

複数目標故障テスト生成の高速化について

日大生産工(学部) ○澤田太生 日大生産工 細川利典
京産大 吉村正義 日大生産工 新井雅之

1. はじめに

近年、超大規模集積回路(Very Large Scale Integrated Circuits : VLSI)の大規模化、複雑化に伴い、テスト生成時間の増大が問題となっている。そのため、テスト生成の高速化が求められる。また、テストコスト削減のためテストパターン数削減手法として複数個の目標故障を対象に同時検出可能なテストパターン生成を試みる多重目標故障テスト生成 (Multiple Target Test Generation : MTTG)[1]-[7]が広く用いられている。

MTTGにおいて目標故障集合の選択はテストパターン数を削減する観点において重要である。小規模回路であれば、全ての未検出故障を目標故障集合に設定すると効率がよいと考えられる。目標故障集合の要素数を可能な限り増加させ、多数の故障を検出するようなテストパターンを生成することが理想である。しかしながら、目標故障集合の要素数が大きすぎる場合、テスト生成を効率よく実行することが難しいという問題がある。特に、目標故障集合が独立故障で構成される場合、その効率は著しく低下する。

本論文では、テスト生成の効率化のために、両立可能故障集合[8]を用いた複数目標故障テスト生成法について提案する。両立可能故障集合は必須割当てが衝突しない故障で構成される。その目的はテスト生成の際、できる限り同時検出される目標故障数を増加させるためである。

本論文では、多重目標テスト生成を擬似ブール最適化問題 (Pseudo-Boolean Optimization: PBO)[9][10]で定式化し、完全故障検出効率[11]を達成するテスト集合のテスト生成の高速化を目的としている。

本論文の構成は以下のとおりである。2章で前提知識について説明し、3章で両立可能故障グラフを用いた目標故障集合の選択について説明する。4章でまとめと今後の課題について述べる。

2. 前提知識

本章では、本論文で取り扱う用語を説明する。2.1節に必須割当て値の定義について述べる。2.2節に両立可能故障集合の定義について述べる。

2.1 必須割当て値[12]

本節では、必須割当て値の定義について説明する。必須割当て値とは、目標故障を検出するために必要な信号線の論理値である。一般に故障励起と含意操作、一意活性化[13]によって割当てられる。例として、図1に信号線 $i-l$ の0縮退故障の必須割当て値を示す。図1の回路は信号線 $i-l$ に0縮退故障を持ち、故障励起のため $i-l$ に1を割当てる。 $i-l=1$ の後方含意操作により $i=1$ が求まる。 $i=1$ の含意操作により $i-k=1$, $b=1$, $h-i=1$ が求まる。 $i-k=1$ の前方含意操作により $k=1$ が求まる。 $h-i=1$ の後方含意操作により $h=1$ が求まる。 $h=1$ の前方含意操作により $h-j=1$ が求まる。 D フロントティアゲートは信号線 l に隣接する or ゲートのみなので、一意活性化を行う。一意活性化により、 $j=0$ が求まる。更に含意操作を行い、 $e=0$ が求まる。これ以上、一意活性化も含意操作も行えないので必須割当てを終了する。以上より本例の必須割当て値は $(b,e,h,h-i,h-j,i,i-k,i-l,j,k)=(1,0,1,1,1,1,1,0,1)$ となる。

2.2 両立可能故障集合

本節では、両立可能故障集合の定義について説明する。両立可能故障集合とは、同じテストパターンで検出できる可能性がある故障の集合である。

図2に例題回路を用いて両立可能故障の例を示す。図2(a)は信号線 e の1縮退故障を検出するための必須割当てを表している。図2(b)は信号線 f の0縮退故障を検出するための必須割当てを表している。故障を励起させるために故障値と反対の論理値を割当て、故障の伝搬のために一意に決定される他の信号線の論理値を割当て、含意操作を実行することで必須割当てを求めることができる。0または1が割当てられている信号線は、故障を検出す

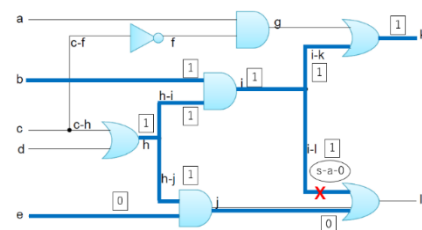


図1. 信号線 $i-l$ の0縮退故障の必須割当て値

On the Acceleration for Multiple Target Test Generation

Tao SAWADA, Toshinori HOSOKAWA,
Masayoshi YOSHIMURA and Masayuki ARAI

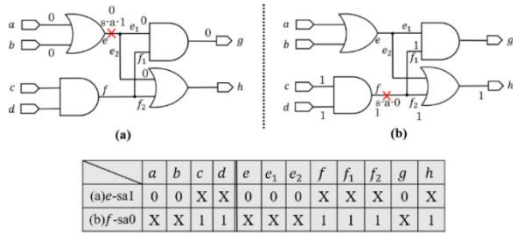


図 2. 両立可能故障の例

るためには必ずその論理値を割当てなければならない信号線を表し、X が割当てられている信号線は、0 または 1 のどちらかを割当てても故障の検出に影響がない可能性があることを表している。図 2 の故障ペアに対する必須割当てに衝突がないことから、同じテストパターンで検出できる可能性のある故障ペアと判断できる。図 2 の故障ペアの場合、同時検出可能なテストパターンは、(a,b,c,d)=(0,0,1,1) である。

3. 両立可能故障グラフを用いたテスト生成法

本章では提案手法である両立可能故障グラフを用いたテスト生成手法について説明する。3.1 節に PBO を用いた MTTG について説明する。3.2 節に本手法の全体アルゴリズムについて説明する。

3.1 PBO を用いた複数目標故障テスト生成

本節では、PBO を用いた MTTG について説明する。図 3 に PBO を用いた MTTG の全体アルゴリズムを示す。このアルゴリズムは対象回路 C と故障集合 F 、目標故障数 n_{TF} を入力とし、テスト集合 T 、検出不能故障集合 UD を出力する。本アルゴリズムではテスト集合 T と冗長故障集合 UD を空で初期化し(行 2)、故障集合 F が空になるまで、テスト生成と故障シミュレーションを繰り返し行う(行 3~13)。故障集合 F が空でない場合、まず、MTTG に

```

Input : Circuit  $C$ , Fault set  $F$ ,
         number of Target Faults  $n_{TF}$ 
Output: Test set  $T$ , Untestable Fault set  $UD$ 
1. PBO_base_MTTG( $C, F, n_{TF}$ ){
2.    $T = \emptyset, UD = \emptyset$ ;
3.   while( $F \neq \emptyset$ )
4.     ( $f, TF$ ) = Fault_Selection( $C, F, n_{TF}$ );
5.      $t$  = Test_Generation( $C, TF, f$ );
6.     if(Successful generation of  $t$ )then
7.        $F' = \text{Fault\_Simulation}(C, t, F)$ ;
8.        $F = F - F'$ ;
9.        $T = T \cup \{t\}$ ;
10.    else
11.       $F = F - \{f\}$ ;
12.       $UD = UD \cup \{f\}$ ;
13.    endif
14.  endwhile
15.   $T = \text{Double\_Detection}(T)$ ;
16.  return ( $T, UD$ );
17.}

```

図 3. PBO-MTTG 全体アルゴリズム

より同時に検出したい目標故障 $TF \cup \{f\}$ を信号線のレベルをキーとして昇順に n_{TF} 個だけ選択する(行 4)。ここで、 TF は二次故障集合とする。次に、回路情報と目標故障情報をもとに論理積標準形 (Conjunctive Normal Form: CNF) [8] の PBO インスタンスを生成し、PBO ソルバーを用いて PBO インスタンスを解き、テストパターン生成を行う(行 5)。テストパターン t が生成されたとき、テストパターン t で故障シミュレーションを行い、検出された故障 F' を F から削除する。生成されたテストパターン t はテスト集合 T に追加する(行 6~9)。それ以外の場合、1 次故障 f を検出不能故障集合 UD に分類し、 F から削除する(行 10~12)。故障集合が空になったら、二重検出法[14]を用いてテスト集合を圧縮し(行 15)、アルゴリズムの出力としてテスト集合を返す(行 16)。

3.2 MTTG で用いられる PBO インスタンス

本節では、本手法で用いられる PBO インスタンスについて説明する。PBO ベースの MTTG では、CUT を正常回路と故障回路からなるテスト生成モデルで構成する。故障回路は目標故障数分用意する。故障検出の判定は、正常回路と故障回路の外部出力値を比較することで可能である。これを PBO インスタンスとして表現するために、故障回路の外部出力とそれに対応した正常回路の外部出力を入力として XOR 演算を行うモデルを構築する。例として、図 1 の回路に対して、目標故障数 2 のテスト生成モデルを図 4 に示す。図 1 の回路は、2 つの外部出力があり、図 4 のモデルにおいて、各外部出力に対して XOR 演算を行う。また、故障はいずれか 1 つの外部出力で検出できていれば十分である。そのため、2 つの XOR 演算出力を OR 演算し、その出力 z が 1 になるような外部入力値を決定することでテストパターンを求めることができる。

PBO では最適化関数と制約を入力として問題を解くため、論理回路を含むテスト生成モデルをブール制約で表現する必要がある。基本ゲートにおけるブール制約表現を表 1 に示す。

テスト生成モデルにおいて故障の検出を行うためには、故障励起条件と故障検出条件の 2 つを充足することが必要となる。この 2 つの条件を合わせて故障検出制約とする。

信号線 A の 0 縮退故障を例とした場合では、 A の正常値は "11" となるため、正常回路においては $A=1$ は必須割当てである。また故障値は "00" であるため、故障回路では $A=0$ となる。一般的に故障励起条件を制約で表現したものを式(3.1)に示す。 G は正常回路中の目標故障信号線の論理値を示し、 F は故障回路中の目標故障信号線の論理値を示す。同様に、1 縮退故障の故障励起条件は式(3.2)で表現する。

$$G = 1, \bar{F} = 1 \quad (3.1)$$

$$\bar{G} = 1, F = 1 \quad (3.2)$$

$$G, F \in \{0,1\}$$

また，故障の検出を行うためには故障検出の判定で用いた XOR または OR 演算出力 OUT を”11”にすればよい．これを式(3.3)に示すように表現する．

$$\begin{aligned} \text{OUT} &= 1 \\ \text{OUT} &\in \{0,1\} \end{aligned} \quad (3.3)$$

しかしながら，MTTG において，複数存在する故障の故障検出制約を式(3.1)，(3.2)および(3.3)で定式化すると，すべての故障を必ず検出しなければならない．そのため，目標故障集合中に同一テストパターンで検出できない両立不可能な故障が含まれていた場合には，すべての制約を充足する入力が存在しないため，”UNSAT”となる．この問題を解決するため，両立不可能である可能性をもつ故障についての故障検出制約は式(3.4)，(3.5)および(3.6)で定式化する．Relax は緩和変数を表す．式(3.1)，(3.2)および(3.3)部分が”UNSAT”である場合に，Relax=1 を割当ててことで，”SAT”となる．

$$\text{Relax} + G \geq 1, \text{Relax} + F \geq 1 \quad (3.4)$$

$$\text{Relax} + G \geq 1, \text{Relax} + F \geq 1 \quad (3.5)$$

$$\text{Relax} + \text{OUT} \geq 1 \quad (3.6)$$

$$G, F, \text{OUT}, \text{Relax} \in \{0,1\}$$

目標故障集合中に両立不可能な故障が含まれていた場合は式(3.4)，(3.5)および(3.6)を用いることで解決することができる．ただし，緩和変数は二次故障にのみ導入する．しかしながら，PBO ソルバーが優先的に Relax=1 を割当てた場合，故障検出要求が無視される．これを避けるために，最適化関数として式(3.7)を設定する．N は緩和変数の個数すなわち二次故障数を示す．式(3.7)により，各 Relax に可能な限り”0”が割当てられるようになり，検出故障数を最大化することができる．

$$\text{Minimize} : \sum_{i=1}^N \text{Relax}_i \quad (3.7)$$

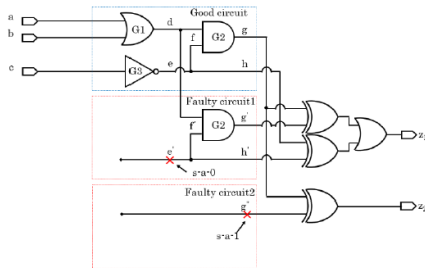


図 4. テスト生成モデルの例

表 1. 基本ゲート制約

Gate	Input		Output	Constraint
AND	x	y	z	$x + \bar{z} \geq 1$ $y + \bar{z} \geq 1$ $\bar{x} + \bar{y} + z \geq 1$
NAND	x	y	z	$x + z \geq 1$ $y + z \geq 1$ $\bar{x} + \bar{y} + \bar{z} \geq 1$
OR	x	y	z	$\bar{x} + z \geq 1$ $\bar{y} + z \geq 1$ $x + y + \bar{z} \geq 1$
NOR	x	y	z	$\bar{x} + \bar{z} \geq 1$ $\bar{y} + \bar{z} \geq 1$ $x + y + z \geq 1$
BUF	x		z	$\bar{x} + z \geq 1$ $x + \bar{z} \geq 1$
INV	x		z	$x + z \geq 1$ $\bar{x} + \bar{z} \geq 1$
XOR	x	y	z	$\bar{x} + \bar{y} + \bar{z} \geq 1$ $x + y + \bar{z} \geq 1$ $\bar{x} + y + z \geq 1$ $x + \bar{y} + z \geq 1$

3.3 両立可能故障グラフを用いた目標故障選択手法

本節では，両立可能故障集合生成アルゴリズムについて説明する．このアルゴリズムは，両立可能故障グラフに対して最大クリーク抽出問題[15]を解くヒューリスティックアルゴリズムである．ここで，両立可能故障グラフとは故障を頂点とし，必須割当て値が衝突しない故障ペアに対して，辺で接続した無向グラフである．図 5 に目標故障集合生成アルゴリズムを示す．

入力は単一縮退故障集合 F とする．出力は二次故障集合（両立可能故障集合） CF と一次故障 pf とする．はじめに CF を φ に初期化する(2 行目)． F を用いて，両立可能故障グラフ G_{CF} を生成する．グラフの頂点は単一縮退故障集合の各故障であり，各故障ペアの必須割当て値に衝突がなく，同時検出される可能性のある故障ペアの場合にその故障ペアに対応する頂点間に辺を接続する(3 行目)． G_{CF} が φ になるまで 5 行目から 7 行目の処理を行う(4 行目)． G_{CF} から最大次数を持つ頂点である故障 f を選択

```

Input : Fault set  $F$ 
Output : Compatible Fault set  $CF$ , primary fault  $pf$ 
1. Fault_Selection( $F$ ){
2.    $CF = \varphi$ ;
3.    $G_{CF} = \text{Create\_Compatible\_Fault\_Graph}(F)$ ;
4.   while( $G_{CF} \neq \varphi$ ){
5.      $f = \text{One\_Fault\_Selection}(G_{CF})$ ;
6.      $CF = CF \cup f$ ;
7.      $G_{CF} = \text{Update\_Graph}(G_{CF}, f)$ ;
8.   }
9.   ( $pf, CF$ ) =  $\text{select\_primary\_fault}(CF)$ ;
10.  return ( $pf, CF$ );
11.}

```

図 5. 目標故障集合選択アルゴリズム

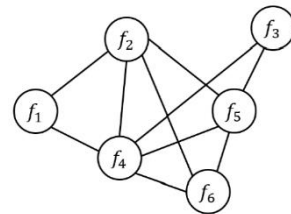


図 6. 両立可能故障グラフの例

する(5行目). CF に故障 f を追加する(6行目). G_{CF} から f の頂点と隣接していない頂点とその頂点に接続している辺を削除し, 次に f の頂点とそれに接続している辺を削除し, G_{CF} を更新する(7行目). CF から pf を選択し, CF から pf を削除し, 二次故障集合とする(9行目). 最後に, CF と pf を返す(10行目).

図6に示す両立可能故障グラフで目標故障選択例を示す. はじめに, G_{CF} の中で次数最大の頂点である故障 f_4 を選択して CF に追加する. G_{CF} から f_4 の頂点とそれに接続する辺を削除する. 次に, 更新された G_{CF} の中で次数最大の頂点である故障 f_2 を選択して CF に追加する. G_{CF} から f_2 の頂点に接続されていない故障 f_3 の頂点とそれに接続されている辺を削除する. 次に, f_2 の頂点とそれに接続されている辺を削除する. 次に, 更新された G_{CF} の中で次数最大の頂点である故障 f_5 を選択して CF に追加する. G_{CF} から f_5 の頂点に接続されていない故障 f_1 とそれに接続されている辺を削除する. 次に, f_5 の頂点とそれに接続されている辺を削除する. 最後に, 故障 f_6 を選択して CF に追加し, G_{CF} から f_6 の頂点を削除する. G_{CF} が $\varnothing$ になるので, 両立可能故障集合生成を終了する. 生成した両立可能故障集合は, $CF=\{f_4, f_2, f_5, f_6\}$ となる. CF が目標故障集合であるが, CF から一次故障 f_4 を選択し, CF から f_4 を削除し, 二次故障集合とする. よって, 目標故障集合選択アルゴリズムは $(pf, CF)=(f_4, \{f_2, f_5, f_6\})$ を返し終了する.

4. まとめと今後の課題

本論文では両立可能故障グラフを用いた目標故障選択手法を提案した. 本手法では必須割当て値の情報を与えることにより, MTTGの高速化を図った.

今後の課題としては, ランダムに目標故障を選択した場合とのテスト生成時間を比較し, 両立可能グラフを用いたテスト生成手法の効果を測る.

参考文献

[1]. G. J. Tromp, “Minimal Test Sets for Combinational Circuits,” Proc. ITC, pp. 204-209, Oct. 1991.
 [2]. Chang, Jau-Shien and Chen-Shang Lin. “Test set compaction for combinational circuits,” Proc. Proceedings First Asian Test Symposium (ATS '92), pp. 20-25, Nov. 1992.
 (DFT), pp.1-6, Oct. 2020.
 [3]. A. Chakradhar, K. Santhosh, “MULTIPLE TARGET TEST GENERATORS FOR OPTIMIZING THE ATPG PROCESS TO REDUCE NUMBER OF TEST SETS”, Proc. JETIR, vol 5, no. 1, pp.75-82, Jan. 2018.
 Tolerance in VLSI and Nanotechnology Systems
 [4] R. Asami, T. Hosokawa, M. Yoshimura, and M. Arai, “A Multiple Target Test Generation

Method for Gate-Exhaustive Faults to Reduce the Number of Test Patterns Using Partial MaxSAT”, Proc. IEEE International Symposium on Defect and Fault

[5]. Eggersglüß, Stephan, Kenneth Schmitz, Rene Krenz-Baath and Rolf Drechsler. “On Optimization-Based ATPG and Its Application for Highly Compacted Test Sets.”, Proc. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems vol. 35, pp. 2104-2117, Dec. 2016.
 [6]. Eggersglüß, Stephan, Kenneth Schmitz, Rene Krenz-Baath and Rolf Drechsler. “Optimization-based multiple target test generation for highly compacted test sets.”, Proc. 19th IEEE ETS, pp.1-6, May. 2014.
 [7] Eggersglüß, Stephan, Rene Krenz-Baath, Andreas Glowatz, Friedrich Hapke and Rolf Drechsler. “A new SAT-based ATPG for generating highly compacted test sets.” IEEE, 15th International Symposium on DEDCS, pp. 230-235, Apr. 2012.
 [8]松永裕介, “SAT ソルバによる両立故障集合検査を用いたテストパターン圧縮手法について”, Design Automation Symposium, 2015.
 [9]. V. Manquinho, R. Martins, and I. Lynce, “Improving Unsatisfiability-Based Algorithms for Boolean Optimization.”, Proc. International Conference on Theory and Applications of Satisfiability Testing, pp.1-12, July. 2010.
 [10] Smirnov, Pavel V., Jeremias Berg, and Matti Järvisalo, “Improvements to the Implicit Hitting Set Approach to Pseudo-Boolean Optimization.”, Proc. International Conference on Theory and Applications of Satisfiability Testing, July. 2022.
 [11]. 藤原秀雄, “ディジタルシステムの設計とテスト”, 工学図書株式会社, pp.146, May. 2004.
 [12]. Cox, Henry and Janusz Rajski. “On necessary and nonconflicting assignments in algorithmic test pattern generation.” Proc. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 13, no. 4, pp. 515-530, Apr. 1994.
 [13]Fujiwara, H ” FAN:A Fast Test Generation System for VLSI Circuits”, 明治大学工学部研究報告 第50号, May 1986.
 [14]. Seiji Kajihara, Irith Pomeranz, Kozo Kinoshita and Sudhakar M. Reddy. “Cost-Effective Generation of Minimal Test Sets for Stuck-at Faults in Combinational Logic Circuits.”, 30th ACM/IEEE Design Automation Conference, pp. 102-106, Jul. 1993.
 [15]富田悦次, 藤井利昭, “最大クリーク抽出の効率化とその実験的評価” 信学 (D), vol. J68-D, no. 3, pp. 221-228, mar. 1985.