

## 独立故障集合によるテストパターン数の下界評価

日大生産工(院) ○楠山 友紀乃 日大生産工 細川 利典

### 1. はじめに

近年, 半導体微細化技術の進歩に伴い, 大規模集積回路 (Large Scale Integrated circuit: LSI) が大規模化・複雑化している. それに伴い, テスタでテストする際のテスト実行時間と, テスタに記憶させるテストパターンのデータ量が増加し, テストコストが増大するという問題が発生している[1]. テストコストはテストパターン数と比例関係にあり, 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.

小規模回路では, ほぼ最小のテスト集合を得る静的圧縮と動的圧縮を適用したアルゴリズム[2]が提案されている. しかしながら, 大規模回路では最小のテスト集合を得るための計算量が多く, 現実的な計算時間での適用は困難である.

大規模回路に適用可能な手法として, 文献[3]では, 自動テストパターン生成(Aut -o Test Pattern Generation: ATPG)が生成した初期テスト集合に対しドントケア抽出[4], 頂点彩色問題を用いた極小解圧縮[5][6], 2重検出法[2]を繰り返し行う静的圧縮手法が提案されている. しかしながら, [3]の手法では, 初期テスト集合に対して静的圧縮を行うため, テスト圧縮効率が初期テスト集合に依存する. したがって, 初期テスト集合にテスト圧縮に非効率的なテストパターンが存在し, テストパターン数削減の妨げになる可能性があり, 最終テスト集合のテストパターン数の増加につながる. よって, 初期テスト集合の圧縮に非効率的なテストパターンを圧縮に効率的なテストパターンに変換, 再生成する, または圧縮に効率的な初期テスト集合を生成することにより, 最終テスト習合のテストパターン数が削減されることが考えられる.

本研究では, 文献[7]のように初期テスト集合を生成する時の動的テスト圧縮において1

次故障や2次故障をテストパターン中のケアビット箇所に基づいて選択することが重要であると考えられる. しかしながら, テスト圧縮の研究を行う上で, まず評価の対象となるベンチマーク回路のテストパターン数の下界を明らかにし, 定量的評価の目標とする必要がある.

文献[2]では ISCAS'85, ISCAS'89 ベンチマーク回路のテストパターン数の下界を, 独立故障集合[2]を用いて求めている. 本論文では, 文献[2]と同様に独立故障集合を用いて ITC'99 ベンチマーク回路のテストパターン数の下界を求め, テスト圧縮の研究を行うための定量的評価の目標を定める. 故障は組合せ単一縮退故障を対象とする.

### 2. 独立故障集合

独立故障集合とは, 同一のテストパターンで検出することが不可能な故障の集合である. 独立故障集合が大きいものから順にテスト生成目標故障として選択することで, より小さいテストパターン集合が得られると考えられる.

また, 独立故障集合の最大サイズを求めることによってテスト対象回路のテストパターン数の下界を求めることができる. テストパターン数の下界とは, テスト対象回路の検出対象故障をすべて検出可能なテストパターン集合の最小数以下の数値である. つまり, テストパターン数の下界を求めることで, テスト圧縮の研究を行う上での定量的評価の目標を定めることができる.

本論文では, 3章で示すように同時検出不可能故障グラフをクリーク分割することによって, 独立故障集合を求めている.

---

Evaluation of Lower Bound for the Number of Test Pattern

Using a Independent Fault Set

Yukino KUSUYAMA and Tosinori HOSOKAWA

### 3. 独立故障集合生成

#### 3-1. 同時検出不可能故障グラフ

同時検出不可能故障グラフとは、各故障を頂点として、同一のテストパターンで同時に検出することが不可能な故障を辺で接続したものである。例えば、あるテスト対象回路の故障 f1 と故障 f2 を考える。それぞれの故障を検出するために必要な信号値割当てを行う。図 1 に示すように、故障 f1 は信号線 c,e,h に 0 を割当て、信号線 b,g に 1 を割当て、f2 は信号線 a,d に 0 を割当て、信号線 b,e,f に 1 を割当てることが必要であると仮定する。それぞれの必須割当てを比較すると、信号線 e で衝突している。よって、同一のテストパターンで検出することが不可能な故障として同一テスト検出不可能故障グラフの頂点 f1 と f2 の間に辺を挿入する。

	a	b	c	d	e	f	g	h
f1	X	1	0	X	0	X	1	0
f2	0	1	X	0	1	1	X	X

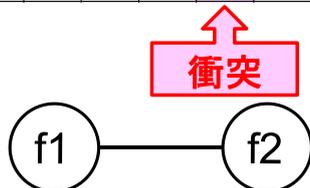


図 1. 同時検出不可能故障グラフ生成の例

図 2 に同時検出不可能故障グラフの例を示す。例として、あるテスト対象回路では検出対象故障が 5 個あり、それぞれの故障を f1, f2, f3, f4, f5 とする。図 2 において、f1 と f5, f1 と f2, f2 と f3, f2 と f4, f3 と f4 は同時検出不可能ということを表している。

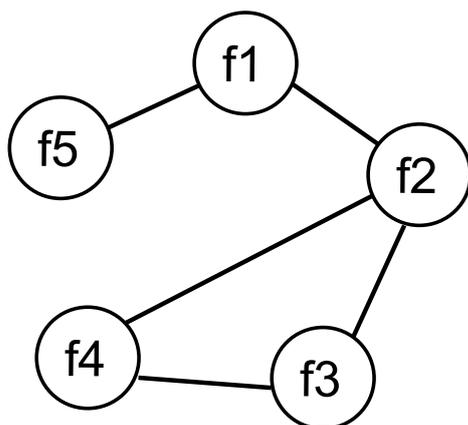


図 2. 同時検出不可能故障グラフの例

#### 3-2. 最大クリーク抽出

同時検出不可能故障グラフを作成した後、最大独立故障集合を求めるために最大クリーク抽出を行う。最大クリーク抽出とは、対象のグラフでクリーク分割を行ったときに、一つのクリーク内の頂点数であるクリークサイズが最大になるように、クリーク分割を行うことである。

図 3 に最大クリーク抽出アルゴリズムを示す。辺の数が最大の頂点が複数ある場合は、各辺数最大頂点で最大クリーク抽出を行い、クリークサイズを比較し、最大なものを最大クリークとする。

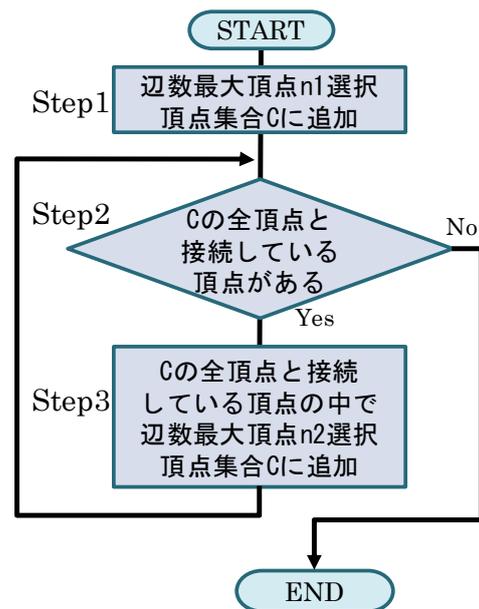


図 3. 最大クリーク抽出アルゴリズムのフローチャート

(step 1)

各頂点の辺の数を比較し、辺の数が最大の頂点を選択し n1 とする。n1 をクリーク頂点集合 C に追加する。

(step 2)

クリーク頂点集合 C に含まれるすべての頂点と接続している頂点が存在するか探索する。存在するならば step3 へ進み、存在しないならば終了する。

(step 3)

クリーク頂点集合 C に含まれるすべての頂点と接続している頂点の中で、辺の数が最大の頂点を選択し n2 とする。n2 をクリーク頂点集合 C に追加する。

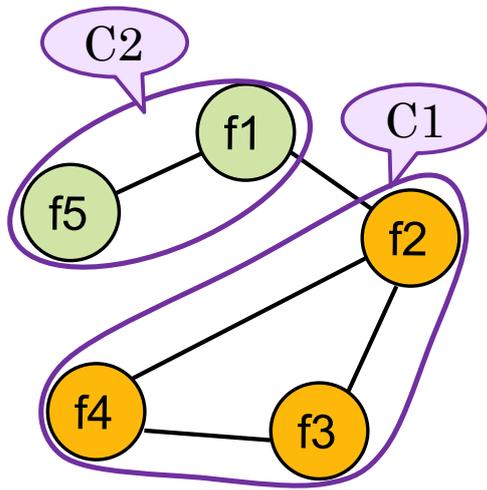


図 4. クリーク抽出した同時検出不可能故障グラフの例

図 4 に図 2 の同時検出不可能故障グラフを最大クリーク抽出した例を示す. クリーク C1 (f2, f3, f4) とクリーク C2 (f1, f5) の分割は解の一例である. この例では独立故障集合は 2 つで, 最大独立故障集合サイズは C2 の 3, つまりテストパターン数の下界は 3 個と求めることができる.

### 3-3. 独立故障集合生成アルゴリズム

図 5 に独立故障集合生成アルゴリズムを示す.



図 5. 独立故障集合生成アルゴリズムのフローチャート

(step 1)

テスト対象回路の等価故障解析を行い, 代表故障集合を求める.

(step 2)

各信号線に 0 又は 1 を割当て含意操作し, その結果から間接含意を静的学習[8]する.

(step 3)

step1 で生成した各代表故障に対し, その故障を検出するための必須割当てを行い, 含意操作を実行する. 各代表故障に対する信号線値の情報から同時検出不可能故障グラフを生成する.

(step 4)

step3 で生成した同時検出不可能故障グラフに対し最大クリーク抽出を行う. 抽出したクリークサイズが最大になるように, 辺が最も多い頂点から選択しクリーク抽出を行う.

(step 5)

step4 で同時検出不可能故障グラフに対し最大クリーク抽出を行った結果から, 最大独立故障サイズを求める. クリーク抽出後, クリークサイズが最大のクリークを探索する. 最大クリークのサイズが最大独立故障集合のサイズであり, テスト対象回路の下界が求まる.

## 4. 実験結果

独立故障集合生成を実装し, 実験を行った. 実験では ITC' 99 ベンチマーク回路を用いた. 表 1 は独立故障集合を用いた回路のテストパターン数の下界計算の実験結果である. 表 1 において「回路名」はテストパターン数の下界を計算する対象の回路名である. 「ゲート数」は対象の回路内の論理ゲートの数, 「外部入力数」は対象回路の外部入力信号線の本数, 「外部出力数」は対象回路の外部出力信号線の本数である. 「下界」は対象回路で独立故障集合生成を行い, 最大独立故障集合サイズを計算し, テストパターン数の下界を計算した結果である.

表 1 から, 回路の規模が大きく, ゲート数が多くなるほど最大独立故障集合サイズが大きくなるのが分かる. しかし, テストパターン数の下界とするには, 回路の規模に対して少ないものが存在する. これは同時検出不可能故障の判定が不十分で, 最大独立故障集合が正確にできていないのではないかと考えられる

回路名	ゲート数	外部 入力数	外部 出力数	下界 (最大独立故障 集合サイズ)
b01_C	42	7	7	9
b02_C	23	5	5	6
b03_C	127	34	34	7
b04_C	660	78	75	25
b05_C	927	35	70	52
b06_C	45	11	15	11
b07_C	391	50	57	29
b08_C	153	30	25	17
b09_C	141	29	29	9
b10_C	178	28	23	15
b11_C	732	38	37	48

表 1. 独立故障集合を用いたテストパターン数の下界

## 5. おわりに

本論文ではテスト圧縮指向テスト生成を提案するための前段階として、独立故障集合を用いて ITC'99 ベンチマーク回路のテストパターン数の下界を計算した。

今後の課題として、独立故障集合生成を改良し、より正確な回路のテストパターン数の下界を計算する。そして、求めたテストパターン数の下界をテスト圧縮の研究を行うための定量的評価の目標として定め、テスト圧縮のさらなる考察を行う。最終的な目標として、独立故障集合を利用したテスト圧縮指向テスト生成を提案することを目指す。

### 「参考文献」

1) Y.Sato, T.Ikeda, M.Nakao, and T.Nagumo, "A bist approach for very deep sub-micron (vds) defect," Proc. International Test Conference, pp. 283291, 2000.  
 2) Seiji Kajihara, Irith Pomeranz, Kozo Kinoshita, "Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits", 30th ACM/IEEE Design

Automation Conference, pp102-106,1993.

3) 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, May 2002.

4) K. Miyase, S. Kajihara "XID: Don't Care Identification of Test Patterns for Combinational Circuits," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, Vol. 23, No. 2, pp. 321-326, Fed. 2004.

5) 八木澤圭, "テストパターン数の静的圧縮における厳密解と貪欲解の比較" 電子情報通信学会, 107, pp.77-82, Feb. 2008.

6) D.Brelaz, "New methods to color the vertices of a graph", Communications of the ACM, 22, pp.251-256, 1979.

7) 山崎 達也, 細川 利典, 吉村 正義, 山崎 浩二, "テスト圧縮効率化のためのテスト生成に関する一考察" 第 64 回 FTC 研究, 2011 年 1 月 21 日.

8) Michael H. Schulz, Erwin Trischler, and Thomas M. Sarfert "SOCRATES: A Highly Efficient Automatic Test Pattern Generation System", IEEE Transactions on Computer-Aided Design, Vol.7, No.1, January 1988.