

自己組織化マップと必須割当て情報を用いた

同時検出可能故障の解析

日大生産工 ○武田 俊 日大生産工 山崎紘史 日大生産工 細川利典
日大生産工 新井雅之 日大生産工 山内ゆかり

1. はじめに

近年、半導体微細化技術の進歩により、大規模集積回路 (Very Large Scale Integrated circuits : VLSI) が大規模化、複雑化している。さらに故障モデルの多様化に伴い、テストコストの増加が問題となっている[1]。テストコストはテストのテスト実行時間に比例して増加するため、テストパターン数の削減などによるテスト実行時間の削減が求められている。テストパターン数の削減方法として、テスト圧縮[2]がある。テスト圧縮を用いてテストパターン数を削減することにより、テストコストの削減が期待できる。テスト圧縮とは、自動テストパターン生成 (Automatic Test Pattern Generator : ATPG) [3] により生成されたテスト集合のテスト品質を維持したままテストパターン数を削減する技術である。テスト圧縮には、テスト生成の過程で圧縮を行う動的圧縮[4][5]と、テスト生成後の初期テスト集合に対して圧縮を行う静的圧縮[2][5]がある。静的圧縮の1つとして、MバイN法が提案されている[6]。MバイN法は、テストパターンをN($N < M$)個生成し、初期テスト集合に追加することでM個のテストパターンを削除するテスト圧縮法である。しかしながら、テスト生成には決定的アルゴリズムを用いるため、1個のテストパターンで同時に検出可能な故障を効率よく分類しないとテスト圧縮時の処理時間が大きくなる可能性がある。それゆえ、ニューラルネットワークの一種である自己組織化マップ (Self organizing maps : SOM) [7]を用いることで、複数同時検出可能な故障を効率よく分類することが期待できる。

本論文では、初期テスト集合に依存しないテスト圧縮法としてSOMと故障検出の必須割当て情報を用いた新たな手法を提案する。本論文の構成は以下のとおりである。第2章では従来のテスト圧縮手法としてMバイN法について述べる。第3章では、自己組織化マップについて述べ、第4章では必須割当てについて述べる。第5章では提案手法であるSOMと必須割当て情報を用いた同時検出可能故障の解析について述べる。第6章では実験結果を示し、第7章ではまとめと今後の課題について述べる。

2. MバイNアルゴリズム

MバイN法は、初期テスト集合Tから削除するテストパターンをM個($M > N$)選択し、選択したテストパターンのみで検出可能な故障を検出するテストパターンをN個生成する。生成したテスト集合と選択したテスト集合を置き換えることにより、テストパターン数を削減するアルゴリズムである。以下に3バイ2法の例を用いて説明する。Tからテストパターンt1, t2, t3を選択した時、t1でのみ検出可能な故障f1, t2でのみ検出可能な故障f2, t3でのみ検出可能な故障f3としたとき、f1, f2, f3を対象としたテスト生成を実行し、故障f1, f2, f3を検出可能なテストパターンt4, t5を生成する。生成されたt4, t5とt1, t2, t3を置き換えることにより、テストパターン数を削減する。

3. 自己組織化マップ

自己組織化マップ(SOM)とはニューラルネットワークの一種で、教師なし強化学習と近傍学習により、多次元ベクトルを二次元マップなどに写像する。これによりSOMに与えたデータが類似度の高いデータごとにマップ上に配置される。この性質よりSOMは複雑なデータのクラスタリングに用いられる。SOMの入力層はi個のm次元のベクトルで表現され、競合層は $p \times q$ 個の任意の数の2次元配列で表現される。SOMの競合層の各ノードは、入力ベクトルと同じ次元(m次元)の重みベクトルを用いて対応している。SOMの概略図を図1に示す。

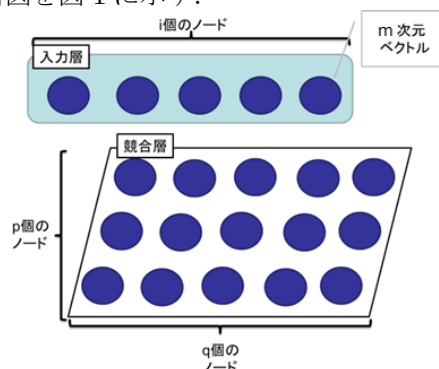


図1. SOMの概略図

An Analysis of Simultaneously Detectable Faults Using Self Organizing Maps and Necessary Assignment Informations

Shun TAKEDA, Hiroshi YAMAZAKI, Toshinori HOSOKAWA, Masayuki ARAI and Yukari YAMAUCHI

競合層のベクトルは、ランダムに初期化するか入力ベクトル集合の中央値に近いものを割当てる。SOMの学習は、入力ベクトルの特徴を繰り返し競合層に学習させることにより実行される。提示された入力ベクトルに対して、最も類似度の高い競合層ノードを勝者ノードと定義し、勝者ノードとその周辺のノードに対して重みベクトルを更新することで学習を実行する。重みベクトルの更新は、入力ベクトルに近づける方向に更新することで入力ベクトルの特徴を学習する。それにより、競合層には似た特徴を持ったノードが近くに集まる。本論文では、この性質を利用してテスト圧縮を実現する。

4. 必須割当て

必須割当てとは故障を検出するために一意に決定される信号線の論理値である。

例を図2に示す。

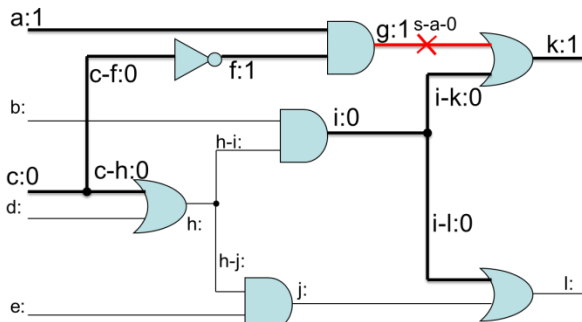


図2. gの0縮退故障の必須割当ての図

図2において、太線の信号線に割当てられている0または1の値が信号線gの0縮退故障の必須割当てである。

5. 自己組織化マップと必須割当て情報を用いたテスト圧縮

本章では、自己組織化マップと必須割当て情報を用いたテスト圧縮について説明をする。本手法は複数故障の同時故障検出テスト生成の対象とする故障を選択することを目標にSOMに必須割当て情報を与え、類似度の高い必須割当てを見つける。類似度の高い必須割当てを持つ故障を同時故障検出テスト生成の対象とすることによって、より効率的なテスト圧縮を目指す。

入力データのデータ長は対象回路の信号線数分存在し、4値で表現する。4値は必須割当ての0、必須割当ての1、必須割当てではないが、故障の励起または故障影響の伝搬、未正当化信号線の正当化に関する可能性のあるX1、必須割当てでなく、故障検出のための割当てが必要でないX2である。

例を図3に示す。

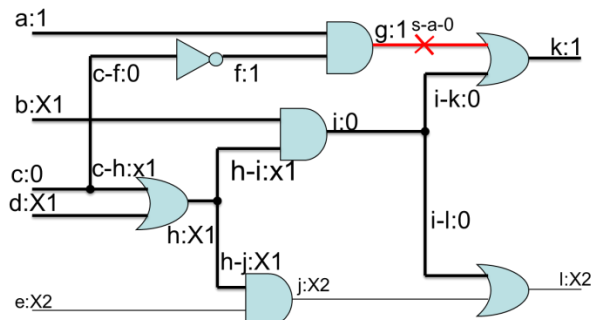


図3. gの0縮退故障の値割当ての図

各信号線には図3に示すように必須割当て(0または、1)、X1、X2のいずれか1つの値が割当てられる。図3の回路図中の太線の信号線が故障を励起および故障の影響の正当化に関する可能性のある信号線であるためX1が割当てられる。

SOMでは勝者ノードを入力データと最も類似度の高い競合層のノードとする。そのため、入力データと競合層のノードの各要素の差の2乗和で類似度を求めるユークリッド距離を用いるが、本論文では、SOMの入力層に与えるデータは必須割当て情報であるため従来手法で用いられているユークリッド距離[8]が近い競合層のノードを勝者ノードとしない。必須割当ては故障を検出するために一意に割当てられる値である。したがって、必須割当ての値が衝突した場合に必須割当ての類似度が低くなるためユークリッド距離は採用しない。本手法では必須割当ての0と1が衝突した回数が最も少ないノードを勝者ノードとする。また、0と1の衝突回数が同じだったノードが複数存在する場合、X1と必須割当て(0または、1)またはX1とX1の衝突とX2の衝突をペナルティ関数を用いて勝者ノードを選択する。

5-1. SOM アルゴリズム

提案したSOMのアルゴリズムを図4に示す。

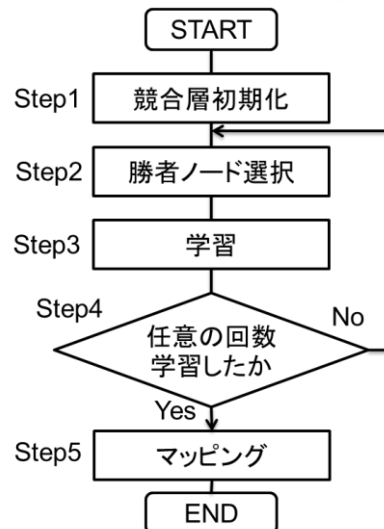


図4. SOM フローチャート図

Step1

入力データは $X=(x_1, x_2, \dots, x_n)$ である。 n は入力データの要素数を表す。 入力データ x_i は各信号線の情報 $l_{i1}, l_{i2}, \dots, l_{im}$ を全ての信号線数分保持し、 m は信号線数を表す。 信号線 k の信号線情報を持つ l_{ik} は 0, 1, X1, X2 の 4 状態に対応する 4 変数を持ち最も値が大きい変数を信号線 l_{ik} の状態とする。 入力データ集合 X の入力データをランダムに選択し、その値をもとに競合層の値を初期化する。

Step2

入力データ集合 X からランダムに入力データを 1 個選択し、入力ノード x_i とする。 x_i と類似度の高い競合層のノードを調べるために最初に競合層の各ノードとの信号線の必須割当ての衝突の回数を求める。 必須割当ての衝突の回数が最も少なかった競合層のノードを第 1 候補とする。 次に入力データと第 1 候補の競合層の各ノードの信号線情報の値の差を求める。 ただし求める値は競合層の信号線情報の 4 変数内で最も値が大きかった変数の値と入力ノード x_i の対応する信号線の変数の値の差のみである。 求めた差をもとにペナルティ関数からペナルティ値を求める。 ペナルティ関数は目標とする入力データと競合層ノードの差が大きいほどペナルティ値が急激に増大する。 また、様々回路規模の回路に対応できるように対象回路の信号線数をペナルティ値の最大とする。

図 5 にペナルティ関数の例を示す。

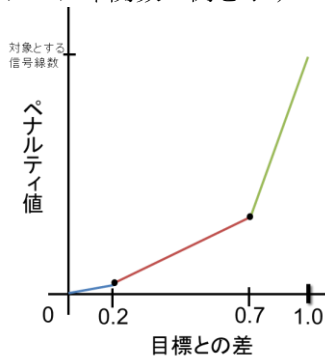


図 5. ペナルティ関数

次に差を求めた信号線で競合層と入力データで信号線の値に衝突が起きていないか判定をする。 必須割当ての 0, 1 の衝突が起きていた場合にはペナルティ値を大きく増大させ、 X1 の衝突を起こしていた場合にはペナルティ値を減少させる。 必須割当ての 1, 0 の衝突が起きていない場合か X2 が存在する場合にはペナルティ値を大きく減少させる。

信号線ごとにペナルティ値を求めてその合計値が最も低い競合層のノードを勝者ノード c とする。

Step3

勝者ノードとその周辺に x_i の値の学習を行う。 本論文では、(1)の学習式を用いて重みベクトルの更新を行った。 (1)式において、 t は現在の学習の繰り返し回数を表し、 $w_j(t)$ は現在(時刻 t)のノードの重みを表している。

$$w_j(t+1) = w_j(t) + h_{ci}(t) \left(x_i - w_j(t) \right) \quad (1)$$

$h_{ci}(t)$ は学習の強度を表現する近傍関数であり、(2)式により求められる。 (2)式において、 r_c は勝者ノード c の競合層上での座標、 r_j は競合層ノード j の座標をそれぞれ表している。 $|r_c - r_j|$ では差の絶対値を求めることで勝者ノード r_c との距離を求めている。 この差が小さいほどに学習係数 $h_{ci}(t)$ が小さくなり学習の影響が少なくなる。 また、 $\alpha(t)$ と $\sigma(t)$ は学習を制御するパラメータであり、本論文では学習回数の増加につれ 1 から 0 までの値をとる単調減少関数を用いた。 したがって、勝者ノードから近い参照ノードが強く学習する。

$$h_{ci}(t) = \alpha(t) \cdot \exp\left(\frac{|r_c - r_j|}{2\sigma^2(t)}\right) \quad (2)$$

Step4

指定された回数学習していない場合は Step2 へ。

Step5

入力データ集合の入力データに対して競合層内で最も類似度の高いノードを Step2 と同様の方法で選択する。 選択された競合層ノードに入力データを割当てる。 この処理を全入力データを行うことで競合層内のどの座標に入力データが存在すかわかる。 この処理をマッピングという。

6. 実験結果

SOM の実験結果を図 6 に示す。 ISCAS'89 ベンチマーク回路の s14207 の組合せ回路を対象とした。 対象故障は Synopsys 社の TetraMAX と乱数を用いて生成した 10,000 個のテストパターンで故障シミュレーションを実行し、検出回数 100 回以下の故障とした。

対象故障数 3273

学習回数 50,000 回

競合層のサイズ 60 × 60 の 36,000

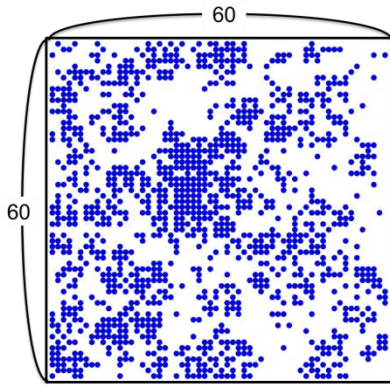


図 6. 実験結果

また、図 6 の同一の座標にマッピングされた故障同士に対して MTTG(Multi Target Test Generation) を行った。対象とした故障はランダムに選択した 2 つ以上の故障がマッピングされている競合層のノードとした。結果を表 1 に示す。

表 1. MTTG 結果

ノード名	対象故障数	成否
node1	3	○
node2	2	○
node3	48	×
node4	26	×
node5	75	×
node6	9	○
node7	2	○
node8	31	×
node9	3	○
node10	2	×
node11	2	○
node12	2	○
node13	2	○
node14	2	○
node15	3	×
node16	4	×
node17	2	×
node18	2	○
node19	4	○
node20	5	○
成功数		12/20

半分以上のノードで MTTG が成功している。また、対象故障数が多いノードは MTTG が失敗する傾向がみられる。

6. おわりに

M バイ N 法による圧縮のための必須割当てを用いた SOM を提案した。今後は MTTG がさらに成功するような SOM アルゴリズムを考えていきたい。

[参考文献]

- 1) Y.Sato, T.Ikeda, M.Nakao, and T.Nagumo, "Abist approach for very deepsub-micron (vds) defect," Proc. International Test Conference, pp. 283291, 2000.
- 2) K. Miyase, S. Kajihara and Sudhakar M. Reddy: "A Method of Static Test Compaction Based on Don't Care Identification," IPSJ Journal, Vol.43, No.5, pp.1290-1293,2002.
- 3) Tracy Larrabee: Test Pattern Generation Using Boolean Satisfiability, IEEE TRANSACTION ON COMPUTER-AIDEDDESIGN, VOL.11, No.1, pp4-15, 1992.
- 4) P.Goel and B.C.Rosales: "Test Generation and Dynamic Compaction of Tests", Digest of Papers 1979 Test Conf., pp.189-192, 1979.
- 5) S. Kajihara, I. Pomerantz, K. Kinoshita, and S. M. Reddy: "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Circuits," IEEE Transaction On Computer-Aided Design Of Integrated Circuits And System, Vol.14, No.12, pp1496-1504, 1995.
- 6) Ilker Hamzaoglu and Janak H. Patel: "Test Set Compaction Algorithms for Combinational Circuits", Computer-Aided Design, 1998. ICCAD 98, pp.283 - 289, 12 Nov. 1998
- 7) T. Kohonen: "Self-Organizing Maps", Berlin, Springer, 2010.
- 8) Per-Erik Danielsson: "Euclidean Distance Mapping", Computer Graphics and Image Processing, pp.227-248, 1980