UDC 681.32.08

ディジタル・システムの自動故障診断

Automatic Diagnosis of Digital System

渡 辺　勝*・杉 本 正 勝*
Masaru WATANABE and Masakatsu SUGIMOTO

現在各所で実現されつつある実時間電子計算機システムにおいては，信頼度が大であるという能力の他に故障が生じても短時間で復旧し得る能力（保守度）が必要である．ここでは，電子計算機システムなどのディジタル・システムの処理装置（PU）の自動故障診断に関して述べる．特に診断用のテスト・パターンを作成するアルゴリズム，実際にテスト・パターンを作成した例および自動故障診断における問題点に関して報告する．

1. まえがき

現在，多くの分野で大型・高速のディジタル・システム（電子計算機システム，電子交換システム，データ伝送システムなど）が使用されているが，システムが大型になり実時間使用されるに従ってシステムの信頼度およびシステムの診断可能性などが大きな問題となってきた．

従来，ハード・ウェア個々のものについての信頼度に関する考慮はもちろんなされてきた．しかし，動作時においてシステムの故障の徴候が検知されても，その故障点を指摘するには，多くの場合保守員の経験によっていた傾向が強く，保守および修理に比較的長時間を要していた．しかし，現在各所で実用化されている実時間電子計算機システムにおいては，信頼度が大であるという条件の他に故障が生じても短時間で復旧しうる能力（保守度）が必能である．

ここではディジタル・システムの処理装置 PU（Processing Unit）を構成している論理回路の故障診断を行なうときに必要なテスト・パターンを作成するアルゴリズムとこのアルゴリズムを実現する目的のプログラムを中心にして述べる．

2. 診断サブ・システム[12)17)]

ディジタル・システムの自動故障診断を実行するにはシステムがどのような条件を有すべきかについて考えてみる．

あるシステム S は，S が n 個の排他的サブ・システム $S_1, S_2, \cdots\cdots, S_n$ ($S=S_1 \oplus S_2 \oplus \cdots\cdots \oplus S_n$) に分割でき，次の3つの条件が満されている場合，システムを自動診断可能であるという．

（a） $S_i (i=1\sim n)$ のサブ・システムの診断に必要な入力および入力順列が求まっていること．

（b） S_i の中に次に述べる（1）～（5）のすべての操作を実行可能なサブ・システム C_{DS} が存在すること．

（1） C_{DS} を除くシステム S_i の選択された入力に入力テスト・パターンを加える．

（2） S_i の出力を得て，すでに計算によって求められている規準の出力パターンと比較する．

（3）（2）の動作の結果に従って診断順序の適当な場所にジャンプする．

（4） 診断の結果を人間に知らせるか，またはシステムが自動的に故障を修理しうる能力を有する（自動修理，自動保守）．

（5）（1）～（5）の操作の実行順序の制御を行なう

（c） システムの単一故障を扱う場合には，少なくとも2つの C_{DS} に属するサブ・システムが存在すること．（C_{DS} を診断サブ・システムと呼ぶ．）

診断サブ・システムを実際に設計するときに注意すべき事項をここで整理してみる．

（a） 故障検出（Failure-Detection）用の必要最小数のテスト・パターンを作成する有効なアルゴリズムが存在すること．

（b） 故障点指摘（Failure-Location）用の必要最小数のテスト・パターンを作成する有効なアルゴリズムが存在すること．

（c） （a），（b）について診断が完備であること．すなわちシステム内に診断の考慮がなされていない部分が存在しないこと．

（d） 間欠的に生ずる故障を検知する方法が存在すること．

（e） 診断サブ・システムのハード・ウェアのコストおよび診断用テスト・パターンを作成するコストを最小化する．

（f） 診断の対象となるハード・ウェアの設計変更に対して，すみやかに修正されたテスト・パターンを作成しうること．

これらの事項のうち，特に（a），（b），（c）の事項につ

* 東京大学生産技術研究所　第3部

いて以後検討していく．

3. 故障診断指数[17]

ここで，ディジタル・システムの有している故障診断の能力を客観的に数値で表現する目的で使用される故障診断指数について説明する．故障点指摘の能力をまず考慮する．$T=\{T_1, T_2, T_3, \cdots\cdots, T_n\}$ を対象する回路（論理回路に限らず一般のシステムを対象としても以下に述べることが意味をもつ）に対するテスト・パターンの集合，$F=\{F_1, F_2, F_3, \cdots\cdots, F_m\}$ を対象とする故障の集合とする．また，'module' とは故障状態にあるとき，修理またはおき換えをほどこされる最小の"回路の部分集合"と定義し，k_i を故障が F_i だと判明したときに，"故障している疑いのある" module の数とする．($1/k_i$ は F_i なる故障が修理される場合の能率を表現する量である．) 最後に，Nを回路中にある全 module 数，P_i を故障 F_i の出現確率とする．(ある時点で単一故障が生じているときには $\sum_{i=1}^{m} P_i=1$ が成立する．)

そこで，故障点指摘指数(分解能) $R_{fl}=1/\sum_{i=1}^{m} k_iP_i$……(1) また故障点指摘システムの有効度 $E_{fl}=R_{fl}/$(故障点指摘システムの全費用)……(2) を定義する．R_{fl} の値は $1/N\leq R_{fl}\leq 1/\sum_{i=1}^{m} P_i=1$ なる範囲にある．また，すべての module が同一の故障確率をとる場合には $P_i=k_i/\sum_{i=1}^{m} k_i$，よって $R_{fl}=\sum_{i=1}^{m} k_i/\sum_{i=1}^{m} k_i^2$……(3) となる．

次に故障検出の能力について考慮してみる．故障検出の能力は生じうる故障の総数mに対して，どれだけの故障が検出しうるかで表現されるので k を検出しうる故障の数として $F_{i1}, F_{i2}, F_{i3}, \cdots\cdots, F_{ik}$ が検出しうる場合に，故障検出指数を $R_{fd}=\sum_{i=1}^{k} P_{ii}$……(4) また，故障検出システムの有効度を $E_{fd}=R_{fd}/$(故障検出システムの全費用)……(5) と定義する．以上に定義した故障診断指数は故障診断の能力を良く表現すると思われ，今後，ディジタル・システムの設計に際して重要な意味をもつ指数である．

4. 論理回路の故障診断用のテスト・パターンの作成法の紹介

論理回路の故障診断用のテスト・パターンを作成する目的には，以下に説明する4つの方法がある．

a. 真理値表法

Gをある論理回路の機能を表現する関数，Fをその回路の特定故障回路を表現する関数とする．GおよびFの真理値表について調べれば，この特定の故障の有無を調らべるテスト・パターンを抽出しうる．そこで，この回路中に発生するすべての故障に対する F を求め，テスト・パターン集合を求め，その後にこの集合より必要最小数のパターンを抽出する方法が真理値表法である．

この方法は最も単純ではあるが，入力変数（入力端子数）の多い論理回路では，真理値表を記憶する記憶場所が大となる欠点を有している．しかし，入力数が 2～6 の回路について手計算でテスト・パターンを作成するには有効な方法である．

b. On, Off アレー法[1)2)3)19)20)]

この方法も，真理値表法と同様に回路出力が "1" となる入力集合（On アレー）を基本とするのであるが，この入力集合を回路の正規表示式の計算によって求めるので，真理値表法より入力集合を記憶する記憶容量が小で良い．この方法も入力変数の数が 2～6 程度の回路を扱うのに適した方法である．この方法には次に述べるライン切断法と呼ぶ改良法がある．

1) ライン切断法

この方法では回路出力が "1" となる入力集合を求める目的で正規表示式を計算するときに，すべての回路要素について計算するのではなく，回路の出力側からさかのぼって故障を仮定している回路のところでやめる．すなわち故障回路要素の出力部に対応するラインを切断し，縮少した論理回路に対し擬似入力パターンを作成し，その後に，切断したラインをつなぎ，擬似入力パターンより正規表示式の計算によって回路本来の入力パターンに変換する．

c. トレーシング法（論理回路のシミュレータを用いる法）

この方法はある入力パターンを仮定し，この入力のもとで，生じうる故障回路の出力値を求め，正常な回路がとるべき出力と比較する．異なっている場合には，この入力パターンによって故障が検出しうることを意味する．次に，この入力パターンで検出しえない故障集合のみを考え，再び適当な入力を仮定して以上に述べた過程を繰り返す．この方法は Gedanken-experiment または Black-box-philosophy による方法とも呼ばれる．

1) トレーシング法 I[4)5)6)8)9)]

以上の説明は原理的なもので，トレーシング法を実際に採用する場合は次のような考え方をとる．ある故障を仮定し，この故障点で擬似入力・パターンを仮定し，この入力のもとで回路の出力が正常な場合と故障が生じた場合とで異なるかどうかを調べる．出力が異なるパターンが存在したら，このパターンを正規表示式の計算によって実際の入力パターンに変換する．

2) トレーシング法 II[14)15)18)]

トレーシング法 I では一つの擬似入力に対して必ず回路の最終出力を求め，その後に正常な回路の回路出力と比較したが，実は出力パターンを求めなくとも最終出力を求める中間段階で，その擬似入力パターンのもとでは故障回路も正常な回路も同一の出力を出すことが判明する場合がある．トレーシング法 II では擬似入力パターンに対する回路応答のシミュレーションの際に常にそのパ

ターンのテスト入力パターンとしての有効性をチェックしてゆき，有効でないことが判明したらただちに，回路の最終出力を求める過程を打ち切り，次に新らたな擬似入力パターンを仮定し再び以上の過程をくり返す．この方法は次に述べるD-アルゴリズムへと拡張することができる．

d. D-アルゴリズム[14)15)]

この方法は，On, Off アレー法におけるライン切断法の考え方とトレーシング法Ⅱの考え方を拡張したものである．この方法には，回路出力が“1”になるような入力集合を求めるという過程はない．

まず，故障を仮定した回路要素の出力ラインの切断によって縮少された論理回路を対象として，故障回路要素に対応した入力に“D”を割り当てる．その後は，入力側から出力側に向けて各回路要素の出力ラインの情報として“1”,“0”,“X”,“D”,“$\overline{D}$”を一定の規則に従って割り当てる．この割り当てによる回路出力パターンが“D”または“$\overline{D}$”を含むときには，この割り当てによる回路情報パターンはテスト・パターンとなりうる．そこで回路情報パターンより実際のテスト・パターンへと正規表示式の計算によって変換を行なう（この操作を特に無矛盾性テスト操作と呼ぶ）．

e. On, Off アレー法を順序回路に適用した例

ここで，On, Off アレー法を使用して順序回路用のテスト・パターンを作成する手順について説明を加える．

診断用のテスト・パターンを作成するときには，順序回路中のフィード・バック・ループを切断し，切断したラインの一方を擬似入力，他方を擬似出力ラインと考えることによってこの順序回路を，ある時点において組合せ回路として考え，組合せ回路の故障診断用のテスト・パターン作成法を使用する．仮定として，この順序回路において現在の状態に関係なく，あるリセット入力のもとで一定の初期状態を有するか，または順序回路の現在の状態がテスト・パターンを加える診断サブ・システムにとって accessible であるとする．図1の回路においては，$(1, 2, 3, 4)=(0, 0, 0, X)$ または $(0, 0, X, 0)$ がリセット入力で，初期状態は $(9, 10)=(0, 0)$ である．

故障の種類はあるラインが s−t−1 または s−t−0 になる単一故障とする．s−t−1，s−t−0 とは回路要素の出力が，なんらかの原因で常に“1”，“0”となる故障

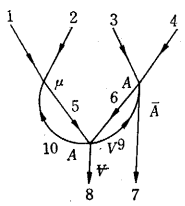

図 1　順序回路の例（μ は多数決論理回路要素を示す）

を意味する．

以下に $GB_i^{\alpha}(\beta, \bar{\beta})$ なるテスト入力集合，すなわちライン i が s−t−α になったとき，正常な回路 (Good Machine) は出力 β，故障回路 (Bad Machine) は出力 $\bar{\beta}$ となる入力集合を求める手順を示す．

第1ステップ：順序回路のフィード・バック・ループを切断し，擬似入力，擬似出力ラインを決定する．擬似出力を含めた全出力について Good Machine および第 i ラインの s−t−α なる故障を仮定した Bad Machine の On, Off アレーを求める．

第2ステップ：Good Machine のアレー A と C_{σ} cube との入力集合積をとる．C_{σ} は Good Machine の初期状態を表現しており initial cube と呼ぶ．この入力集合積を A^{σ} とする．同様に Bad Machine の初期状態 τ に対して，B^{τ} を求める．A^{τ} なる入力集合のうち，On アレーに属するものを A_1^{τ}，Off アレーに属するものを A_0^{τ} と表わす．B_1^{τ}, B_0^{τ} も同様である．

第3ステップ：アレー A_1^{σ} のうち擬似入力座標を削除して $\widehat{A_1^{\sigma}}$ とする．同様に $\widehat{A_0^{\sigma}}$, $\widehat{B_1^{\tau}}$, $\widehat{B_0^{\tau}}$ を求める．

第4ステップ：$D^1AB=(\widehat{A_1^{\sigma}}\cap\widehat{B_0^{\tau}})\cup(\widehat{A_0^{\sigma}}\cap\widehat{B_1^{\tau}})$ を計算する．D^1AB は Good Machine は σ，Bad Machine は τ なる初期状態より動作するとき，Good Machine と Bad Machine とを識別する長さ1の入力集合で，$GB_i^{\alpha}(1, 0)\cup GB_i^{\alpha}(0, 1)$ を意味する．

第5ステップ：$D^1AB=\phi$ ならば，Good Machine と Bad Machine とを識別しうる長さ2の入力集合 D^2AB を求めるステップに移る．説明を容易にするため，考えている回路は2個の擬似出力を有しているとし回路出力を出力1，擬似出力を出力2，出力3とする．第1ステップで求めた On アレーおよび Off アレーを次のように表現しておく．

A_{11}=Good Machine
　の出力1に関する On アレー
A_{01}=Good Machine
　の出力1に関する Off アレー
B_{11}=Bad Machine
　の出力1に関する On アレー
$B_{01}, A_{12}, A_{02}, B_{12}, B_{02}, A_{13}, A_{03}, B_{13}, B_{03}$

も同様である．

第6ステップ：A_{11} よりその成分 a_{i11} をとり，擬似入力に対応する座標はマスクして“X”を入れる．$B_{01}a_{i_{11}}=B_{01}\cap a_{i_{11}}$ を求める．もし，$B_{01}a_{i_{11}}\neq\phi$ であれば第7ステップを実行する．

第7ステップ：a_{i11} の擬似入力座標によって指定される2つのアレーの入力集合積をとる．例えば，第4番目の座標が出力2に対応し，第5番目の座標は出力3に対応している場合，$a_{i_{11}}=(10\times10)$ に対しては $A_{12}\cap A_{03}$ となる．この集合積と C_σ との入力集合積をとり S_A ($S_A=A_{12}\cap A_{03}\cap C_\sigma$)とする．同様に $B_{01}a_{i_{11}}$ に含まれる B_{01} の要素に対して S_B を求める．$b_j=(10\times01)$ であれば $S_B=B_{02}\cap B_{13}\cap C_\tau$ となる．

第8ステップ：S_A より $\widehat{S}_A$，S_B より $\widehat{S}_B$ を求め，$D^2a_ib_j=\widehat{S}_A\cap\widehat{S}_B$ を作る．$D^2a_ib_j\neq\phi$ ならば，これは Good Machine と Bad Machine とを識別しうる長さ2の入力集合の一部である．

第9ステップ：$D^2a_ib_j$ を保存しておき第7ステップ以降をすべての b_j について行なう．次に A_{11} がつきるまで第6ステップ以降を繰り返し実行する．

第10ステップ：$D^2_{AB}=D^2_{a_1b_1}\cup D^2_{a_1b_2}\cdots\cdots$ を求める．$D^2_{AB}\neq\phi$ であれば Good Machine と Bad Machine とを識別しうる長さ2の入力集合のうち，第2番目に加えるべき入力集合となる．第1番目に加えるべき入力は，$D^2a_ib_j$ を規定する $a_{i_{11}}$ において擬似入力座標を削除したものである．$D^2_{AB}=\phi$ であれば，第6ステップ以降と同様な方法を用いて D^n_{AB} ($n\geq3$) を求めれば良い．

5. 故障診断用のテスト・パターン抽出[10)13)]

On, Off アレー法またはトレーシング法を使用することにより図2に示すような診断D行列を作成することができる．この行列の列の欄にはテスト・パターンの候補である入力パターンの番号，行の欄には故障番号がそれぞれ対応している．この行列の要素 d_{ij} は故障 i がテスト・パターンの候補 j によって検出しうるとき“1”，その他のとき“0”（表にはブランクで示されている．）を有する．この表を使用して診断用のパターンを抽出することを考えよう．

a. 故障検出用のテスト・パターン抽出

i \ j	7	8	10	12	14	15	16	22	24	27	28	30	31	32
1	1	1	1	1	1	1	1							
2								1	1	1	1	1	1	1
3	1	1				1								
4			1	1	1									
5										1	1		1	
6								1	1			1		
7	1	1				1								
8	1	1				1								
9			1	1	1									
10			1	1	1									
12										1	1		1	
14								1	1			1		
15	1	1				1		1	1			1		
18			1	1	1					1	1		1	
22	1	1				1				1	1		1	
23								1		1				
24			1	1	1			1	1			1		

図2　D行列の例

故障検出用のパターンを求めるには，まずD行列の列（t-パターンと呼ぶ）t_i および t_j において $t_i\geq t_j$ ならば t_j を行列より消去する．$t_i\geq t_j$ とは $t_{ik}\geq t_{jk}(1\leq k\leq n)$ が成立することである．このような消去操作を行なった後の行列から，次に適当なテスト・パターン $(t_{j1}, t_{j2}, \cdots\cdots, t_{jk})$ を選択し任意の故障につき $Vj\in\{j_1\sim j_k\}d_{ij}=1$，しかも k を最小とする．

この選択の問題は組合せ回路の最簡設計の問題と同様な手法によって解くことができる．

b. 故障点指摘用のテスト・パターン抽出について

D行列より故障点指摘用のテスト・パターンを抽出するには多くの方法があるが，ここでは良好なテスト・パターンを求め得る「重み計算法」について考える．

重み計算法による診断テスト・パターン抽出法

m 個のテスト集合 $t_1, t_2, \cdots\cdots, t_m$ に対するD行列があるとする．2つの故障パターン $f_i=\langle d_{i_1}, d_{i_2}, \cdots\cdots, d_{im}\rangle$ と $f_j=\langle d_{j_1}, d_{j_2}, \cdots\cdots, d_{jm}\rangle$ とにおいて，ある k について $d_{ik}\neq d_{jk}$ であれば故障 f_i と f_j とは識別可能である．そこで，あるテストの相対的な重要さを比較する目的で重みを定義する．

テストは，それが最大の“故障対”数を識別しうるとき優先して採用すべきである．そこで t-パターンにおける“0”の数と“1”の数の積を重みとする．以上の考え方に，module という考え方を加えてみる．同一の module に含まれる任意の故障 f_i, f_j に関しては，f_i と f_j とを識別する必要はないので重みの計算法に修正を加えなくてはならない．

抽出の手順を示す前に使用する記号について説明する．今，S 個のテストがすでに抽出されているとする．この前に述べたD行列より分枝行列 $D\{t_{i_1}{}^{e_1}, t_{i_2}{}^{e_2}, \cdots\cdots, t_{i_s}{}^{e_s}\}$ を作成する．($e_i=0,1$) 分枝行列の列は $t_{i_1}, t_{i_2}, \cdots\cdots, t_{i_s}$ なる t-パターンを取り除いた残りの列であり，行はテスト $t_{i_1}{}^{e_1}, t_{i_2}{}^{e_2}, \cdots\cdots, t_{i_s}{}^{e_s}$ に関係した行である．また $D\{t_{i_1}{}^{e_1}, t_{i_1}{}^{e_2}, \cdots\cdots, t_{i_s}{}^{e_s}\}$ を $De\{t_{i_1}, t_{i_2}, \cdots\cdots, t_{i_s}\}$ と書く，ここに e は $(e_1, e_2, e_3, \cdots\cdots, e_s)$ を2進—10進変換した数である．De に関して次の記法を定義する．

n_0　t-パターン中の“0”の数

n_1　t-パターン中の“1”の数

$(n_0, n_1)t_j$　$De[t_{i_1}, t_{i_2}, \cdots\cdots, t_{i_s}]$ の t_j なる t-パターンにおける $n_0 \times n_1$ の値 ($t_j \neq t_{ix}$, $x=1, 2, \cdots\cdots, s$)

$t_j \cap P_i$　t-パターンと module パターン P_i との積 (module パターンは全体で k 個ある.)

$(n_0, n_1)t_j \cap P_i$　$t_j \cap P_i$ に於ける $n_0 \times n_1$ の値.

この定義のもとで，$De[t_{i_1}, t_{i_2}, \cdots\cdots, t_{i_s}]$ の分枝行列の重みは

$$W_j(e)=(n_0, n_1)t_j-\sum_{i=1}^{k}(n_0, n_1)t_j \cap P_i \qquad (6)$$

テスト t_j の重みは

$$W_j=\sum_{e=0}^{2^s-1} W_j(e) \qquad (7)$$

となる．次にテスト・パターンの抽出の手順を示す.

(1)　D行列を作成する．Dが $m \times n$ の行列である場合に，$j=1, 2, \cdots\cdots, m$ について W_j を計算する.

(2)　W_j のうちで最大なものを選択する．選択されたテストを t_{i_1} とする．ここで分枝行列 $D_0[t_{i_1}]$, $D_1[t_{i_1}]$ を作成する．この W_j のうちで最大のものに注目して，新らたなテスト t_{i_2} を抽出する.

(3)　$t_{i_1}, t_{i_2}, \cdots\cdots, t_{i_s}$ が選択された場合には，分枝行列 $De[t_{i_1}, t_{i_2}, \cdots\cdots, t_{i_s}]$ をつくり，$j=1, 2, \cdots\cdots, m$, $j \neq i_x (x=1, 2, \cdots\cdots, s)$ に対して W_j を計算し $t_{i_{s+1}}$ を抽出する.

(4)　(3)の操作を実行する際に，すべての j に対して $W_j=0$ であれば，それ以前に抽出された t-パターンが故障点指摘用のテスト・パターンである.

6. 故障診断用のテスト・パターン作成の実例[20)~25)]

トレーシング法によって故障点指摘用のテスト・パターンを作成した例を TPGSS (Test Pattern Generation System by Simulation) と呼ばれるプログラムにそくして述べる.

TPGSS プログラムでは，対象となる論理回路に任意の数の s—t—1 または s—t—0 の故障が生じた状態の回路出力を論理回路のシミュレータで求め，故障点指摘用のテスト・パターンを求めることができる．この方法を説明するフロー・チャートを図3に示す．この図において主要な部分は‘12’の判定箱の「現在の入力がテスト・パターンとして採用しうるか？」の部分である.

判定規準としては 5. で述べた“重み”を使用することができる．この場合は重み計算法によって作成された n 個のテストで故障回路の集合が 2^n 個の部分集合に分割されるので 2^n 分割法と呼ぶ.

2^n 分割法は故障回路の出力と正常な回路の出力との比較結果のみを利用し，出力パターンそのものを利用し

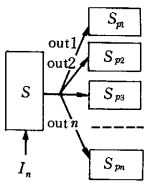

図4　故障回路集合の出力による分割

図3　TPGSS プログラムのフロー・チャート

ていないのでテスト・パターン作成の効率は必ずしも良くない.

そこで判断規準として“重み”の他に，次に述べる評価関数を採用する.

まず故障回路の集合 S に対して，各々の入力端子に同一の入力を加えてその検出を行ない．出力による回路の分割を行なう（図4参照）．ここで $H=\sum_{i=1}^{n}|S_{pi}|$ を計算する．$|S_{pi}|$ は分割によってできた部分集合 S_{pi} に含まれる故障回路の数である．この関数は $|S_{p1}|=|S_{p2}|=\cdots\cdots=|S_{pn}|$ となるとき最小となるので，Hを最小とする入力・パターンをテスト・パターンとして採用すれば良い．次に故障回路要素がどの module (IC, またはパッケージ)に含まれているかが判明している場合は module の段階まで故障点指摘を行なえば良い．そこで S_{pi} なる部分集合を module パターンによってさらに分割する．すなわち $S_{pi} \rightarrow S_{pi1}$ (module 番号1に含まれる故障)，$S_{pi2}, \cdots\cdots, S_{pik}$ なる分割を行なう．この分割において $\{S_{pij}\}$ の次元が1のときこの集合を終端集合と呼び，この集合に対しては入力パターンによるこれ以上の分割を行なう必要はない．そこで，評価関数 H_p は $H_p=\sum_{i=1}^{n}(\dim\{S_{pij}\})^2$ とし，この H_p が最小なる入力をテスト・パターンとすれば良い.

図5の回路に対して以上の方法を採用してテスト・パターンを作成した結果について説明しよう．この回路は

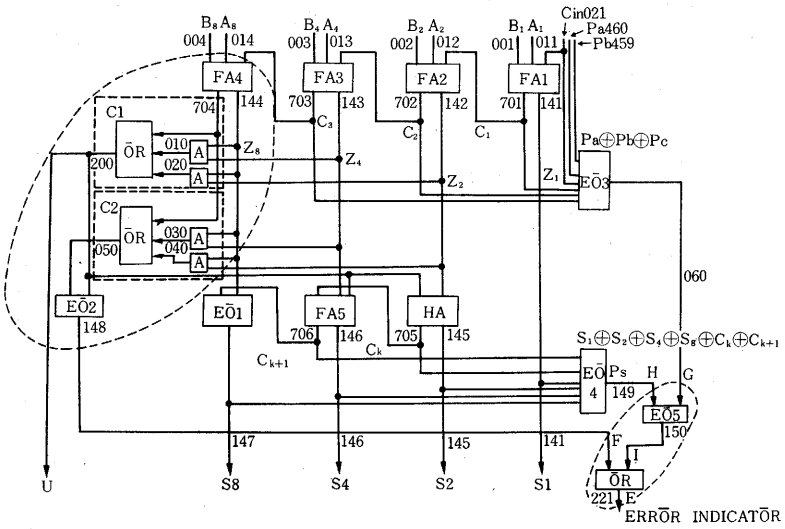

図 5　Carry-Dependent Sum Adder 回路図

最後に Carry-Dependent Sum Decimal Adder に関して故障検出用のテスト・パターンを作成した結果について報告する．作成法は一定数の乱数入力のもとで各故障論理回路の出力を求め，正常な回路の出力と故障回路の出力とを比較し，D行列を作成し，このD行列より検出用のテスト・パターンを抽出する方法を採用した．最終的に得られたテスト・パターンは図8に示す10個である．この図には各々のテスト・パターンで対象としている故障回路の集合の何割が検出しうるかを示す量である ρ の値もつけ加えてある．この Carry-Dependent Sum Decimal Adder において，検出しえない故障が12個存在した．その一つは図5の回路における EŌ2 の部分の故障である．C1 と C2 の部分が2重系になっていることから EŌ2 の入力に (1,0) または (0,1) なる入力を加えることができないことによるものである．他の例は回路設計の段階で冗長な論理を採用したことによるものであった．

Carry-Dependent Sum Decimal Adder[16]と呼ばれる10進加算器であり，回路構成に特別な注意をはらい，回路中の任意の単一故障による演算誤りのチェックが可能である．この回路に生じうる s—t—1 および s—t—0 の単一故障の数は270個であり，故障点を単一 module までつきとめることを目標にする．ここで，図5の回路は次の10個の module より構成されていると仮定する．
〔FA1〕，〔FA2〕，〔FA3〕，〔FA4, C1, C2, EŌ2〕，〔FA5〕，〔HA〕，〔EŌ1〕，〔EŌ3〕，〔EŌ4〕，〔EŌ5, ŌR〕．

TPGSS ではまず 2^n 分割法によって故障回路集合の分割を行ない3個のテスト・パターンで $\max(D^iS_j) \leq 90$ になる分割を得た．(図6参照)

次に，評価関数として H_p を使用してテスト・パターンを選択する．図7には 2^n 分割法で求まった D^3S_2 に関して求められた診断表を示す．D^3S_1 ……D^3S_8 の部分集合全体では約60個のテスト・パターンが求まる．

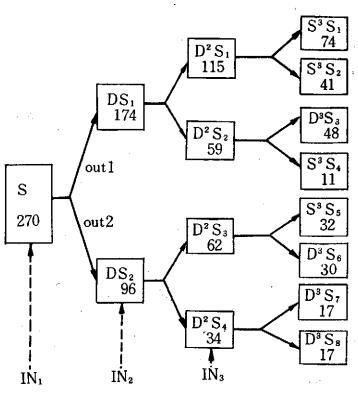

図 6　故障診断表 I (箱の中の数はその部分集合中の故障回路の数)

K	INPUT	Hp
1	(1387)	56
2	(83)	67
3	00100101100 (300)	43
4	(1051)	54
5	(1557)	46
6	(1930)	43

K	INPUT	Hp
1	(1701)	37
2	(853)	38
3	(760)	38
	(90)	
11	11101101110 (1902)	20
12	(41)	36

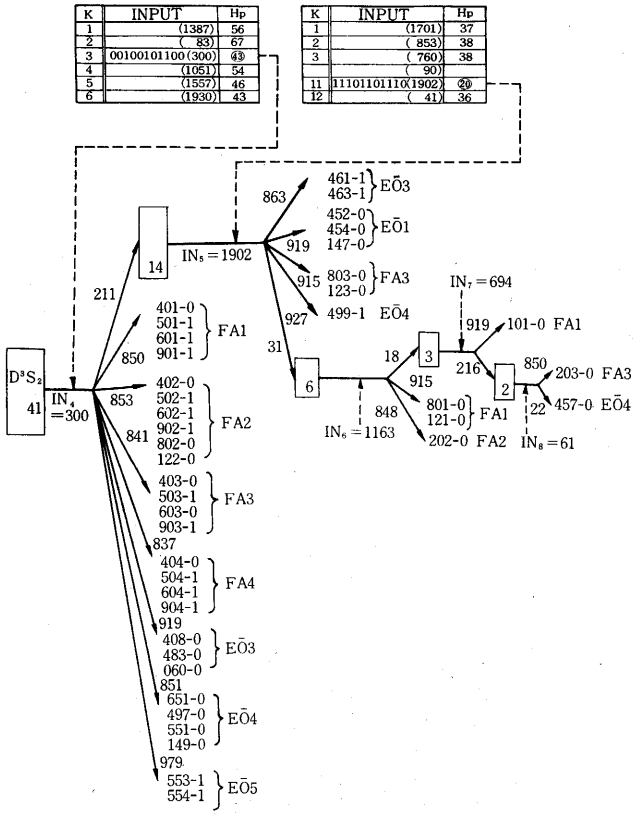

図 7　故障診断表 II

n	テスト・パターン(10進法)	テスト・パターン											出力パターン						ρ
		B8	B4	B2	B1	Pb	A8	A4	A2	A1	Pa	Cin	E	U	S8	S4	S2	S1	
1	787	0	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	1	105/258=0.41
2	1949	1	1	1	1	0	0	1	1	1	0	1	1	1	1	1	0	1	46/153=0.30
3	1521	1	0	1	1	1	1	1	0	0	0	1	0	1	1	1	1	0	37/107=0.35
4	204	0	0	0	0	1	1	0	0	1	1	0	0	0	0	1	0	0	30/70=0.43
5	318	0	0	1	0	0	1	1	1	1	1	0	0	1	0	1	1	1	5/40=0.13
6	1482	1	0	1	1	1	0	0	1	0	1	0	0	1	0	0	1	1	6/35=0.17
7	587	0	1	0	0	1	0	0	0	1	0	1	1	0	0	1	1	0	9/29=0.31
8	1122	1	0	0	0	1	1	0	0	0	1	0	0	1	0	1	1	0	13/20=0.65
9	651	0	1	0	1	0	0	0	1	0	1	1	0	0	1	0	0	0	6/7=0.86
10	711	0	1	0	1	1	0	0	0	1	1	1	1	0	0	1	1	1	1/1=1.00

図 8　Carry-Dependent Sum Adder 故障検出用テスト・パターン

なお TPGSS システムは本研究所の OKITAC-5090 C を使用して開発を行なった.

あ と が き

ディジタル・システムの自動故障診断のうち，特にテスト・パターンの作成法に重点をおいた．テスト・パターンの作成法には，ここで述べた他にも種々提案されているが，電子計算機を使用して作成するには，On, Off アレー法，トレーシング法またはこれらを修正したものが適当と考えられる．特に，1000個以上の論理回路要素を有する回路網のテスト・パターンを作成するときには計算時間の面でトレーシング法が有利である.

実際にディジタル・システムの自動診断を行なうには，テスト・パターンの作成および 2. で述べた診断サブ・システムをハード・ウェア化する必要があるが，これらの作業を現在研究が進められている DA (Design Automation) の中に包含し信頼度，保守度の高いディジタル・システムを自動設計することが今後の研究開発の方向であると思われる.

終りに，この研究を進めるに当って，協力していただいた電子計算機室の藤田講師，渡辺研究室，電子計算機室の諸氏に謝意を表します．(1968 年 10 月 24 日受理)

参 考 文 献

1) J. M. Galey, R. E. Norby, and J. P. Roth "Techniques for the Diagnosis of Switching Circuit Failures" IEEE Trans. on CE, No. 74, Sept. 1964.
2) J. M. Galey "Diagnosis of Failures in Sequential Circuits" Research Memorandum SR-157, IBM Corporation, Apr. 1961.
3) J. P. Roth and E. G. Wagner "Algebraic Topological Methods for the Synthesis of Switching Systems. Part III, Minimization of Nonsingular Boolean Trees" IBM J. Res. and Dev., vol. 3, Oct. 1959.
4) K. Maling and E. L. Allen "A Computer Organization and Programming System for Automated Maintenance" IEEE Trans. on EC, Dec. 1963.
5) S. Seshu and D. N. Freeman "The Diagnosis of Asynchronous Sequential Switching Systems" IRE Trans. on EC, Aug. 1962.
6) S. Seshu "The Logic Organizer and Diagnosis Programs" Report R-226, July 1964, Coordinated Science Laboratory, Univ. of Illinois.
7) W. C. Carter, H. C. Montgomery, R. J. Preiss, and H. J. Reinheimer "Design of Serviceability Features for the IBM System/360" IBM J. Res. and Dev., vol. 8, No. 2, Apr. 1964.
8) S. Seshu "On an Improved Diagnosis Program" IEEE Trans. on EC, Feb. 1965.
9) J. R. Bashkow "A Programming System for Detecting and Diagnosis of Machine Malfunctions" IEEE Trans. on EC, Feb. 1963.
10) D. B. Armstrong "On Finding a Nearly Minimal Set of Fault Detection Tests for Combinational Logic Nets" IEEE Trans. on EC, Feb. 1966.
11) J. Partridge, L. D. Hanley, and E. C. Hall "Progress Report on Attainable Reliability of Integrated Circuits Systems Application" Micro-Electronics and Large System, Spartan Books, 1965.
12) R. E. Forbes, D. H. Rutherford, and C. B. Stieglitz "A Self-Diagnosable Computer" Proceedings-Fall Joint Computer Conference, 1965.
13) H. Y. Chang "An Algorithm for Selecting on Optimum Set of Diagnostic Tests" IEEE Trans. on EC, Oct. 1965.
14) J. P. Roth "Diagnosis of Automata Failures: A Calculus and a Method" IBM J. Res. and Dev., July 1966.
15) J. P. Roth, et al "Programmed Algoithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits" IEEE Trans. on EC, No. 5, Oct. 1967.
16) M. Y. Hsiao "The Carry-Dependent Sum Adder" IEEE Trans. on EC, June 1963.
17) N. Deo "The Self-Diagnosability of a Computer" IEEE Trans. on EC, Oct. 1966.
18) 橋本昭洋，嵩 忠雄，尾崎 弘 "組合せ論理回路の検査に関する一考察" 昭和 39 年 4 月，電気通信学会雑誌 1964.
19) 渡辺 勝，杉本正勝 "論理回路の故障診断用テスト入力パターン作製プログラム・システム" 昭和 41 年，電気四学会連合大会—(1939)，1965. 4.
20) 杉本正勝 "ディジタル・システムの自動故障診断" 東京大学生産技術研究所，電気談話会報告 Vol. 16, No. 21, Nov. 24, 1966.
21) 渡辺 勝，杉本正勝 "論理回路の故障点指摘用テスト・パターン作成プログラム・システム" 昭和 42 年，電気四学会連合大会 (2890).
22) 渡辺 勝，杉本正勝，野村邦彦 "論理回路の故障検出用最適テスト入力パターン抽出プログラム" 昭和 42 年電気四学会連合大会 (2748).
23) 渡辺 勝，杉本正勝，小野寺隆 "ディジタル・システムの自動故障診断用のハード・ウェア" 昭和 42 年，電気四学会連合大会 (2745).
24) 杉本正勝 (修士論文) "ディジタル電子計算機システムを対象とした論理回路の故障診断に関する研究" 昭和 42年 2 月.
25) 渡辺 勝，杉本正勝，小野寺隆 "モデュール化された論理回路の故障診断用テスト・パターン作成法—MTR 法" 昭和 42 年，電子通信学会創立 50 周年記念全国大会(1005).