順序回路故障検出系列発生の一手法†

村上道郎†† 小澤康明††

Abstract

The 'D-algorithm' by Roth is considered, now, that it is practically very useful method to detect the failures of a combinational circuit, and has been used by many systems.

By improving it, we made up a procedure of a fault detecting pattern generation to detect the failures of sequential circuit with flip-flops. This procedure is based on Kubo's philosophy wich is to transform a sequential circuit to a combinational circuit with delay elements. Adding this philosophy, we made up a algorithm to solve the logical equation of flip-flops, and introduced the idea of NODE and BRANCH to standarize the procedure of the D-algorithm.

In this paper, first the outline of the D-algorithm is presented, then, our procedure of fault detecting pattern generation for a sequential circuit is presented next.

1. まえがき

LSI，論理パッケージ等の論理機能試験の完全を期するために，その入力系列を電算機により発生させる自動故障検出系列発生方法 (Automatic fault detecting pattern generating method) の研究が盛んに行なわれ，数多くの論文が発表されてきた．

中でも J.P. ROTH (IBM) の D-アルゴリズム[1]は，実用的である等の利点から最も有効であると考えられ，これを利用した方法が一般に採用されてきた．当初，それは組合せ回路のみしか取扱えなかったが，久保（日電）[2]，ROTH[3]の研究により，非同期回路をも含めた順序回路への適用も可能になった．

すなわち，順序回路は **Fig. 1** (**a**) の一般型で表わされるが，これを **Fig. 1** (**b**) に示すように FP から GP へのフィード・バック・ループを切断し，GP と FP の縦続複写を行なえば，各クロック (時間遅れ) の場における論理値を各複写部分で表わすことが可能となり，フリップ・フロップ（以下，FF と略す.）を単なる1クロック分の遅れ素子に限定すれば，順序回路を組合せ回路のように見なすことができ，D-アルゴリズムの適用が可能となるのである．

しかし，これらの方法では R-S, J-K などの FF を取扱うには，ゲート単位に展開しなければならず，回路のライン数を増大させるなどの欠点があり実用的でなかった．

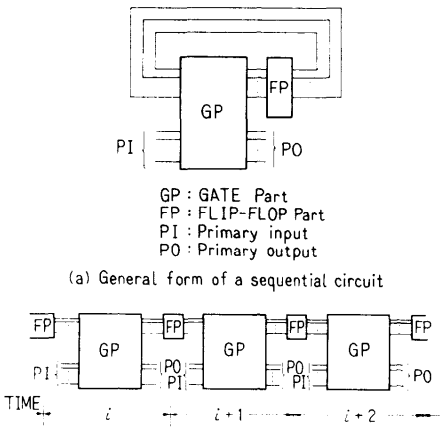

Fig. 1 Model of a Sequential circuit

そこで我々は，これらの研究を進め，(1) FF はゲートに展開せずに特性方程式で表わされたマクロ・ゲートと見なし，それを与えられた条件で解くことにより各種の操作を行なう．(2) ノード，ブランチの概念を導入して，D-アルゴリズムの FF の処理をも含めた操作を定義化し，アルゴリズム自体の統一を図る．

以上の2点を主改良点とした故障検出系列発生の手法を開発した．

† A Procedure of Fault Detecting Pattern Generation for a Sequential circuit, by Michio Murakami and Yasuaki Ozawa (Component Laboratory, Engineering Development Division, OKI Electric Industry Co., Ltd.)

†† 沖電気工業株式会社・開発本部・部品研究所

本論文はこの手法の紹介であり，初めにノード，ブランチの概念とその操作法について，次にプログラム，特にFFの特性方程式の解法について説明し，最後に結果を簡単に説明する．

2. ノード，ブランチ，優先関数を用いたD-アルゴリズムの拡張

D-アルゴリズムにおけるD操作†，C操作††にFFの処理機能を追加し，その操作の一般化を図るため，ノード，ブランチの概念を導入した．

ノードとは，故障検出系列を求める試行の過程中，D操作，C操作において2つ以上の選択の可能性のある段階，また，ブランチとは，そのうちの1つの試行と定義される．

優先関数とは，ノードにおいてブランチの選択順序を決め，それに従ってブランチを採用することによりD操作，C操作を有効に行ない，系列発生効率を高めるために採り入れられた関数である．

次のように定義される．

D操作（C操作）において，ノードを発生した段階で，1つのブランチを構成する複数個または1個のラインの各々の外部出力（入力）端子までの距離††† を計算し，それらの逆数の最大値をそのブランチの値とする関数である．

したがって，あるノードでブランチを選択する場合，各ブランチの優先関数値を計算し，その最大のものから採用していけば，系列長の最小のものを得ることができ，効率のよい系列の発生が可能となる．

また，操作の過程で論理矛盾が発生すれば，現在の（最後に定義した）ノードに戻り，別のブランチを優先関数値にしたがって選択するので，ROTHのD-アルゴリズムと同様に，最悪の場合にはブランチのすべての可能な組合せを尽すことができる．

2.1 ノード，優先関数の分類

ノードはそのタイプにより，D操作に関して3種類，C操作に関して2種類，計5種類に分類できる††††．

（1） D操作に関して

○タイプ 1

ある素子のD-intersection† を行なったとき，その素子の出力を入力として持つ素子 (Successor) を要素とするベクトル，活性ベクトル**A**を定義し，その活性ベクトルの次数（要素の数）が複数の場合，これは選択の可能性のある段階であり，タイプ1のノードを発生する．

優先関数は **Fig. 1**（b）上で活性ベクトル **A** に含まれている各ラインから外部出力までの距離で表わされ，近いもの程高い．

たとえば，**Fig. 3** のライン7に "D" を付加した場合，活性ベクトル **A** は (8, 9, 11, 12) となり優先関数により 11, 12, 8, 9 の順でブランチとして採用される．

○タイプ 2

故障情報 "D", "$\bar{D}$" を FF の出力 Q_{N+1} に出すための組合せが複数個存在する場合である．

優先関数値は，出力値 Q_N を決定する必要のないものが最も高い．Q_Nの値を必要とすることは，**Fig. 1**（b）において外部入力に最も遠いラインの値を決定する必要が生じることを意味するからである．

たとえば，**Fig. 2** の R-S 型 FF において，クリアー入力が "D"，その他"がすべて "X" の場合には **Table 1** に示すように3通りの組合せが存在し，タイプ2のノードを発生し，優先関数にしたがって，3番目の組合せが最初のブランチとなる．

Table 1 Combination of CL="D".

Q_N	S	R	ϕ	CL	Q_{N+1}
1	×	0	×	D	$\bar{D}$
1	×	×	0	D	$\bar{D}$
×	1	0	1	D	$\bar{D}$

×: don't care condition

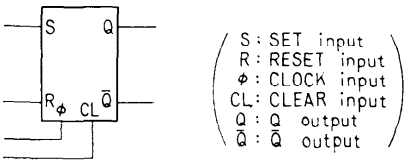

Fig. 2 Model of R-S type flip-flop.

○タイプ 3

FF の出力に "D", "$\bar{D}$" が出た時点で，その FF の Q, $\bar{Q}$ 出力両方が存在する場合には，Q, $\bar{Q}$ のどちらを伝搬させるかについて選択の可能性があるのでタイプ3のノードを発生する．実際の処理方

† D-operation: 故障情報 "D" を外部出力へ伝搬させるための必要最小限の論理値を決定する操作．

†† C-operation: D 操作で決定した各論理値を得る外部入力の論理値を決定する操作．文献 1) 参照．

††† 距離が1とは1つの素子からラインにより直接結ばれる次の素子までの距離のことをいい，この距離はFFによる時間遅れを考慮したFig. 1 (b) 上で測る．

†††† ROTH の D-アルゴリズムでは，D操作，C操作共に1種類，計2種類と考えられる（タイプ 1, 4）．

† 故障情報 "D", "$\bar{D}$" を入力として持つ1個のゲート，FF に対しその故障情報を出力に出す入力の論理値を求める操作．文献(1)参照

法，優先関数はタイプ1と同じである．

(2) C操作に関して

○タイプ4

ゲートのC操作を行なうとき，すなわち，与えられた出力値を得るための入力の論理値の組合せが複数個存在するときは，タイプ4のノードを発生する．

優先関数は **Fig. 1**(b) 上で値を決定するラインから外部入力までの距離で表わされ，近いもの程高い値を示す．

たとえば，AND ゲートの出力値 '0' を得るには，'X' を持つ入力のどれか1個に '0' を与えればよいのであり，組合せは 'X' を持つ入力の個数だけ存在する．

○タイプ5

FFのC操作を行なうときは，タイプ5のノードを発生する．

優先関数値はタイプ2と同様な意味から Q_{N-1} の値を決定する必要のないもの程高い．

たとえば，R-S 型 FF の1時点前の入出力値すべて 'X' のとき，出力値 '1' を得る組合せは，**Table 2** に示すように3通り存在し優先関数により3番目の組合せが始めに選択される．これは1時点前のセット入力とクロック入力を '1' に，リセット入力，クリアー入力を '0' にするものである．

Table 2 Combination of Q_N='1'

Q_{N-1}	S	R	CL	ϕ	Q_N
1	×	0	0	×	1
1	×	×	0	0	1
×	1	0	0	1	1

2.2 操作方法

以上に述べたノード，ブランチ法によるD操作，C操作により故障検出系列の発生を行なう操作方法を，**Fig. 3** の回路のライン7の縮退 '1' 故障の具体例によって説明する．なおFFに関する操作については**3.1**で詳述し，ここでは結果のみを示す．

また，問題を簡単にするためにに **Fig. 1**(b) おける時間 i を1とし，その時以降の全論理値を 'X' とする．**Fig. 4** にモデル化した **Fig. 3** の回路を示し，図中のライン番号の右下の添字で時間を表わすものとする．

まず，ライン 7_1 に 'D̄' を付加する．この時点での活性ベクトル **A** は (11_2, 12_2, 8_1, 9_1) だからタイプ1のノードを発生する（ノード番号1）．仮にライン 11_2 を選択すれば，このラインに 'D̄' を伝搬させる組合せは1個でライン 11_1, 8_1 を '0' に，18_1 を '1'，にすればよい．ここでさらにライン 14_1, 17_1 が '0'，ライン 2_1 が '1' に一意決定される．

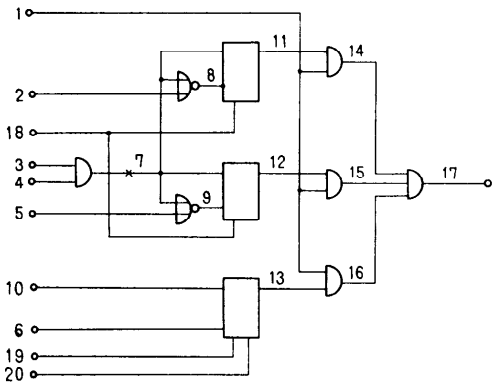

Fig. 3 Example circuit for manual procedure.

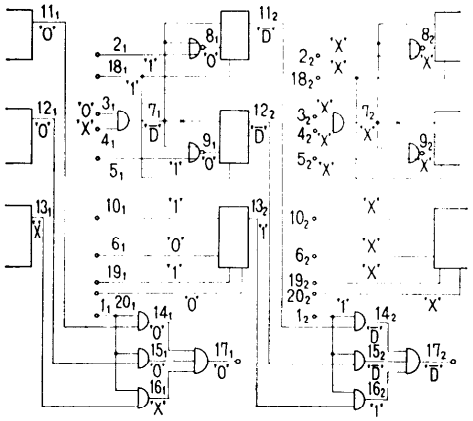

Fig. 4 Modeled form of Fig. 3

さて，時間が2に進み，ライン 14_2, 17_2 に 'D̄' を伝搬させるために，ライン 1_2, 15_2, 16_2 を '1' に決定する．しかし，このときの一意決定でライン 12_2, 13_2 が '1' と決まるが，ライン 12_2 の '1' を得る組合せは存在せず，ここで論理矛盾が発生する．これは，ライン 17_2 の D-intersection にともなう論理矛盾であるので，これ以後のブランチ（ライン 17_2, 15_2, 16_2, 12_2, 13_2）の論理値をすべて 'X' にし，現在のノードの次のブランチを選択する．結局，活性ベクトル**A**の次の要素であるライン 12_2 を選択する．これは，ライン 11_2 の場合と同じで，ライン 12_1, 9_1 を '0' にする唯一の組合せでライン 12_2 が 'D̄' となる．一意決定は可能で，ライン 5_1 が '1'，ライン 15_1 が '0'，ライン 15_2 が 'D̄' に決定される．次にライン 17_2 の

の D-intersection を行ない，ライン 16_2，13_2 の "1" が決定される．

ここでD操作を終了し，C操作に入る．

C操作を行なうべきラインは 7_1，13_2 の2個であり，13_2 に "1" を得る組合せは **Table 2** に示したものと同じで3通り存在し，タイプ5のノードを発生する（ノード番号2）．優先関数により3番目の組合せを選択し，ライン 10_1，19_1 を "1" にライン 6_1 を "0" にする．

次にライン 7_1 であるが，これは故障想定点であり，ここでは縮退 "1" 故障を考えているので，ライン 7_1 を "0" と考えればよく，タイプ4のノードが発生される（ノード番号3）．仮にライン 3_1 を "0" とすれば，他にC操作を行なうラインはなく，ここで全操作を終了する．

結局，**Fig. 4** に示すように全ラインの値が決まる．

3. プログラム

Fig. 5 に主要構造を示す．基本的には D-アルゴリズムと同様な構造となっており，D操作を行なう DDRIVE，C操作を行なう CDRIVE を有している．また，一意決定は UNIQD で制御を行ない，一意決定前進，後退をそれぞれ UQFOPE，UQBOPE で行なっている．

ここでは，われわれの手法の特長である FF 特性方程式(以下，論理式と呼ぶ.)の操作を行なう FFOPER の方法について主に述べる．

3.1 FFOPER

FFOPER とは FF に関するすべての操作

（1） 一意決定前進

（2） 一意決定後退，C操作

（3） D操作

を実際に行なうルーチンである．

これらの処理はすべて積の和型で表わされた論理式に基づいて行なわれる．以下に各種 FF の論理式を示す．

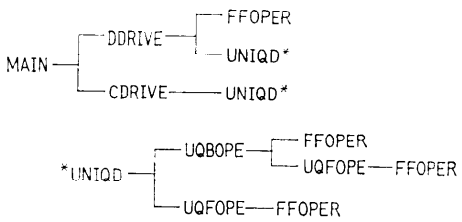

Fig. 5 Program structure of the pattern generating procedure.

○R-S 型

$$Q_{N+1}=\bar{C}\cdot S\cdot\bar{R}\cdot\phi+\bar{C}\cdot Q_N\cdot\bar{\phi}+\bar{C}\cdot Q_N\cdot\bar{R}. \quad (1)$$

○DELAY 型

$$Q_{N+1}=S\cdot\bar{C}\cdot\phi+Q_{N+}\cdot\bar{C}\cdot\bar{\phi}. \quad (2)$$

○LATCH 型

$$Q_{N+1}=Q_N\cdot R+\bar{S}\cdot R. \quad (3)$$

ただし，C はクリアー入力，また R-S 型はマスター，スレイブに展開して R-S 型にプリセット入力を付加したものと考える（**Fig. 6**）．

以下，前述の各操作方法について R-S 型FFを例として説明する．

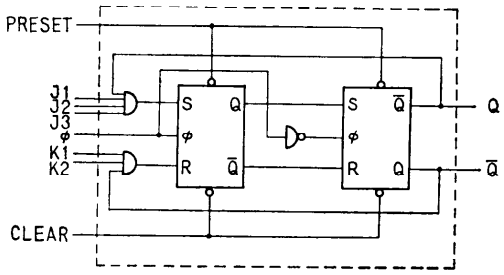

Fig. 6 Modeled form of J-K type flip-flop.

3.1.1 一意決定前進

一意決定前進は，与えられた条件で論理式を計算し出力値を求めれば良い．

〔例〕 **Fig. 4** の 11_2 の論理値を求める場合，$Q_N=0$，$S=D$，$R=0$，$\phi=1$，またクリアー入力はないので $C=0$ と見なし，11) 式に代入すると

$$Q_{N+1}=(1)\cdot(\bar{D})\cdot(1)\cdot(1)+(0)\cdot(0)\cdot(0)+(1)\cdot(0)\cdot(0)=(\bar{D}) \quad (4)$$

となり (N+1) 時点に "$\bar{D}$" が伝搬することが理解できる．

3.1.2 C操作，一意決定後退

C操作と一意決定後退の処理内容はほとんど同じである．まず，論理式にN時点での入出力条件を入れ，与えられた (N+1) 時点での出力値 Q_{N+1} を得るすべての組合せを求める．次に，C操作であれば複数個の解の組から優先関数にしたがって解を採用し，また，一意決定後退であれば，一意的に決定可能な変数（ライン）の論理値を求めればよい．

一般に次の手順で求めることができる．

（1） 与えられた入出力条件を論理式に代入する．

（2） 与えられた Q_{N+1} が "0" であれば，すべての項を "0" にするすべての可能な組合せを求める．

Q_{N+1} が 'I' であれば，どれか1項を 'I' にする組合せを可能なすべての場合について求める．

（3） 求まった解の中から論理的に出力値 Q_{N+1} を決定できない，いわゆる FF の禁止状態（R-S型で，S=1，R=1，ϕ=1 等）を起こすものを除く．

（4） 求まった解の組について

C操作のときは，優先関数値により採用順序を決定する．

一意決定後退のときは，①解が1個のみ存在する場合はその解が一意決定可能となり，②複数個存在する場合はすべての解に共通して一定値を持つ変数が存在するとき，その変数のみが一意決定可能となる．

〔例〕 **Fig. 4** のライン 13_2 の 'I' を得るには，Q_{N+1}=1，N時点でのすべての入出力値を 'X' とすれば

（1）式は

$$1=\bar{C}\cdot S\cdot\bar{R}\cdot\phi+\bar{C}\cdot Q_N\cdot\bar{\phi}+\bar{C}\cdot Q_N\cdot\bar{R}. \qquad (5)$$

したがって

$$\begin{pmatrix}\bar{C}\cdot\bar{R}\cdot\phi=1\\ \bar{C}\cdot Q_N\cdot\bar{\phi}=1\\ \bar{C}\cdot Q_N\cdot\bar{R}=1\end{pmatrix} \qquad (6)$$

であり，解は3通り存在し，

（1） C=0，R=0，ϕ=1，

（2） C=0，Q_N=1，ϕ=0，

（3） C=0，Q_N=1，R=0

となる．

C操作の場合には，これらの解を優先関数値にしたがって採用する．この場合は，（1）が最も高く，（2），（3）は同値だから（1），（2），（3），または（1），（3），（2）の順で解として採用される．

一意決定後退の場合，解が複数個存在して C=0 がすべての解に共通して存在するから，C=0 のみが一意決定されることになる．

3.1.3 操作

D操作は与えられた入出力条件を論理式に代入し，出力値 Q_{N+1} を 'D'，'$\bar{D}$' にするすべての組合せを求めるものである．

一般に次の手順で求めることができる．

（1） 与えられた入出力条件を論理式に代入する．

（2） 論理式中に存在するすべての 'D'（'$\bar{D}$'）に対して，以下の3手順を満足するすべての組合せを求める．

① 'D'（'$\bar{D}$'）を含む1つの項に対し，'D'（'$\bar{D}$'）以外の変数をすべて 'I' にする．

② 'D'（'$\bar{D}$'）を含まない項はすべて '0' にする．

③ '$\bar{D}$'（'D'）を含む項はすべて '0' とする．

（3） 求まった解の中で禁止状態で起こすものを除く．

（4） 求まった解を優関数にしたがって採用する．

〔例〕 **Fig. 4** のライン 11_2 に '$\bar{D}$' を伝搬させる場合を考える．

ライン 7_1 が '$\bar{D}$'，その他がすべて 'X' であるので，S=$\bar{D}$，C=0 を（1）式に代入すると

$$Q_{N+1}=(\bar{D})\cdot\bar{R}\cdot\bar{\phi}+Q_N\cdot\bar{\phi}+Q_N\cdot\bar{R}. \qquad (7)$$

したがって，'$\bar{D}$' を含む項は第1項のみであるので，第1項を '$\bar{D}$' にその他を '0' にするすべての組合せを求めればよい．

故に

$$\bar{R}\cdot\phi=1, \qquad (8)$$

$$Q_N\cdot\bar{\phi}=0, \qquad (9)$$

$$Q_N\cdot\bar{R}=0 \qquad (10)$$

を満足すればよい．

（8）式より R=0，ϕ=1 が一意的に決定され，これを（9），（10）式に代入すると R=0 だから（10）式は満足され，また，（9）式を満足するには Q_N=0 とすればよい．故にこの場合の解は1個のみ存在し，

$$R=0,\ \phi=1,\ Q_N=0 \qquad (11)$$

となる．

以上，特に R-S 型 FF の例で説明したが，その他の FF や AND-NOR などのマクロ・ゲートでも，特性方程式さえ与えられれば同様の操作で処理できる．

4. 結果

本手法は論理回路故障検出系列発生システムの中で系列発生の一手法として開発されたものであり，このシステムに組込み実現するために，幾つかの制限を設けているが，その実行結果としては想定故障数 100～

Table 3 Result of the pattern generating system.

型式	ゲート数	FF 数	発生系列数	(未検出故障数)/(全故障数)	故障検出率	演算時間*
SL**	40 (個)	8 (個)	68 (個)	0/118 (個)	100 (%)	40 (sec)
S***	109	13	67	3/318	99	70
S	154	9	104	2/382	99	180

* Time in the case of IBM 360/75 or UNIVAC 1108
** Sequential circuit with loop
*** Sequential circuit without loop

400で概ね故障検出率 96～100% を得ている．代表例を **Table 3** に示す．

未検出となった故障はほとんどが冗長性により本質的なものと，故障により FF を望む値にセットできずに検出不可能となったものである．

このシステムは，フォートランで書かれており，全システムのサイズ約 150 kW，ステップ数約 5.5 k である．

5. むすび

Table 3 で解るように本手法として，また，システムとして，故障検出率，演算時間ともに満足すべきものであり，充分実用に耐え得ると考えている．

しかし，本手法の改善のために，FF の処理方法の高効率化，すなわち，一意決定後退の場合，特性方程式を解かずに，$Q_N=1$，$Q_N=0$ とする論理値表を用意しておく等の方法を検討する必要があると思われる．

6. 謝辞

本手法を開発するにあたり，御協力下さった当社開発本部集積回路研究所，並びに，有線事業部制御プログラムグループの諸氏に感謝する．

参考文献

1) J. P. ROTH, et al.: Programmed Algorithm to Compute Tests to Distinguislh Between Failures, IEEE trans. on E. C. Vol. EC-16 No. 5 pp. 567～580, Oct. 1967.
2) H. KUBO: A Procedure for Generating Test Sequences to Detect Sequential Circuit Failures, NEC Re & Dev. No. 12, pp. 69～78, Oct. 1968.
3) J. P. ROTH, et al.: A Heuristic Algorithm for Sequential Circuit Diagnosis, IBM Re & Rep. RC 2639, Sep. 1969.

(昭和 47 年 8 月 28 日 受付)
(昭和 48 年 1 月 11 日再受付)