

## 2 サイクルゲート網羅故障に対する複数目標故障テスト生成の高速化について

日大生産工（院） ○日向野隼多 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. まえがき

近年、半導体集積技術の発展に伴い、設計される超大規模集積回路(Very Large Scale Integrated circuits: VLSI)の大規模化、複雑化、高速化が急速に進展している。また、それに伴い不良 VLSI にはセルや信号線に物理的欠陥が存在し、論理回路においても論理機能が故障により別な論理機能に変化する論理故障や本来の実動作速度でテストした場合に誤った動作をするタイミング故障など様々な故障が存在する[1]。したがって、信号線における欠陥に加え、セル内の欠陥も増加している。本論文ではセル内の欠陥に着目する。セル内の欠陥はレイアウト解析をおこなうことで、欠陥が存在するセルに対してどの入力パターンでセルの出力信号線に故障励起可能か否かを判断する。セル内の欠陥の位置やタイプによっては特定の入力パターンでのみ故障励起可能な場合があり、縮退故障などの基本的な故障モデルに対するテスト集合では検出されない場合がある[2]。つまり、セル内の欠陥は縮退故障モデルなどに比べ検出が困難である。したがって、テスト時にセル内の欠陥を見逃す可能性があり、不良 VLSI を良品と判断するテストエスケープを引き起こす可能性がある[2]。

本論文では、セル内の欠陥によりセルの出力信号線の値が遷移する際に遅延が発生する故障に着目する。セルの出力信号線に遷移を発生させるセルの各 2 パターンの組合せに対して、その遷移が遅延する故障を 2 サイクルゲート網羅故障モデルと呼ぶ[3]。2 サイクルゲート網羅故障を検出するテストベクトルはセルの遅延欠陥を検出できる。本論文では、トランジスタ回路上の説明にはセル、デジタル回路上の説明にはゲートと呼ぶ。

2 サイクルゲート網羅故障数はゲート数とゲートの入力数に依存するため、大規模回路になるにつれ、故障数やテストベクトル数が増大する可能性がある。したがって、テスト生成における動的テスト圧縮[4]が重要となる。

動的テスト圧縮法の 1 つとして多重目標故障テスト生成 (Multiple Target Test Generation: MTTG) が提案されている[4-5]。MTTG とは、複数の故障を対象にテスト生成をおこなう手法である。ただし、目標と

した複数の故障は同時に考慮せず、それぞれ個別に故障の影響を考慮してテスト生成をおこなう。

文献[6]で 2 サイクルゲート網羅故障を対象とした MTTG 手法が提案されている。さらに、文献[6]では、生成したテストベクトルがキャプチャセーフ[7]であることを保証するために、テスト生成に擬似ブール最適化問題(Pseudo Boolean Optimization : PBO)[8]を用いて、テストベクトルによって遷移を引き起こす信号線を閾値未満にする条件を付加している。しかしながら、文献[6]では MTTG やキャプチャセーフを充足するための制約が多いため、テスト生成時間に課題を有する。

本論文では、MTTG において目標故障を一次故障と二次故障に分割してテスト生成のための制約条件を変えることにより、高速化を図る手法を提案する。ここで一次故障とは、必ず検出しなければならない 1 つの故障であり、二次故障は一次故障以外の目標故障である。二次故障の検出数を最大化するように、PBO の最適化関数の設定をおこなう。

本論文の構成は以下のとおりである。第 2 章では、2 サイクルゲート網羅故障モデルについて説明し、第 3 章では、2 サイクルゲート網羅故障に対する複数目標故障テスト生成の高速化手法を提案する。第 4 章では、実験結果について説明し、最後に第 5 章でまとめと今後の課題について説明する。

## 2. 2 サイクルゲート網羅故障モデル

2 サイクルゲート網羅故障[3]は、あるゲートの出力信号線に遷移を発生させるそのゲートの入力への各 2 パターンの組合せに対して、その繊維の遅延を引き起こす故障である。2 サイクルゲート網羅故障モデルはセル内の欠陥によりセルの出力信号線の値が遷移する際に遅延が発生する故障を網羅する。

表 1 に 2 入力 AND の 2 サイクルゲート網羅故障集合を示す。出力信号線の値が遷移する場合の入力への 2 パターンの組合せにおいて故障を定義するため、ゲートの入力数を  $n$  とした場合、2 サイクルゲート網羅故障数は  $2 \times (2^n - 1)$  個となる。ただし、2 入力 EXOR (EXNOR) ゲートの場合は  $2^2 \times 2 = 8$  個である。表 1 の 2 サイクルゲート網羅故障数は  $2 \times (2^2 - 1) = 6$  で

---

On the Acceleration of Multiple Target Test Generation  
for 2cycle Gate Exhaustive Faults  
Hayata HIGANO, Toshinori HOSOKAWA,  
Masayoshi YOSHIMURA and Masayuki ARAI

表 1. 2 入力 AND ゲートにおける  
2 サイクルゲート網羅故障集合

| Fault ID | input  |        | output |        |
|----------|--------|--------|--------|--------|
|          | 1-time | 2-time | 1-time | 2-time |
| $f_1$    | 00     | 11     | 0      | 1/0    |
| $f_2$    | 01     | 11     |        |        |
| $f_3$    | 10     | 11     |        |        |
| $f_4$    | 11     | 00     | 1      | 0/1    |
| $f_5$    | 11     | 01     |        |        |
| $f_6$    | 11     | 01     |        |        |

ある.  $f_1$ から $f_3$ は, 出力信号線が 0 から 1 に遷移する時間が遅延した故障であり,  $f_4$ から $f_6$ は出力信号線が 1 から 0 に遷移する時間が遅延した故障である. また, 各ゲートの 2 サイクルゲート網羅故障集合は同一のテストベクトルでは検出されない独立故障集合[6]である.

### 3. 2 サイクルゲート網羅故障に対する複数目標故障テスト生成の高速化手法

本章では, 本論文の提案手法である 2 サイクルゲート網羅故障に対する複数目標故障テスト生成の高速化手法を説明する. 3.1 節で PBO について説明し, 3.2 節で一次故障を必ず検出できるように考慮する制約の定式化について説明し, 3.3 節に 2 サイクルゲート網羅故障に対する複数目標故障テスト生成の高速化手法のアルゴリズムについて説明する. 本論文では, チャプチャセーフのための制約条件の説明は割愛する.

#### 3.1 PBO

擬似ブール最適化(Pseudo Boolean Optimization: PBO)は, 与えられたすべてのブール制約を充足し, 最適化関数を最小化するブール変数割当てを求める最適化問題である. 割当てが存在する場合は充足可能(Satisfiable: SAT)となり, そうでない場合は充足不可能(Unsatisfiable: UNSAT)となる.

#### 3.2 故障検出制約と故障伝搬制約

テスト生成を高速化するために, 目標故障集合を一次故障と二次故障に分割する. 一次故障を必ず検出し, 二次故障をなるべく多く検出するよう制約を定式化する. そのため, 緩和変数を用いて故障検出制約の定式化をおこなう.

式(1)に目標故障を $a, b, c$ の 3 つとしたテスト生成モデルにおける故障検出制約の定式化の例を示す. この時, 故障 $a$ を一次故障, 故障 $b, c$ を二次故障と考える.

$$\begin{aligned} Const_{det}: \quad & det_a \geq 1, r_b + det_b \geq 1, r_c + det_c \geq 1 \\ & r_b, det_b, r_c, det_c \in \{0,1\} \\ & minimize \quad r_b + r_c \quad (1) \end{aligned}$$

式(1)における $det_a, det_b, det_c$ は故障 $a, b, c$ における故障検出変数であり, 1 のときはその故障が検出可能, 0 のときは検出不可能であることを示す.  $r_b, r_c$ はそれぞれ故障 $b, c$ における緩和変数である. 式(1)において, 故障 $a$ が一次故障であるため,  $det_a$ は 1 にならなければ充足可能とならない. それに対して,  $det_b, det_c$ は緩和変数 $r_b, r_c$ を導入することでできるだけ 1 になるように定式化されている.

一次故障には緩和変数を導入しないため, 一次故障が必ず検出されるようにするために, 故障伝搬変数を導入する. 故障伝搬変数は一次故障の故障回路における各ゲートに対して導入する. 式(2)に故障回路のゲート $X$ における故障伝搬制約の例を示す. この時, ゲート $Y, Z$ をゲート $X$ のファンアウトとする.

$$\begin{aligned} Const_{prop}: \quad & (\overline{X_D} + X_G + X_F)(\overline{X_D} + \overline{X_G} + \overline{X_F}) \geq 1 \\ & (\overline{X_D} + Y_D + Z_D) \geq 1 \\ & X_D, X_F, X_G, Y_D, Z_D \in \{0,1\} \quad (2) \end{aligned}$$

式(2)における $X_D, Y_D, Z_D$ はそれぞれゲート $X, Y, Z$ における故障伝搬変数,  $X_G$ は正常回路のゲート $X$ の出力値,  $X_F$ は故障回路のゲート $X$ の出力値を表す. 故障伝搬変数は, 1 のときは故障の影響を伝搬していること, 0 のときは故障の影響が消滅していることを示す.  $X_D$ が 1 のときに, 正常回路および故障回路のゲート $X$ の出力である $X_G, X_F$ が異なる値であれば, 故障の影響が伝搬する. また,  $X_D$ が 1 のときに,  $X$ のファンアウトの故障伝搬変数である $Y_D, Z_D$ がどちらかが 1 であれば,  $X$ の故障の影響が $Y, Z$ の少なくともいずれかに伝播する. 上記の 2 つの条件を満たしたとき, 式(2)は充足可能である.

また, この故障を励起するためには, 故障伝搬変数にも故障励起条件が必要である. 式(3)に, 故障箇所のゲートを $X_{first}$ としたときの故障伝搬変数に関する故障励起制約を示す.

$$Const_{exc}: \quad X_{first_D} \geq 1 \quad (3)$$

式(3)における $X_{first_D}$ は故障箇所のゲート $X_{first}$ の故障伝搬変数を示す.

### 3.3 2 サイクルゲート網羅故障に対する複数目標故障テスト生成の高速化手法

本節では, 提案手法である 2 サイクルゲート網羅故障に対する複数目標故障テスト生成の高速化手法のアルゴリズムを述べる. 図 1 に全体アルゴリズムを示す. 入力回路  $C$ , 消費電力閾値  $WSA_{th}$  である. 出力は

図 1. 全体アルゴリズム

```

Algorithm Low-Power Oriented Multiple Target Test Generation for 2cycle Gate Exhaustive Fault
Input : Circuit C, WSA  $WSA_{th}$ 
Output : Test set  $T_{set}$ , Untestable fault set  $UTF_{set}$ , High-power Fault set  $HPF_{set}$ , Det fault set  $DF_{set}$ 

1.  $T_{set} = \emptyset$ 
2.  $F_{set} = 2cycle\_gate\_exhaustive\_fault\_set(C)$ ;
3.  $UTF_{set} = Untestable\_Delete(C, F_{set})$ ;
4.  $F_{set} = F_{set} - UTF_{set}$ ;
5.  $HPF_{set} = High\_Power\_Fault\_Delete(C, F_{set}, WSA_{th})$ ;
6.  $F_{set} = F_{set} - HPF_{set}$ ;
7. while( $F_{set} \neq \emptyset$ ) then
8.    $TF = Fault\_Selection(C, F_{set})$ ;
9.    $\phi PBO = PBO\_Test\_Generation(C, TF, WSA_{th})$ ;
10.   $Solution = Solver(\phi PBO)$ ;
11.   $tv = Generation\_Test\_Vector(Solution)$ ;
12.   $(F_{set}, DF_{set}) = Fault\_Simulation(C, tv, F_{set})$ ;
13.   $T_{set} = T_{set} \cup tv$ ;
14. endwhile
15. return( $T_{set}, UTF_{set}, HPF_{set}, DF_{set}$ );
16. end

```

テスト集合 $T_{set}$ , テスト不能故障集合 $UTF_{set}$ , アンセーフ故障集合 $HPF_{set}$ , 検出故障集合 $DF_{set}$ である。また、故障集合を $F_{set}$ , 目標故障集合を  $TF$ , 制約式を $\phi PBO$ , PBO ソルバーの解を  $Solution$ , テストベクトルを  $tv$  とする。はじめに、テスト集合を初期化する(1 行目)。次に、回路から 2 サイクルゲート網羅故障集合 $F_{set}$ を作成する(2 行目)。次に、故障ごとにテスト不能故障であるか否か判断し、テスト不能故障の場合は $UTF_{set}$ に追加する(3 行目)。次に、テスト不能故障と判断された故障を $F_{set}$ から削除する(4 行目)。次に、残りの故障からさらにアンセーフ故障を探索し、 $F_{set}$ から削除する(5, 6 行目)。次に、残りの故障を未検出故障集合とし、未検出故障集合がなくなるまで 8 行目～13 行目までの工程を繰り返しテスト生成をおこなう(7 行目)。はじめに未検出故障集合から目標故障  $TF$ を選択する(8 行目)。次に  $TF$ に対して制約式 $\phi PBO$ を作成する(9 行目)。次に、 $\phi PBO$ を満たすような解  $Solution$ を探索する(10 行目)。次に、 $Solution$ からテストベクトル  $tv$ を生成する(11 行目)。そして、 $tv$ に対して故障シミュレーションをおこない、検出された故障は $DF_{set}$ に追加し、 $F_{set}$ から削除する(12 行目)。最後に $T_{set}$ に  $tv$ を追加する(13 行目)。全ての未全ての未検出故障を検出した後、 $T_{set}$ ,  $UTF_{set}$ ,  $HPF_{set}$ ,  $DF_{set}$ を返し終了する(15 行目)。

提案手法では、一次故障を必ず検出し、二次故障をできるだけ多く検出することで、2 サイクルゲート網羅故障に対する複数目標故障テスト生成の高速化を図る。全体アルゴリズムは[6]と同じだが、図 1 の 9 行目において、 $TF$ から一次故障として選択した 1 個の目標故障に対して 3.2 節で説明した故障検出制約および故障伝搬制約を追加した制約式 $\phi PBO$ を作成する。

## 4. 実験結果

本章では、実験結果について説明する。提案手法は C 言語で実装され、Core i7-13700, 16GB メモリ搭載の PC を用いて実験をおこなった。対象回路は ISCAS'89 ベンチマーク回路である。ソルバーとして、Clasp3.3.4[9]を使用した。また、WSA 閾値は総信号線の 20%に

設定した。

表 2 に実験結果を示す。表 2 において左から回路名、2 サイクルゲート網羅故障数、MTTG で WSA 閾値を満たしテスト生成が成功された故障数、消費電力制約を入れずに充足不能になる故障数(図 1 の 3 行目)、故障検出率、故障検出効率、テストパターン数、テスト生成時間である。

表 2 から、全ての回路において、従来手法と比較して、テストパターン数を削減し、テスト生成時間を短縮できた。特に、テスト生成時間については、平均で 52.9%削減した。

## 5. まとめ

本論文では、2 サイクルゲート網羅故障に対する複数目標故障テスト生成の高速化手法を提案した。

実験結果から、2 サイクルゲート網羅故障をモデルとしたテスト生成の時間は平均で 52.9%削減した。

今後の課題は、2 サイクルゲート網羅の微小遅延故障故障をモデルとしたテスト生成手法が挙げられる。

## 参考文献

- 1) 藤原秀, デジタルシステムの設計とテスト, 工学図書株式会社, 東京, 2004, pp 135-135.
- 2) K.Y. Cho, S. Mitra, and E.J. McClusky, "Gate Exhaustive Testing", IEEE International Conference on Test, Austin, USA, Nov.2005, no.31.3, pp.1-7.
- 3) Irith Pomeranz, "Efficient Identification of Undetectable Two-Cycle Gate-Exhaustive Faults", Proc.of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, Volume:41 Issue 3.
- 4) G. Tromp, "Minimal Test Sets for Combinational Circuits," IEEE International Conference on Test, no.7.3, pp.204-209, Nashville, USA, Oct.1991.
- 5) J.S. Chang, and C.S. Lin, "Test set compact ion for combinational circuits," Browse Journals&Magazines, vol.14, Issue.11, pp.1370-1378, Nov 1995.
- 6) 溝田桃菜, 細川利典, 吉村正義, 新井雅之"擬似ブール最適化を用いた2サイクルゲート網羅故障のための低消費電力指向服す目標故障テスト生成法"ディペンダブルコンピューティング研究会, 2024年2月, 信学技法, vol 123, no.389, DC-2023-99, pp.29-34
- 7) X. Wen, K. Miyase, S. Kajihara, H. Furukawa, Y. Yamato, A. Takashima, K. Noda, H. Ito, K. Hatayama, T. Aikyo and K. K. Saluja, "A Capture-Safe Test Generation Scheme for At-Speed Scan Testing," Proc. ETS, p p. 55-60, 2008
- 8) Vasco Manquinho, Ruben Martins, and Ines Lynce, "Improving Unsatisfiability-Based Algorithms for Boolean Optimization," Theory

and Applications of Satisfiability Testing-SAT 2010

9) M. Gebser, B. Kaufmann, A. Neumann, and

T. Schaub, "Conflict-Driven Answer Set Solving," IJCAI, pp.386-392, Hyderabad, India, January 2007

表2. 実験結果

| 回路名    | 全故障数  | 検出故障数 | テスト不能故障数 | 故障検出率 [%] | 故障検出効率 [%] | テストパターン数 |      | テスト生成時間 |      |
|--------|-------|-------|----------|-----------|------------|----------|------|---------|------|
|        |       |       |          |           |            | 提案       | 従来   | 提案      | 従来   |
| s208   | 508   | 25    | 483      | 04.92     | 100.0      | 4        | 4    | 0.35    | 1.00 |
| s298   | 1056  | 131   | 925      | 12.40     | 100.0      | 19       | 21   | 0.42    | 1.00 |
| s344   | 799   | 352   | 447      | 56.04     | 100.0      | 53       | 62   | 0.48    | 1.00 |
| s349   | 802   | 350   | 452      | 43.64     | 100.0      | 44       | 63   | 0.50    | 1.00 |
| s382   | 1192  | 227   | 965      | 19.04     | 100.0      | 17       | 24   | 0.47    | 1.00 |
| s5378  | 13502 | 6556  | 6946     | 48.56     | 100.0      | 411      | 672  | 0.43    | 1.00 |
| s9234  | 23102 | 13533 | 9569     | 58.58     | 100.0      | 1054     | 1363 | 0.55    | 1.00 |
| s13207 | 33306 | 17119 | 16187    | 51.40     | 100.0      | 743      | 1165 | 0.43    | 1.00 |
| s15850 | 37856 | 16749 | 21107    | 44.24     | 100.0      | 258      | 304  | 0.56    | 1.00 |
| s35932 | 80946 | 35901 | 45045    | 44.35     | 100.0      | 131      | 191  | 0.52    | 1.00 |