を幾つかの部分回路に分割し、もとの回路を部分回路の接続として表わす。次に、各部分回路の論理機能を表わす状態遷移表と出力表を求め部分回路を機能ブロック(順序機械)として表わすことにより、被検査回路を順序機械のネットワークとしてモデル化する。
- ・検査目標を信号線の単一論理縮退故障<sup>(4)</sup>に限定すれば必ずしも回路の全ての状態遷移を尽す必要はないので、ネットワークの各順序機械にできるだけ多くの新遷移が生じるようにテスト系列を生成する。

TESTA (Automatic Test Generation System Using State Diagram Tracing Approach) は、回路図に示される論理素子間の接続を回路情報とし、上記の方法に従って順序回路のテスト系列を自動生成するCADシステムである。本システムはACOSシステム700上にインプリメントされ、その実用性が確認された。本稿では、TESTAのメイン・モジュールである回路のモデル化プログラムとテスト系列生成プログラムの各処理手順を第2, 3節で概説し、さらに第4節でシステムの構成と評価結果を示す。

## 2. 論理回路のモデル化の手順

本システムでは、図1に示す処理によって順序回路が順序機械のネットワークとしてモデル化される。以下では、図1の各処理について述べる。

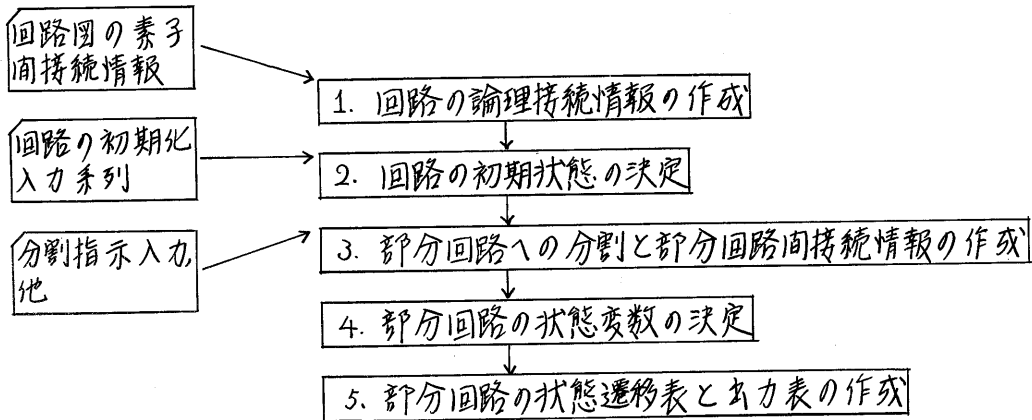


図1 回路のモデル化処理

### 2.1. 回路の論理接続情報の作成

回路図に示される論理素子および入、出力端子間の接続関係として与えられる回路情報から、論理シミュレーションに適した形式の論理接続テーブルを作成する。

現在使用可能な基本論理素子：NAND, AND, NOR, OR, EXNR, EXORの各論理ゲートと、SR, マスタ・スレーブJK, エッジトリガJKおよびDの各フリップ・フロップと遅延素子。

現在処理可能な回路規模：入力端子数+出力端子数 $\leq 114$ , クロック相数 $\leq 9$  (ただし、正負クロックの混合は不可), 回路構成要素数(基本論理素子および外部端子の合計) $\leq 1000$ 。

### 2.2 回路の初期状態の決定

順序回路では一般に電源投入直後の回路状態が不明であり、回路の使用にあたり所定の状態(以下では初期状態と言う)へのセット・アップが必要となる。このセット・アップは回路を任意の状態から初期状態へ遷移させ得る入力系列(以下では初期化入力系列と言う)を加えることにより実現される。ここでは、初期化入力系列を入力として論理シミュレータにより回路の初期状態における各論理素子の状態を求める。この論理素子の状態から2.5項で述べる部分回路の状態遷移表作成に必要な各部分回路の初期状態が決定される。

### 2.3 部分回路への分割と部分回路間の接続情報の作成

回路を部分回路  $N_1, N_2, \dots, N_k$  に分割し、各部分回路の論理接続情報(もとの回路と同一形式)と部分回路間の接続情報を作成する。これによって、もとの回路は部分回路  $N_1, N_2, \dots, N_k$  の接続として表現される。

ユーザはこの処理のために各部分回路  $N_i, i=1, 2, \dots, k$ , の出力点として外部出力端子および論理素子の集合  $\Sigma_i, i=1, 2, \dots, k$ , を指定する。システムは、この指定に基づいて、以下のチェックを行う。

- (i) 同一の出力点が2つ以上の  $\Sigma_i$  に含まれていないこと。
- (ii) 全ての外部出力端子が出力点に指定されていること。

続いて、以下の処理を行う。

- (i) 出力点に指定された各論理素子の出力に因るに例示するように一対の擬似出力端子と擬似入力端子を挿入して回路を分割する。
- (ii) 集合  $\Sigma_i, i=1, 2, \dots, k$ , の各要素に対応する外部出力端子と擬似出力端子の各々から外部入力端子または擬似入力端子に至るまでの論理素子の接続関係を回路の入力側に向かって追跡する。この追跡によって到達した外部入力端子および擬似入力端子を部分回路  $N_i$  の入力端子とする。さらに、追跡した各論理素子の接続関係から部分回路  $N_i$  の論理接続情報を作成する。
- (iii) 回路を分割するために挿入した擬似出力端子と擬似入力端子の対関係から部分回路  $N_1, N_2, \dots, N_k$  の接続情報を作成する。

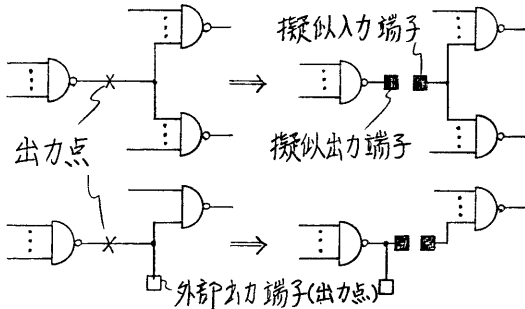


図2 出力点への擬似端子の挿入

・回路の分割における注意事項

- (i) 部分回路の最大規模 (2.5項の状態遷移表と出力表の作成に要する計算時間とメモリ容量の制限による)  
入力端子数 + 状態変数の数  $\leq 18$ , 出力端子数  $\leq 18$ , 回路構成要素数  $\leq 100$ .
- (ii) 1つの非同期フィードバック・ループを構成する全ての論理ゲートは必ず同一の部分回路に属すること。
- (iii) 回路の分割は、可能な限りいかなる大きさの伝搬遅延を挿入しても回路の動作に影響を与えない信号線 (例えば、外部クロックで制御されるフリップ・フロップの出力) で行うこと。
- (iv) 部分回路の数および部分回路の直列接続の度合を可能な限り小さくすること。

## 2.4 部分回路の状態変数の決定

ここでは、各部分回路の内部状態を代表する信号値を表わす変数 (状態変数) の決定を行う。

- (i) 回路が論理ゲートだけで構成されている場合: 回路に存在するフィードバック・ループを検出し、各フィードバック・ループを構成する論理ゲートを少なくとも1つ含む論理ゲートの極小集合を求める。この集合に含まれる論理ゲートの

出力を状態変数  $y_1, y_2, \dots, y_m$  とすれば、 $(y_1, y_2, \dots, y_m)$  によって回路が安定状態にあるときの内部状態が表現できる。(10)

(ii) 回路がフリップ・フロップを含む場合：本システムでは、各フリップ・フロップを閉回路の例示するようにモデル化している。このように各フリップ・フロップに状態変数  $Y$  および  $Q$  (ただし、SRフリップ・フロップでは  $Q$  だけである) を設けることによりフリップ・フロップに係わる全てのフィードバック・ループの状態をこの  $Y$  と  $Q$  で代表できる。したがって、まず各フリップ・フロップに状態変数  $Y$  および  $Q$  を設けてその出力線を切断した後、(i)の方法に従ってその他の論理ゲートで構成されるフィードバック・ループを代表する状態変数を決定すればよい。

(iii) 回路が遅延素子を含む場合：図4は遅延素子の出力を状態変数にしなければならぬ回路例であるが、一般に遅延素子の出力を状態変数にしなければならぬか否かはその用途に依存する。しかしながら、回路の論理接続情報だけからその用途を自動識別することは困難であるため、本システムでは各遅延素子の出力を状態変数にするか否かをユーザが指定することになっている。状態変数に指定された遅延素子がある場合にはその出力で回路を切断した後、(i) または (ii) に従って残りの状態変数を決定すればよい。

上記に従って決定された状態変数に基づいて各部分回路を次のように分類する。

タイプ0：状態変数の無い部分回路。

タイプ1：状態変数がフリップ・フロップの  $Y$  と  $Q$  だけで、かつ各フリップ・フロップの  $C$  入力が入部クロックに接続されている部分回路。

タイプ2：上記以外の部分回路。

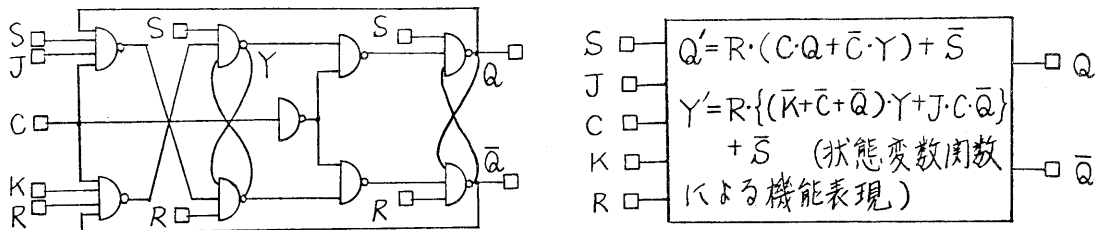


図3 JKフリップ・フロップの非同期論理モデル

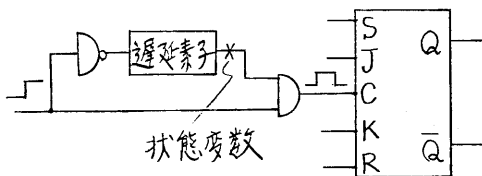


図4 パルス発生部をもつ順序回路

### 2.5 部分回路の状態遷移表と出力表の作成

ここでは、各部分回路の論理機能を入力変数の2値ベクトル  $X$  と前項で求めた状態変数の2値ベクトル  $Y$  と  $K$  に関して状態遷移表と出力表で表現する。

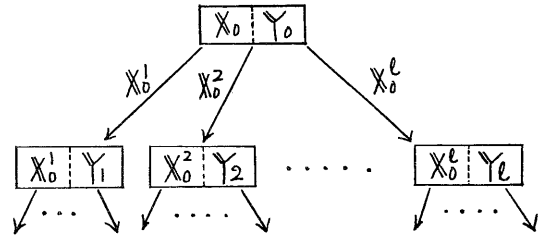
・状態遷移表の作成：2項で求めた回路の初期状態における各論理素子の状態から部分回路の初期状態  $(X_0, Y_0)$  を決定し、この  $(X_0, Y_0)$  から図5に示すように単一入力変化の下で順次安定状態間の遷移をたどって状態遷移表を作成する。尚、ここではクロック信号も1つの入力信号とみなしている。

状態遷移表の作成において単一入力変化の下での遷移に限定した理由：

(i) 同期順序回路（タイプ1の部分回路に相当する）に対しては、クロック信号を1つの入力とみなせば単一入力変化を考慮するだけで十分である。

(ii) 非同期順序回路（タイプ2の部分回路に相当する）の多くは単一入力変化の条件下で設計されている。

(iii) 入力変数が多くなると多入力変化まで考慮した状態遷移表を作成するには膨大な計算時間が必要となる。



ただし、 $X_0 = (a_1, a_2, \dots, a_l)$ ,  $X_0^1 = (\bar{a}_1, a_2, \dots, a_l)$ ,  
 $X_0^2 = (a_1, \bar{a}_2, \dots, a_l)$ ,  $\dots$ ,  $X_0^l = (a_1, a_2, \dots, \bar{a}_l)$

図5 状態遷移表の作成

本システムは状態遷移表の作成に状態変数関数（論理式）を用いる方法と論理シミュレータ（1ユニット遅延、非同期イベント、入力並列方式）を用いる方法を備えている。状態変数関数による方法は論理シミュレータによる方法に較べて計算時間を大幅に短縮できるが、論理素子の遅延を考慮しなければならない回路に適用することはできない。どちらの方法を用いるかは各部分回路毎にユーザが指定することになっている。

尚、状態遷移表の作成において次の条件下での遷移を禁止とする。

- (i) フリップ・フロップのマスタ・セットとリセット入力が共にオン状態となる。
- (ii) 多相クロックの回路で2つ以上のクロック信号がオン状態となる。
- (iii) クロック信号が1つでもオン状態のときに外部入力（ただし、擬似入力はこの限りでない）が変化ある。
- (iv) 論理素子の遅延に起因するスパイク、振動、ハザードが発生する。（この条件は論理シミュレータを用いる場合にだけチェックする）

### ・出力表の作成

部分回路の各出力  $Z_j, j=1, 2, \dots, m$  の出力表は入力ベクトル  $X$  と状態ベクトル  $Y$  に関する出力関数  $Z_j = f_j(X, Y)$  より作成される。

## 3. テスト系列生成の手順

前節の手順により作成された順序機械のネットワークに対して各順序機械にできるだけ多くの新遷移が生じるようにテスト系列を生成する。本システムでは、以下に述べる発見的な手法を用いてネットワークの現在の状態から次の入力を決定することにより実用的な計算時間とメモリ容量でのテスト系列の生成を実現している。

テスト系列の生成にあたり、まず各順序機械の入力変化を次のように制限する。

- (i) 単一入力変化は常に許容する。
  - (ii) 多入力変化は状態の遷移先が入力の変化の順序に依存しない場合にだけ許容する。(タイプ2の部分回路の入力変化がこれによって制限を受ける)
- このため、ネットワークの入力も単一入力変化が基本となりネットワークの現在の入力(ただし、クロック入力を除く)が  $I_a = (a_1, a_2, \dots, a_t)$  であれば  $I_{a'} = (a_1, a_2, \dots, a_{i-1}, \bar{a}_i, a_{i+1}, \dots, a_t)$ ,  $i=1, 2, \dots, t$ , のいずれかを次の入力とする。ただし、クロック入力がある場合は  $I_a$  も次入力の候補とし、また各入力毎に所定のクロック信号を加える。

・次入力の決定法

所望の故障検出率(単一論理縮退故障の90%以上)を達成できれば、テスト系列は短い程望ましい。本システムでは、次の点を考慮することによってテスト系列長の短縮がはかれると考え図6に示す次入力の決定法を採用した。

- (i) 早い時期にできるだけ多くの新遷移を各順序機械に生じさせる。
- (ii) 各順序機械の状態遷移の影響をできるだけ早くネットワークの出力に伝搬させる。

・テスト系列の構成

図6の方法で生成される入力系列によって各順序機械に生じる新遷移を未処理新遷移として記憶しておき、この情報をもとにテスト系列は図7のようにサブ・シーケンスを最小単位として構成される。サブ・シーケンスの区切り(図中、 $I_p, I_r$ )は次の2つの条件によって判定される。

- (i) 入力  $I_r$  を加えた時点でネットワークの外部出力端子に持続される順序機械の少なくとも1つに出力または状態の変化が生じる。

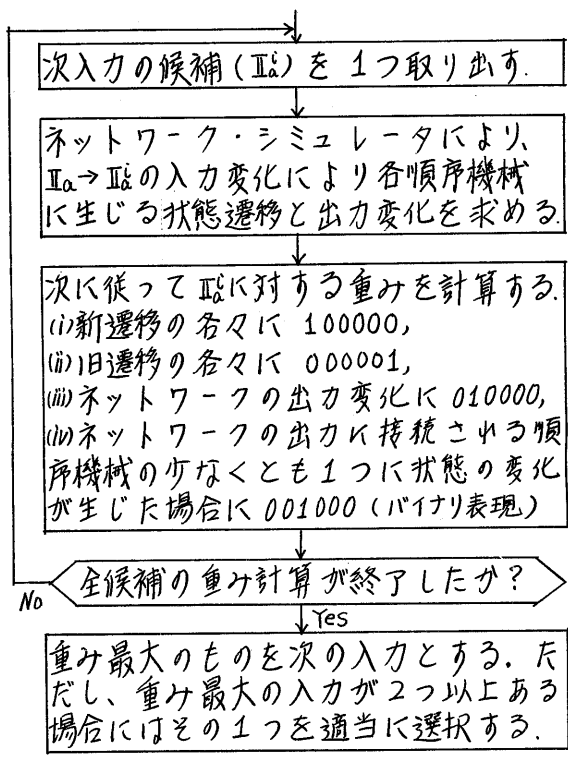


図6 次入力の決定法

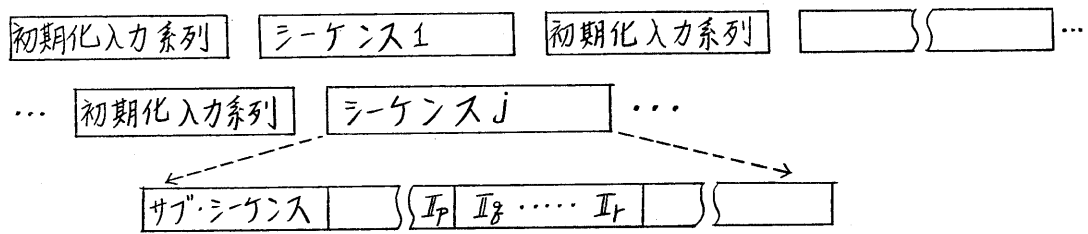


図7 テスト系列の構成

(ii) 条件(i)に寄与したとみなせる未処理新遷移が少なくとも1つ存在する。  
 この2つの条件が満たされたとき入力 $U_1$ から $U_4$ までを1つのサブ・シーケンスとし、また条件(iii)の未処理新遷移を有効遷移とする。尚、サブ・シーケンスの最大長 $l_{max}$ は制御入力の1つとしてユーザが指定することになっている。

図8において、 $A_1, A_2, \dots, A_4$ の各時点でサブ・シーケンスが構成され各順序機械に生じた新遷移が $\sqrt{\quad}$ で示す間隔で有効遷移になったとする。この後に生成された長さ $l_{max}$ (サブ・シーケンスの最大長)の入力系列( $A_4 \sim A_e$ )が上記のサブ・シーケンスの条件を満足しなかった場合、 $A_4$ が図7に示したシーケンスの区切りとなる。このとき、//////を付した部分で各順序機械に生じた新遷移を全て無効とし(新しいシーケンスの生成において再度新遷移として取扱う)、ネットワークを初期状態にもどして次のシーケンスの生成をはじめ。

上記のテスト系列の構成法も順序機械1  
 テスト系列の短縮をはかるため  
 に採用したものである。  
 その理由は:

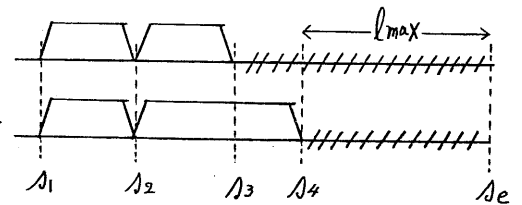


図8 シーケンスの区切り

(i) 新遷移を伴わない入力系列では故障検出率の増加が期待できない。したがって、ネットワークが旧遷移を繰返す状態に陥ったとき、ネットワークを初期状態にもどして別のルートで新遷移を捜すことが望ましい。

(ii) 初期化入力系列は一般に正常回路に固有のものであり、故障回路では任意の状態から初期状態へのセット・アップが不可能となる場合が多いので初期化入力系列は優れた故障検出能力をもつ。したがって、テスト系列に適度に初期化入力系列を挿入することにより故障検出率の増加が期待できる。

従来よりテスト系列の品質は被検査論理回路における信号線の単一論理縮退故障の検出率により判定されており、故障検出率70%以上が一応の目標となっている。そこで、本システムでは図9のアプローチを採ってテスト系列生成の効率化をはかっている。尚、 $Y = (\text{各順序機械における有効遷移数の合計}) / (\text{各順序機械における全遷移数の合計})$ 、 $L$ および $Y$ の値は制御入力の1つとしてユーザが指定することになっている。

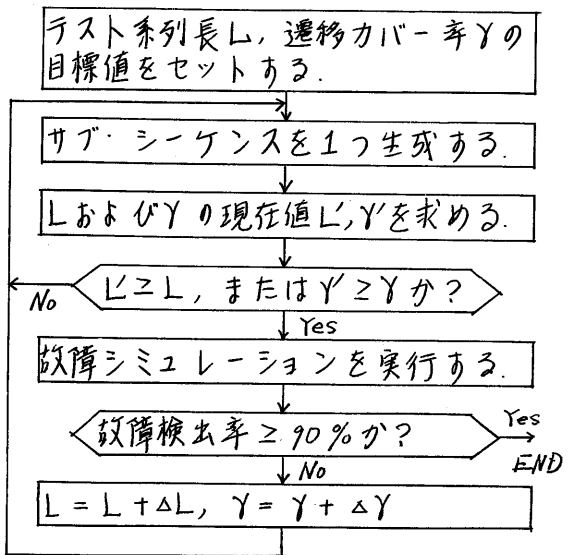


図9 テスト系列の生成

#### 4. システムの構成と評価結果

ここでは、第5および3節で述べた手順を基にしてACOSシステム700上にインプリメントしたTESTAのプログラム構成について述べ、システムの実行例と評価結果を示す。

TESTAは次の5つのプログラムより構成され、プログラム規模はACOS-6のアセンブラ約13Kステップ、フォートラン約7Kステップである。

回路のモデル化プログラム (NETMOD) : 被検査回路を幾つかの部分回路に分割し、もとの回路を部分回路の接続として表わす。次に、各部分回路の論理機能を表わす状態遷移表と出力表を求め部分回路を機能ブロック (順序機械) として表わすことにより、被検査回路を順序機械のネットワークとしてモデル化するプログラム。

状態遷移表の前処理プログラム (PREPRO) : 順序機械の各遷移条件に故障検出に対する有効性に基づき優先度を付すプログラム。この優先度は次入力決定ルーチンの重み計算のところで考慮される。尚、この処理はユーザ・オプションとなっている。

テスト系列生成プログラム (PATGEN) : 順序機械のネットワークに対し、各順序機械の状態遷移を追跡しながらテスト系列を生成するプログラム。

テスト系列編集プログラム (SEQEDT) : PATGENで生成されたテスト系列のフォーマット変換、マニュアルで作成したテスト・パターンの追加、PEVALSの出力結果に基づき故障検出に寄与していない部分系列の削除を行うプログラム。

テスト系列の評価サポート・プログラム (PEVALS) : 故障シミュレーション結果とテスト系列生成に伴うネットワーク動作の詳細から、各部分回路におけるテスト系列長と故障検出率、遷移カバー率、状態カバー率の関係、未検出故障等を調べるプログラム。

現在のTESTAは、回路図に示される論理素子および入、出力端子の接続関係として与えられる回路情報から論理接続テーブルを作成するため論理シミュレーション・システムLOGOS2の回路記述プログラムを、またPATGENで生成されたテスト系列の故障検出率の評価に故障シミュレーション・システムFOCUSを使用している。上記のプログラムとLOGOS2およびFOCUSを含めた処理フローを図10に示す。

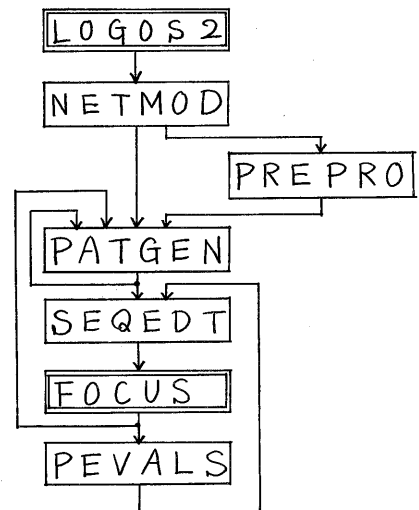
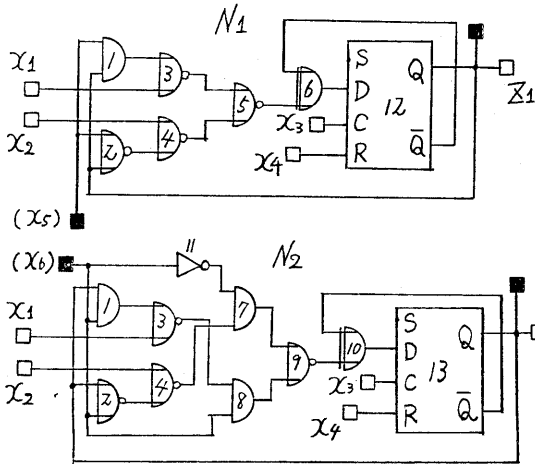
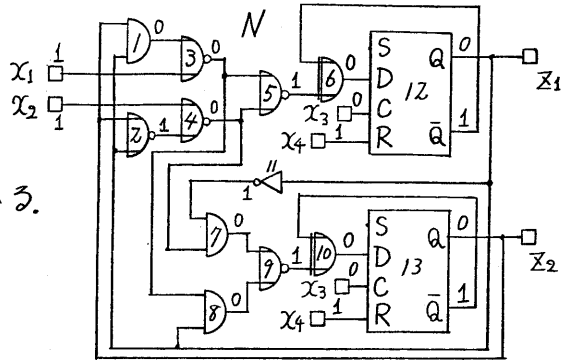


図10 TESTAの処理フロー



・TESTAの実行例

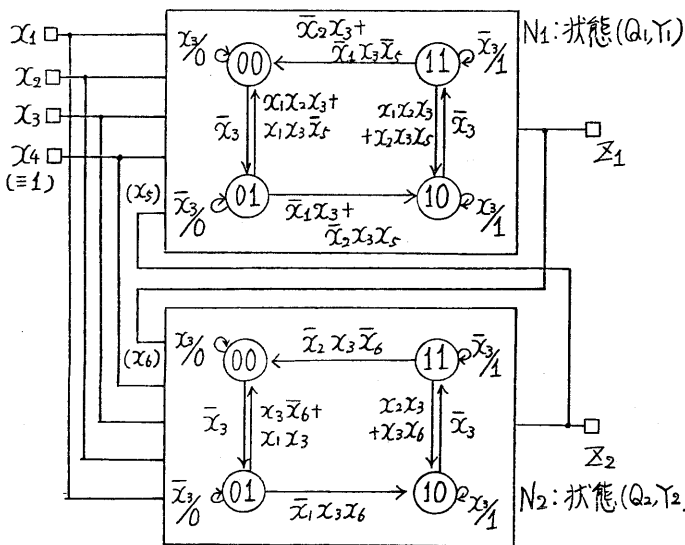
- (i) 被検査回路Nを初期化入力系列  $x_1 x_2 x_3 x_4 : 1100 \rightarrow 1101$  によって図のようにセット・アップする。尚、回路のセット・アップにだけ使用するリセット入力  $x_4$  はここで論理1に固定される。
- (ii) 回路Nを  $\Sigma_1 = (Z_1, 12 \# Q)$ ,  $\Sigma_2 = (Z_2, 13 \# Q)$  を出力点とする部分回路  $N_1, N_2$  に分割する。



(iv)  $l_{max} = 10, \gamma = 0.95, L = 50$  の下で次のテスト系列が生成され、その故障検出率は94.4%であった。

尚、図において X : dont care, クロック信号  $x_3$  の記述は省略されているが実際には各入力パターン毎に  $0 \rightarrow 1 \rightarrow 0$  なるクロック信号が加えられる。

(iii) 部分回路  $N_1$  および  $N_2$  の状態変数はそれぞれフリップ・フロップ12, 13に設定されるYとQだけであり、回路Nは次のようにモデル化される。ここでは、状態遷移表と出力表を合せてMealyの状態線図で表現する。



X	X	Z	Z		
1	2	4	1	2	
0	0	0	x	x	← 初期化入力系列
0	0	1	0	0	
0	1	1	1	0	
0	1	1	0	1	
0	1	1	1	1	
1	1	1	1	1	
1	0	1	0	1	
1	0	1	1	0	
1	0	1	1	0	
1	0	1	0	0	
0	0	1	1	0	← 初期化入力系列
0	0	0	x	x	
0	0	1	0	0	
0	1	1	1	0	
0	1	1	1	0	
0	0	1	0	1	
0	0	1	0	0	
0	0	1	0	0	
1	0	1	0	0	
0	0	1	0	0	
0	1	1	0	1	← 初期化入力系列
1	1	1	0	1	
0	1	1	1	1	
1	1	1	1	1	
0	1	1	1	1	
0	0	1	0	1	
0	0	1	0	1	
0	0	1	0	1	
0	0	1	0	1	
0	0	1	0	1	

## ・評価結果

回路名	A	B	C
入力端子数	77	21	14
出力端子数	34	10	3
論理ゲート数	255	136	113
フリップ <sup>0</sup> ・フロップ <sup>0</sup> 数	6	36	45
テスト系列長	645	2068	867
故障数	585	417	635
故障検出率(%)	94.0	92.6	93.5
CPUタイム	10'05"	10'10"	18'53"
コア容量(K.ワード)	127	127	127

状態遷移に着目した発見的手法によりテスト系列の生成をはかっている。この方法は回路の機能検査の観点からテスト系列の生成を試みたものであり、中規模順序回路に対するシステムの評価結果によりこの方法の実用性が確認された。今後回路が増々大規模、複雑化するに伴い従来のハードウェア的アプローチに変わり高機能レベルでの回路記述と機能検査の観点からのテスト系列の生成が必要となろう。

〔謝辞〕本研究を進めるにあたり色々と御助言いただいた日本電気(株)、中央研究所周辺機器研究部の萩原部長、森田課長、当社、第三研究室の山田分室長、菊池分室長、南谷主任に感謝します。

### 〔参考文献〕

- (1) J. P. Roth, "Diagnosis of automata failures: a calculus and a method," IBM J. R&D, vol. 10, pp. 278-291, July 1966.
- (2) H. Kubo, "A procedure for generating test sequences to detect sequential circuit failures," NEC R&D, No. 12, pp. 69-78, Oct. 1968.
- (3) S. G. Chappell, "LAMP: automatic test generation for asynchronous digital circuits," B.S.T.J., vol. 53, pp. 1477-1503, Oct. 1974.
- (4) D. M. Schuler, E. G. Ulrich, T. E. Baker and S. P. Bryant, "Random test generator using concurrent logic simulation," Proc. 12th DA Conf. pp. 261-267, June 1975.
- (5) F. C. Hemme, "Fault detecting experiments for sequential circuits," Proc. 5th Ann. Symp. on Switching Theory and Logic Design, pp. 95-110, Nov. 1964.
- (6) 山田, 内藤, "順序回路のテスト系列を求めろー手法," 信学技報 R76-14, 1976, 7月.
- (7) —, "テスト系列生成システムにおける順序回路のモデル化プログラム," 昭53, 信学全大 2-123.
- (8) —, "遷移路追跡法による順序回路のテスト生成, テスト系列短縮のー考察," 昭53, 信学全大 2-124.
- (9) —, "Test generation method for highly sequential circuits," Proc. of COMPCON '79 Spring, pp. 104-107, Feb. 1979.
- (10) S. H. Unger, Asynchronous sequential switching circuits, pp. 194-197, Wiley-Interscience, New York, 1969.

## 5. おまけ

順序回路のテスト系列自動生成システム TEST A のメイン・モジュールである回路のモデル化プログラムとテスト系列生成プログラムにおける各処理手順を概説し、プログラム・システムの構成を示した。本システムでは、複雑で規模の大きな順序回路に対処するために回路を順序機械のネットワークとしてモデル化し、順序機械の