UDC 681.325.6.001.4

構造記述関数を用いた組合せ回路の故障検査について

正員 樹下 行三† 正員 高松 雄三††
正員 柴田 正治††† 正員 松藤 信哉††

A Method for Generating Tests in Combinational Circuits by Structure Description Functions

Kozo KINOSHITA†, Yuzo TAKAMATSU††, Masaharu SHIBATA††† *and* Shinya MATSUFUJI††, *Regular Members*

あらまし　本論文は，論理回路のゲートの接続関係および出力を記述する一つの論理式（構造記述関数と呼ぶ）を用いた故障検査について考察したものである．これまで提案されている構造記述関数は，回路が大きくなると式が急激に大きくなり実用化が困難である．そこで本論文では，まず，計算機処理に適した構造記述関数の表現式およびその算出アルゴリズムを与えている．この表現式は経路の数の大きさで表されている．次に，その構造記述関数に"経路に対するブール微分"という概念を導入し，経路が活性化できるための必要十分条件を与えている．又，この条件を基にして経路微分による検査系列生成法を述べている．次に，構造記述関数に故障微分を導入した検査系列生成について述べ，経路微分と故障微分のそれぞれの特徴を有効に併用した検査系列生成法を提案している．最後に，この方法のプログラムにより適用した結果を示している．ここで述べたプログラムを用いると検出率100%の検査系列が得られるので，回路の冗長性などの検証に有用であると思われる．

1. まえがき

論理回路の故障診断の問題はかなり古くから取り扱われており，これまで数多くの論理回路の検査系列を生成する手法が提案されている[1]~[4]．

本論文は，論理回路をその構造および出力を表す一つの論理式（以後，構造記述関数と呼ぶ）を用いて表現し，それを基にした検査系列生成法について考察したものである[5]．このような構造記述関数としては，文字命題記述[6]（literal proposition），enf[7]（equivalent normal form），SPOOF[8]（Structure-and Parity-Observing Output Function）などが知られており，更に，これらの関係および構造記述関数を用いた検査系列生成について考察がなされている[9]．又，SPOOF表現を計算機処理で求める方法も提案されているが[10]，実用化までは至っていないようである．

本論文では，まず計算機処理に適した構造記述関数の一表現法およびその算出アルゴリズムを示す．ここで提案する表現式は回路の経路の数の大きさで表されており，SPOOF表現を圧縮したものである．次に，"経路に対するブール微分"という概念を導入し，回路の経路が活性化できるための必要十分条件を与える．又，すべての経路に対するブール微分を行うことにより得られる検査系列と，その回路が実現している関数のブール微分[11]との関係を示す．更に，構造記述関数と故障との関係を述べ，検査系列生成のために故障微分を導入する．故障微分による検査系列生成は理論的には簡潔であるが，それをすべての故障に対して行うことは計算量の点から容易ではない[12]．そこで本論文では，経路微分と故障微分のそれぞれの特徴を有効に併用した検査系列生成法を提案する．この方法を用い

†広島大学総合科学部総合科学科，広島市
Faculty of Integrated Arts and Sciences, Hiroshima University, Hiroshima-shi, 730 Japan
††佐賀大学理工学部電子工学科，佐賀市
Faculty of Science and Engineering, Saga University, Saga-shi, 840 Japan
†††佐賀県立鳥栖工業高校，鳥栖市
Tosu Technical High School, Tosu-shi, 841 Japan
論文番号：昭 56-443[D-103]

ると検出率100％の検査系列が生成でき，検出可能でないすべての故障を知ることができる．このような検出可能でない故障を知ることは一般に容易ではない．最後に，この方法のプログラムにより適用された結果について述べる．この方法によると検出率100％の検査系列が得られるので，回路の冗長性などの検証に有用であると思われる．

2. 構造記述関数

まず，本論文で基本となる構造記述関数(Structure Description Function, 以下, SDFと略す)を定義し，その算出アルゴリズムを示す．又，以下の議論で必要となる諸定義について述べる．

ここで対象とするゲートは，AND, OR, NOT, NAND及びNORの5種とする．

次の条件を満足するような元の回路と等価な樹枝状回路で実現される関数を定義する．以下，この論理式を等価樹枝状回路表現(Equivalent Fanout-free Form, EFFと略記する)と呼ぶ．

(1) 入力端子から出力端子へ至るすべての経路を入力変数が同一であっても別個の入力変数として扱う．

(2) 入力に否定を許すAND及びORゲートのみからなる樹枝状回路である．

このようなEFFは次に示すアルゴリズムEFFを用いて得られる．以下，回路の信号線には，ラベルA, B, …が付されているものとし，ラベルは正または負の符号をもつものとする．又，i個の$L(i)$, $L(i-1)$, …, $L(1)$の系列を$\tilde{L}(i)$で表す．

〔アルゴリズムEFF〕

1° $i\leftarrow 1$, $j\leftarrow 1$.

2° 出力端子に接続されている信号線(Aとしよう)を取り出し，$L(i)\leftarrow A$とする．

3° 以下の手順を行え．

3.1° $i\leftarrow i+1$とし，ラベル$L(i-1)$から入力端子へ向い，ラベル$L(i-1)$がゲートGの出力線であればGの入力線のうちマークされていない入力線(Bとしよう)を取り出しBをマークする．3.3°へ．そうでなければ，ラベル$L(i-1)$が接続されている入力側の信号線(Cとする)を取り出し3.2°へ行く．

3.2° $L(i)\leftarrow C^{*}$, Cが入力端子に接続されていれば3.5°へ．そうでなければ，3.1°へ行く．

3.3° $L(i)\leftarrow B^{+}$(GがNOR, NAND, NOTのとき), $L(i)\leftarrow B^{*}$(GがOR, ANDのとき)とする．ここで，$B^{+}(B^{*})$はラベル$L(i-1)$と異符号(同符号)であることを表す．

3.4° Bが入力端子に接続されていれば3.5°へ．そうでなければ，3.1°へ行く．

3.5° 経路p_jとして$\tilde{L}(i)$を登録する．4°へ．

4° 以下の手順を行え．

4.1° ラベル$L(i)$が出力端子に接続されていれば，5°へ行く．ラベル$L(i)$がゲートGの入力線であれば4.3°へ．そうでなければ，4.2°へ行く．

4.2° $i\leftarrow i-1$として4.1°へ．

4.3° Gにマークされていない入力線(Dとしよう)が存在すれば，4.4°へ．そうでなければ，Gの入力線すべてのマークを消し，$i\leftarrow i-1$として，4.1°へ行く．

4.4° $L(i)\leftarrow D^{*}$(又は，D^{+})として，p_jとp_{j+1}の関係$R(j)$を次のように決定する．

$$R(j)\leftarrow G_N\cdot G_P\cdot \mathrm{sgn}(L(i-1))$$

ここで，G_Nは$\tilde{L}(i)$上のゲートの数であり，又，

$$G_P=\begin{cases}1 & (G\text{がOR又はNANDのとき})\\ -1 & (G\text{がAND又はNORのとき})\end{cases}$$

$$\mathrm{sgn}(L(i-1))=\begin{cases}1 & (\text{ラベル}L(i-1)\text{が正})\\ -1 & (\text{ラベル}L(i-1)\text{が負})\end{cases}$$

4.5° $j\leftarrow j+1$として，3°へ行く．

5° $R(j)$の絶対値の大きい順に，正のときはp_jとp_{j+1}との演算がOR，負のときはANDとなるように括弧付の式を構成する．

(アルゴリズムEFF終り)

〔例1〕 アルゴリズムEFFを用いて，図1の回路のEFF(S_fと書く)を求める．

1° $i\leftarrow 1$, $j\leftarrow 1$. 2° 出力端子に接続されているラベルLを取り出す．$L(1)\leftarrow L$.

3° $i\leftarrow 2$, Jを取り出しマークする．Lを出力線とするゲートはORであるから，$L(2)\leftarrow J$, $i\leftarrow 3$. $E1$を取り出す．$E1$をマークする．Jを出力線とするゲートはANDであるから，$L(3)\leftarrow E1$, $i\leftarrow 4$, Eを取り出す．$L(4)\leftarrow E$. Eは入力端子に接続されているラベルである．経路p_1としてx_{4EE1JL}を登録する．

4° $i\leftarrow 3$, ラベル$L(3)$($E1$)はゲートの入力線であり，そのゲートの入力線にマークされていないラベル$F1$が存在する．$L(3)\leftarrow F1$, $R(1)\leftarrow 2\times(-1)\times 1$, $j\leftarrow 2$, として3°へ行く．

3° $i\leftarrow 4$, Fを取り出す．$L(4)\leftarrow F$, $i\leftarrow 5$, Cを取り出しマークする．Fを出力線とするゲートはORであるから，$L(5)\leftarrow C$. Cは入力端子に接続されているラベルである．p_2として$x_{3CFF1JL}$を登録する．

以下，同様にして，経路p_3, p_4, …, p_9として，そ

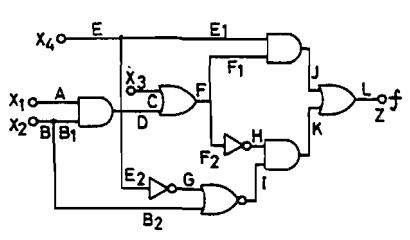

図1 論理回路の例†
Fig.1 - Logic circuit for example 1.

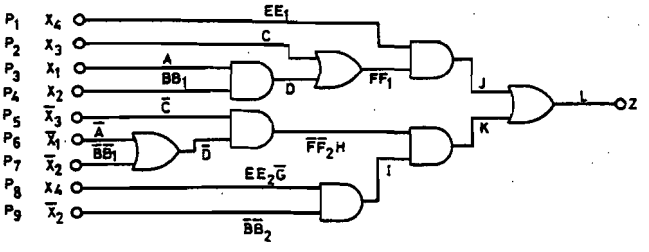

図3 等価樹枝状回路
Fig.3 - Equivalent fanout-free circuit for the circuit of Fig.1.

れぞれ，x_1ADFF_1JL，$x_2BB_1DFF_1JL$，$\bar{x}_3\bar{C}\bar{F}\bar{F}_2HKL$，$\bar{x}_1\bar{A}\bar{D}\bar{F}\bar{F}_2HKL$，$\bar{x}_2\bar{B}\bar{B}_1\bar{D}\bar{F}\bar{F}_2HKL$，$x_4EE_2\bar{G}IKL$，$\bar{x}_2\bar{B}\bar{B}_2IKL$（$\bar{}$記号は負のラベルを表す）が得られる．又，$R(j)$，$j=2,3,\cdots,9$は，3，−4，1，−4，5，−2，−3，0となる．

5° $R(j)$からS_fを構成すると図2のようにして，

$$S_f=p_1\cdot(p_2\vee p_3\cdot p_4)\vee p_5\cdot(p_6\vee p_7)\cdot p_8\cdot p_9$$

が得られる．

又，このとき得られる図1の等価樹枝状回路は図3で与えられる．

（例終）

上記のアルゴリズムEFFで得られるEFFを展開し，経路p_iの積和形で表した論理式を作るとSPOOF[8]となり，又，回路の信号線のラベルの代りにゲート名で経路p_iを表せばenf[7]となる．

SPOOF又はenfは積和形で表されているので，経路の個数の増加と共にその表現式は大きくなり実用的ではなくなる．しかしながら，本論文で導入したEFFは経路の数で表されるので，比較的大きな規模の回路のSDFとして用いることができると考えられる．

以下，本論文では，論理関数fを実現する回路のEFFをS_fで表す．又，S_fを展開して積和形で表した論理式をSp_fで表し，その積項をT_jで，T_jの部分積

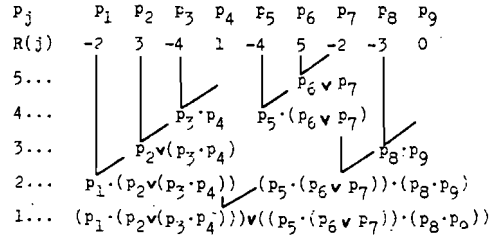

図2 S_fの構成
Fig.2 - Synthesis of S_f.

† この例は文献03 p.98の例である．

をt_jのように表すことにする．そうすると，Sp_fは，一般に次式で表される．

$$Sp_f=T_1\vee T_2\vee\cdots\vee T_k \qquad (1)$$

又，これらの式に現れる入力から出力へ至る経路をp_jで表す．一般に経路p_jは$p_j=x_i^*\alpha$で表されるから，x_i^*をp_jの入力リテラルと呼び，入力リテラルがx_i^*である経路p_jを$p_j<x_i^*>$と書く．但し，x_i^*はx_i又は$\bar{x}_i$であり，αは入力x_i^*から出力へ至る経路p_jを表すラベルの系列である．

又，信号線$A,B,\cdots$の信号値を$v_A, v_B,\cdots$のように表すことにする．

3. 経路微分と故障微分

3.1 経路活性化条件

ここでは，2.で述べたSDFに経路に対するブール微分という概念を導入し，経路活性化条件について考察する．

n個の入力変数$x_1, x_2,\cdots, x_n$を有する論理回路において，n個の入力変数x_iに定数$a_i\in\{0,1\}$又は，ドントケアdを適当に割り当てたn次元ベクトルを入力といいIで表す．以後，小文字のdはドントケアの意味で使用する．

〔定義1〕 与えられた回路に対して入力$I_1=(b_1,\cdots,b_{i-1},a_i,b_{i+1},\cdots,b_n)$を加える場合を考える．ここで，$b_j$は$a_j$又は$d$である．このとき，経路$p_j<x_i^*>$上の任意の信号線の信号値を1箇所変化させたとき，その信号線を入力とするゲートの出力値が変化すれば，経路$p_j<x_i^*>$はI_1でa_i-クリティカル経路であるという．

〔例2〕 図1の回路で$I_1=(1,0,0,1)$を加える．このとき，経路$p_7<\bar{x}_2>$について考えよう．$v_A=1$，$v_C=0$，$v_I=1$，$v_J=0$となる．従って，v_Kの値を変えるとKを入力とするゲートの出力値v_Lの値が変

る．以下，同様にして，定義1より$p_7<\bar{x}_2>$はI_1で0－クリティカル経路であることが分かる．同様の考察により，$p_5<\bar{x}_3>$及び$p_9<\bar{x}_2>$もI_1で0－クリティカル経路であることが分かる．（例終）

〔定義2〕 $I_1=(b_1,\cdots,b_{i-1},a_i,b_{i+1},\cdots,b_n)$とする．経路$p_j<x_i^*>$が$I_1$で$a_i$－クリティカル経路であり，且つ，$I_2=(c_1,\cdots,c_{i-1},\bar{a}_i,c_{i+1},\cdots,c_n)$で$\bar{a}_i$－クリティカル経路であるとき，$p_j<x_i^*>$は$I=(e_1,\cdots,e_{i-1},d,e_{i+1},\cdots,e_n)$で活性化経路であるという．但し，$c_j=b_j$又は$c_j=d$，$e_j=b_j\cap c_j$（$j\neq i$）であり，演算$\cap$は，$a\cap a=a$，$a\cap d=d\cap a=a$，$a\cap\bar{a}=\phi$（$\phi$は定義されないことを意味する）である．

〔例3〕 図1の回路に$I_1=(0,d,1,1)$を加え，経路$p_2<x_3>$を考えよう．$p_2<x_3>$は定義1からI_1で1－クリティカル経路である．又，$I_2=(0,1,0,1)$で0－クリティカル経路であることも分かる．従って，定義2より，$p_2<x_3>$は$I=(0,1,d,1)$で活性化経路である．（例終）

活性化経路においては，その入力Iにおいて，その定義から入力端子の変化は出力端子まで伝搬する．すなわち，活性化法で用いられた活性化経路[7]と同じになる．ところが，クリティカル経路においては，必ずしも入力端子の変化が出力端子まで伝搬するとは限らない．例えば，例2でみたように$p_7<\bar{x}_2>$は$I_1=(1,0,0,1)$で0－クリティカル経路であるが，$I_2=(1,1,0,1)$で1－クリティカル経路とならない．$v_f=0$となるからである．

しかし，次に定義する同時a_i－クリティカル経路という概念を導入することにより，従来の活性化法では得られなかった検査系列が求められる．

〔定義3〕 入力変数x_iから出力へ至る経路が2個以上存在するとして，それを$p_j<x_i^*>$，$p_k<x_i^*>$，$\cdots$，$p_l<x_i^*>$とする．$I_1=(b_1,\cdots,b_{i-1},a_i,b_{i+1},\cdots,b_n)$について$p_j$，$p_k$，$\cdots$，$p_l$がすべて$a_i$－クリティカル経路であるとき，経路$p_j<x_i^*>$，$p_k<x_i^*>$，$\cdots$，$p_l<x_i^*>$は$I_1$で同時$a_i$－クリティカル経路であるという．

〔例4〕 例2から，$p_7<\bar{x}_2>$，$p_9<\bar{x}_2>$は$I_1=(1,0,0,1)$で同時0－クリティカル経路である．（例終）

〔定義4〕 経路$p_j<x_i^*>$を$\bar{a}_i$－クリティカル経路とする入力Iが存在しないとき，$p_j<x_i^*>$は非クリティカル経路であるという．

例えば，図1の回路の$p_6<\bar{x}_1>$は非クリティカル経路である．

以下，回路のSDFが与えられたとき，その中で定義されている経路が活性化経路かクリティカル経路か，又は非クリティカル経路であるかを判定する方法を述べよう．

〔定義5〕 次式で定まるdS_f/dp_jをS_fの経路p_jに対する経路微分という．

$$\frac{dS_f}{dp_j}=S_f(p_j=1)\oplus S_f(p_j=0) \tag{2}$$

ここで，$S_f(p_j=a)$はS_fに現れるp_jの値をaにした式を表す．

〔定義6〕 dS_f/dp_jのすべての$p_l(l\neq j)$に対応する入力リテラルx_l^*を代入して得られる関数をp_jの経路活性化関数といい$fp_j(X)$で表す．又，$fp_j(X)=1$を満たす入力の集合をI_jで表す．

〔例5〕 図1の回路のS_fのp_1に対する経路微分dS_f/dp_1，及びp_1の経路活性化関数$fp_1(X)$を求める．

例1で求めたS_fから，

$$\begin{aligned}\frac{dS_f}{dp_1}&=(p_2\vee p_3p_4\vee p_5p_6p_8p_9\vee p_5p_7p_8p_9)\\&\quad\oplus(p_5p_6p_8p_9\vee p_5p_7p_8p_9)\\&=(p_2\vee p_3p_4)\cdot(\bar{p}_5\vee\bar{p}_6\vee\bar{p}_8\vee\bar{p}_9)\\&\quad\cdot(\bar{p}_5\vee\bar{p}_7\vee\bar{p}_8\vee\bar{p}_9)\end{aligned}$$

又，$fp_1(X)$はp_j，$j=2,3,\cdots,9$の入力リテラルが，それぞれ，x_3，x_1，x_2，$\bar{x}_3$，$\bar{x}_1$，$\bar{x}_2$，x_4，$\bar{x}_2$であるから，dS_f/dp_1に代入することにより，

$$\begin{aligned}fp_1(X)&=(x_3\vee x_1x_2)\cdot(x_3\vee x_1\vee\bar{x}_4\vee x_2)\\&\quad\cdot(x_3\vee x_2\vee\bar{x}_4\vee x_2)\\&=x_3\vee x_1x_2\end{aligned}$$

が得られる．従って，I_1は，

$$I_1=\{(d,d,1,d),(1,1,d,d)\}$$

である．（例終）

〔補題1〕 回路の経路$p_j<x_i^*>$がクリティカル経路であるための必要十分条件は，$I_j\neq\phi$（ϕは空集合を表す）であることである．（証明略[5]）

定義1，2及び定義4と補題1から次の定理が得られる．

〔定理1〕 経路$p_j<x_i^*>$が活性化経路であるための必要十分条件は$I\in I_j$なるIが存在することである．又，$p_j<x_i^*>$がクリティカル経路であるための必要十分条件は，$I_1\in I_j$なるI_1が存在することである．但し，I，I_1は，それぞれ，$I=(b_1,\cdots,b_{i-1},d,b_{i+1},\cdots,b_n)$，$I_1=(b_1,\cdots,b_{i-1},a_i,b_{i+1},\cdots,b_n)$，$b_j\in\{0,1,d\}$，$a_i\in\{0,1\}$である．

〔系1〕 $I_j=\phi$（ϕは空集合を表す）のとき，且つ，そのときに限り$p_j<x_i^*>$は非クリティカル経路である．

〔系2〕 回路の経路$p_j<x_i^*>$，$p_k<x_i^*>$，…，$p_l<x_i^*>$が同時a_i-クリティカル経路であるための必要十分条件は，$I_1\in I_j\cap I_k\cap\cdots\cap I_l$なる入力$I_1$が存在することである．ここで，演算$\cap$は，$^\forall I_j\in I_j$，$^\forall I_k\in I_k$について定義2の演算$\cap$を施すことである．

経路$p_j<x_i^*>$が活性化経路であることは，その経路の遅延テスト[10]（測定）が可能であることを示している．従って，回路の経路が活性化経路であるための必要十分条件を与えている定理1は故障検査系列生成に対してと同様有用であると思われる．

〔定義7〕 ある回路の入力変数x_iから出力へ至るすべての経路を$p_{j1}<x_i^*>$，$p_{j2}<x_i^*>$，…，$p_{jl}<x_i^*>$とする．このとき，$m(2\leqq m\leqq l)$個の経路$p_{j1}<x_i^*>$，…，$p_{jm}<x_i^*>$が$I_1=(b_1,\cdots,b_{i-1},a_i,b_{i+1},\cdots,b_n)$で同時$a_i$-クリティカル経路であり，且つ，(i)残りの$(l-m)$個の経路$p_{j_{m+1}}<x_i^*>$，…，$p_{jl}<x_i^*>$が$I_2=(b_1,\cdots,b_{i-1},\bar{a}_i,b_{i+1},\cdots,b_n)$で$\bar{a}_i$-クリティカル経路でない，及び(ii)$I_2$で$a_k$-クリティカル経路（$a_k=b_k$，$k\neq i$）である経路$p_q<x_k^*>$が存在しない，とき，経路$p_{j_1}<x_i^*>$，…，$p_{j_m}<x_i^*>$は$I=(b_1,\cdots,b_{i-1},d,b_{i+1},\cdots,b_n)$で同時活性化経路であるという．

〔定理2〕 ある回路の入力変数x_iから出力へ至る経路$p_{j_1}<x_i^*>$，…，$p_{j_m}<x_i^*>$が$I=(b_1,\cdots,b_{i-1},d,b_{i+1},\cdots,b_n)$で同時活性化経路であるとき，入力$I$を加える場合を考える．このとき，入力$x_i$の変化は経路$p_{j_1}<x_i^*>$，…，$p_{j_m}<x_i^*>$を経て出力端子に伝搬する．

（証明略[5]）

これまでの考察から検査入力を生成することは，経路微分を行い，その活性化関数を1とする入力集合を求めることに帰着される．経路微分を行うとき，次の補題2が有用である．

〔補題2〕 Sp_fの項T_jにおいて，経路p_jが含まれている項を$T_{j_1},\cdots,T_{j_m}$，そうでない項を$T_{i_1},\cdots,T_{i_l}$とすると，式(2)の経路微分は次式で与えられる．

$$\frac{dS_f}{dp_j}=(t_{j_1}\vee\cdots\vee t_{j_m})\cdot\bar{T}_{i_1}\cdots\bar{T}_{i_l}\qquad(3)$$

ここで，$T_{j_k}=p_j\cdot t_{j_k}$（$k=1,\cdots,m$）である．

定理1，2及びブール微分[11]の定義から次の定理3が成立する．

〔定理3〕 次のいずれかの条件を満足する$I\in I<x_i^*>$は関数fのx_iに関するブール微分df/dx_iを1とする入力である．又，逆も成り立つ．

(1) $p_j<x_i^*>$を活性化経路とする入力I

(2) $p_{j_1}<x_i^*>$，…，$p_{j_m}<x_i^*>$を同時活性化経路とする入力I

但し，$I<x_i^*>$は入力リテラルがx_i^*である経路$p_j<x_i^*>$のI_jの和集合を表す．

〔例6〕 図1の回路のすべてのI_jを求めると次のようになる．

$p_1<x_4>: I_1=\{(d,d,1,d),(1,1,d,d)\}$

$p_2<x_3>: I_2=\{(0,d,1,1),(0,1,d,1),(d,0,1,1)\}$

$p_3<x_1>: I_3=\{(d,1,0,1)\}$

$p_4<x_2>: I_4=\{(1,1,0,1)\}$

$p_5<\bar{x}_3>: I_5=\{(d,0,0,1)\}$

$p_6<\bar{x}_1>: I_6=\phi$

$p_7<\bar{x}_2>: I_7=\{(1,0,0,1)\}$

$p_8<x_4>: I_8=\{(d,0,0,d)\}$

$p_9<\bar{x}_2>: I_9=\{(0,d,0,1),(d,0,0,1)\}$

定理3から，この回路が実現する関数fのブール微分df/dx_iを求めると，次のようになる．

$df/dx_1=x_2\bar{x}_3x_4$　（I_3から）

$df/dx_2=\bar{x}_1\bar{x}_3x_4$　（I_9から）

$df/dx_3=\bar{x}_1x_2x_4$　（I_2から）

$df/dx_4=x_3\vee x_1x_2\vee\bar{x}_2\bar{x}_3$　（I_1及びI_8から）

（例終）

3.2 故障微分による検査系列の生成

ここでは，仮定した故障を検査する入力を，EFFを用いて求める方法として，S_fの故障微分について述べる．

以下，経路に含まれるラベルが否定形（2.では負のラベルとした）のとき否定ラベル，そうでないとき，肯定ラベルという．又，信号線Bの値が$a\in\{0,1\}$に縮退するときの故障を$F(B=a)$で表す．ラベルBを含む経路p_jを考えたとき，Bが否定（肯定）ラベルのときp_jの値を$\bar{a}(a)$とすることを，故障$F(B=a)$による経路p_jの値という．

〔定義8〕 S_fにおいて，$F(B=a)$による経路の値を代入したものを，信号線Bのa縮退故障関数といい，$S_f(B=a)$で表す．

〔定義9〕 S_fの故障$F(B=a)$による故障微分を次の式で定義する．

$$\frac{dS_f}{dF(B=a)}=S_f\oplus S_f(B=a)\qquad(4)$$

式(4)に現れるすべての経路に対応する入力リテラルx_i^*を代入して得られる関数を$f_{F(B=a)}(X)$で表すと，

$f_{F(B=\alpha)}(X)=1$ を満たす入力は，故障 $F(B=\alpha)$ を検査する入力となる．又，$f_{F(B=\alpha)}(X)=0$ であれば，故障 $F(B=\alpha)$ は検査できない（検出可能でない故障）ことを意味する．

〔例7〕 図1の回路の故障 $F(A=1)$ による故障微分を求めよう．ラベル A を含む経路は p_3，$\bar{A}$ を含む経路は p_6 であるから，$p_3=1$，$p_6=0$ を例1の S_f に代入して，

$$S_f(A=1)=p_1(p_2\vee p_4)\vee p_5p_7p_8p_9$$

従って，

$$\frac{dS_f}{dF(A=1)}=\{p_1(p_2\vee p_3\cdot p_4)\vee p_5(p_6\vee p_7)p_8p_9\}$$
$$\oplus\{p_1(p_2\vee p_4)\vee p_5p_7p_8p_9\}$$

次に，$p_1, p_2, \cdots, p_9$ に対応する入力リテラル $x_4, x_3, x_1, x_2, \bar{x}_3, \bar{x}_1, \bar{x}_2, x_4, \bar{x}_2$ を代入して整理すると，

$$f_{F(A=1)}(X)=\bar{x}_1x_2\bar{x}_3x_4$$

が得られる．従って，（0，1，0，1）は信号線 A の1縮退故障を検出する入力である．　　（例終）

ここで述べた故障微分は3.1で述べた経路微分と関係があり，経路微分は故障微分を1とする一つの条件になっている．すなわち，故障微分は経路微分を用いて表すことができる[5]．例えば，例7で求めた $dS_f/dF(A=1)$ は，

$$\frac{dS_f}{dF(A=1)}=\bar{p}_3\cdot\left(\frac{dS_f}{dp_3}\right)\vee p_6\cdot\left(\frac{dS_f}{dp_6}\right)_{p_3=1}$$

となる．

4. 検査系列生成法

これまで述べたように，回路の検査系列を生成するには，経路微分を行い $f_{p_j}(X)=1$ とする入力を求めるか，又は，故障微分を行い $f_{F(B=\alpha)}(X)=1$ とする入力を求めればよい．このとき，経路微分のみの方法では検出率† 100％の検査系列を生成することは一般には困難である．一方，故障微分のみの方法を用いて検査系列を生成することは回路が大きくなると生成時間が増加し[12]、実用的でない．ここでは，経路微分と故障微分のそれぞれの特徴を有効に併用した検査系列生成法を述べる．

4.1 プログラムの構成

図4に経路微分と故障微分を併用した検査系列生成プログラムの概要を示す．プログラムは，(Ⅰ)回路記述から回路テーブルの作成，(Ⅱ) EFFの算出，(Ⅲ)経路微分，(Ⅳ)故障微分，の四つの部分から構成されている．以下，簡単に説明する．

(Ⅰ)　ここでは，被検査回路の回路記述データを読み込み，二つの回路テーブル（内部表現）を作成する．

(Ⅱ)　回路テーブルから，2.で述べたアルゴリズムEFFを用いてEFFを算出する．

(Ⅲ)　S_f の経路 p_j に関する経路微分を行い，$f_{p_j}(X)=1$ を満たす入力を生成する．

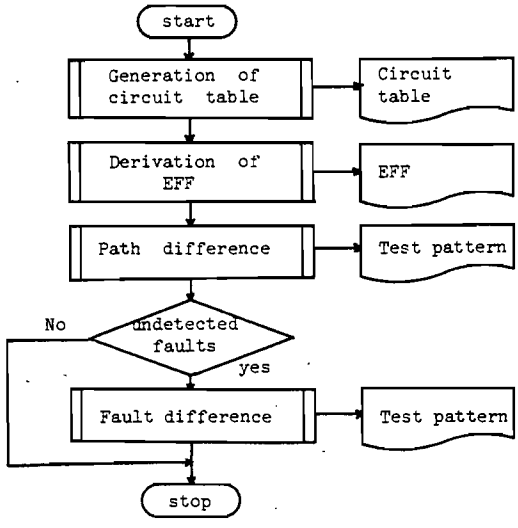

図4　全体の構成図
Fig.4 - General flow.

† 検出率$=\frac{\text{テストにより検出された故障数}}{\text{被検査回路の故障数}}\times 100$

表1　実行結果

回路番号	経路微分		故障微分			検出可能でない故障数	テストジェネレーション（秒）
	時間(秒)	検出率(%)	時間(秒)	検出率(%)	生成パターン数		
1	77.0	94.0	42.7	100	15	0	120
2	17.2	91.9	1.7	100	18	0	18.9
3	121	83.0	240	99.4	77	4	361
4	98.4	83.5	83.6	97.4	104	17	182

表2 被検査回路の構造

回路番号	ゲート数	信号線数	入力数	出力数
1	96	264	14	8
2	70	177	9	5
3	107	312	14	8
4	118	331	10	4

(Ⅳ) S_f の故障 $F(B=\alpha)$ による故障微分を行い，$f_{F(B=\alpha)}(X)=1$ を満たす入力を生成する．

4.2 実行結果と評価

作成したプログラムを100ゲート程度からなる回路に対して実行した結果の一部を表1に示す．プログラムはFORTRANで記述され，おおよそ2,000ステップである．使用計算機は九州大学大型計算機センターFACOM M-200である．

ここで述べたプログラムを用いると，検出率100％の検査系列が得られる．表1の中で回路番号3，4のそれが100％でないのは被検査回路に検出可能でない故障が存在するためである．このような検出可能でない故障をすべて知ることは困難であるため，これまで，あまり検出率が100％の実行結果は得られていないようである．しかしながら，EFFの故障微分という概念を用いると，検出可能である故障のすべての検査系列が得られると同時に検出可能でない故障も知ることができる．例えば，回路番号3及び4について，それぞれ，検出可能でない故障が4個および17個存在している．このような結果は，回路の冗長性などの検証に利用できると考えられ有用であろう．

5. むすび

ここでは，圧縮したSDF表現としてEFFを導入し，そのEFFを用いた検査系列生成法を考察した．

まず，EFFに経路微分という概念を定義し，経路の活性化条件を明らかにし，経路微分による検査系列生成法を示した．又，EFFに故障微分を導入し，仮定した故障の検査系列の生成について述べた．次に，EFFの経路微分と故障微分を併用した検査系列生成プログラムの実行結果について簡単に示した．このプログラムを用いると検出率100％の検査系列が得られるので，回路の検証などに有用であると思われる．

しかしながら，回路の経路の数が多くなると検査系列生成時間が大きくなると考えられるので，EFFの分割表現を導入することなどが考えられるが[15]，これらは今後の課題である．なお，本研究は一部文部省科学研究費（課題番号一般C455146）による．

文献

(1) Chang, H.Y., Manning, E.G. and Metze, G.: "Fault diagnosis of digital systems", John Wiley & Sons, Inc. (1970).

(2) Friedman, A.D. and Menon, P.R.: "Fault detection in digital circuits", Prentice-Hall (1971).

(3) Breuer, M.A. and Friedman, A.D.: "Diagnosis & reliable design of digital systems", Computer Science Press (1976).

(4) Roth, J.P.: "Diagnosis of automata failures: a calculus and a method", IBM Res. & Dev., 10, pp.278-291 (July 1966).

(5) 樹下，高松，柴田："構造記述関数を用いた組合せ回路の故障検査法"，信学技報，EC79-71 (1980-02).

(6) Poage, J.F.: "Derivation of optimum tests to detect faults in combinational circuits", Proc. Symp. Math. Theory of Automata, Polytechnic Institute of Brooklyn, pp.483-528 (1963).

(7) Armstrong, D.B.: "On finding a nearly minimal set of fault detection tests for combinational logic nets", IEEE Trans. Electron. Comput., EC-15, 1, pp.66-73 (Feb. 1966).

(8) Clegg, F.W.: "Use of SPOOF's in the analysis of faulty logic networks", IEEE Trans. Comput., C-22, 3, pp.229-234 (March 1973).

(9) Hayes, J.P.: "Test generation using equivalent normal forms", J. Des. Autom. & Fault-Tolerant Comput., 3, 3-4, pp.131-154 (1979).

(10) Si, S.-C. and Susskind, A.K.: "A method for obtaining SPOOF's", IEEE Trans. Comput., C-24, 5, pp.560-562 (May 1975).

(11) Sellers Jr., F.F., Hsiao, M.Y. and Bearnson, L.W.: "Analyzing errors with the Boolean difference", IEEE Trans. Comput., C-17, 7, pp.676-683 (July 1968).

(12) 樹下，高松，柴田，松藤："構造記述関数の故障微分による検査系列生成について"，情報処理学会第21回全国大会論文集，p.1097 (昭55).

(13) 安居院，内藤："論理回路の故障診断"，産報（昭51).

(14) Shedletsky, J.J.: "Delay testing LSI logic", Proc. 1978 Int. Symp. Fault-Tolerant Computing, pp.159-164 (June 1978).

(15) 石橋，松藤，高松，樹下："論理回路の構造記述関数の一分割法"，昭55九州連大，534.

（昭和55年10月15日受付，56年2月18日再受付）