

回路構造情報を用いた SAT ベーステスト生成の高速化

日大生産工（学部）○川連 裕斗

日大生産工

細川 利典

日大生産工（院）

山崎 紘史

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 circuits: LSI）が大規模化・複雑化し、テストパターンの生成時間が増加している。また、経路活性化法に基づくアルゴリズム[1]では、論理段数の深いかつ再収斂ゲートの多い大規模回路に対しては、現実的な時間で 100%の故障検出効率を得るテストパターンを生成することが困難となっている。テスト生成中、バックトラック[2]の制限値を設定し、その制限値内でテストパターンが生成できない検出困難故障に対しては、テストパターンの生成を途中で打ち切っている。

近年では SAT-solver の処理速度の高速化に伴い、回路の等価性検証やテスト生成技術など論理関数を処理する分野で、充足可能性問題（Satisfiability problem: SAT）[2][3][4]を利用した手法が提案されている。

SAT を用いたテスト生成はできるだけ多数の故障伝搬に必須な割当てを条件式に追加することで高速化することができる。本論文では SAT を用いたテスト生成に対して文献[5]で提案された絶対支配信号線（Absolute Dominator）の情報を抽出するアルゴリズムを用いて、一意活性化のための必須割当て条件式を乗法標準形（Conjunctive Normal Form: CNF）に追加することで処理の高速化を図る。

2. 充足可能性問題の解法を用いたテスト生成

SAT とは、CNF が与えられたときに、全ての変数の値を 1（真）または 0（偽）のどちらかに定めることで、式全体の値を 1（真）にできる割当てが存在するか否かを判定する問題である。CNF を 1（真）に

きる割当てが存在した場合、充足可能といい、与えられた CNF が正当化可能であることを示す。

SAT は CNF を入力として問題を解決するため、SAT を用いてテストパターンの生成を行うためには、論理回路を CNF に変換する必要がある。

2.1. 論理回路の CNF 変換

表 1 に各論理ゲートの CNF 変換規則を示す。各論理ゲートは表 1 の規則に従い CNF に変換する。それぞれの括弧でくくられた論理和式を節と言い、節を論理積をとった式が各ゲートの CNF である。例として 2 入力 (X, Y) 1 出力 (Z) の AND ゲートを考える。全体の値が 1（真）になるのは (X, Y, Z)=(0, 0, 0), (1, 0, 0), (0, 1, 0), (1, 1, 1) の 4 通りだけである。これにより、CNF が実際の論理ゲートの動作を表現していることがわかる。回路全体の CNF は各論理ゲートの動作を表現していることがわかる。

正当化操作は、未正当化信号線に値を割当てていく処理である。SAT の解法を用いることでどれかひとつでも真になりえないと判断できた時点で、その割当ての組み合わせが正当化不可能であることがわかる。したがって、効率的にかつ早期に割当て方の矛盾を発見でき、高速な正当化可能判定処理が可能である。

表 1. 論理ゲートの CNF 変換規則

ゲートタイプ	入力	出力	CNF
AND	X Y	Z	$(\neg Z + X) \cdot (\neg Z + Y) \cdot (\neg X + \neg Y + Z)$
OR	X Y	Z	$(Z + \neg X) \cdot (Z + \neg Y) \cdot (X + Y + \neg Z)$
EXOR	X Y	Z	$(\neg X + Y + Z) \cdot (X + \neg Y + Z) \cdot (\neg X + \neg Y + Z) \cdot (X + Y + \neg Z)$
NOT	X	Y	$(X + Y) \cdot (\neg X + \neg Y)$
FOUT	X	Y Z	$(\neg X + Y) \cdot (X + \neg Y) \cdot (\neg X + Z) \cdot (X + \neg Z)$

On Acceleration of SAT-Based Test Generation Using Information for Circuit Structures

Yuto KAWATSURE, Toshinori HOSOKAWA, and Hiroshi YAMAZAKI

2.2. SAT を用いたテスト生成

初めにテスト対象となる回路に対して，正常動作する回路と，ある 1 本の信号線が縮退故障[1]である故障回路を用意する．そして，正常回路と故障回路の入力に共通の外部入力を接続する．外部出力に対しては，故障の影響を励起・伝搬させるために片方の外部出力を真，もう一方を偽にさせる必要があるため，それぞれの外部出力を EXOR ゲートの入力に接続する．

故障の検出は少なくともどれかひとつの外部出力へ故障の影響が伝搬できればよいため，各外部出力を接続した EXOR ゲートの出力を OR ゲートの入力に接続し，この出力が常に 1（真）になる情報を付加する．故障情報は故障信号線に対して，外部入力側に正常値，外部出力側に故障値を設定することで示す．

図 1 に SAT を用いたテスト生成モデルの例を示す．図 1 の G0 から G6 が正常回路であり，G0' から G6' が故障回路である．AND ゲート G4' の出力信号線に 0 縮退故障があるとする．このため AND ゲート G4' の出力が正常値 1 となるように設定し，OR ゲート G6' は，G4' からの入力が故障値 0 になるように設定する．

a, b, c, d, e は外部入力であり，これらは正常回路と故障回路の入力に接続している．正常回路の出力は m と n，故障回路の出力は m' と n' である．これらの出力 m と m'，n と n' をそれぞれ EXOR ゲートの入力に接続し，EXOR ゲートの出力が x と y である．EXOR ゲートの出力である x と y を OR ゲートの入力に接続し，その OR ゲートの出力が z であり，故障の励起・伝搬のために 1 を設定する．

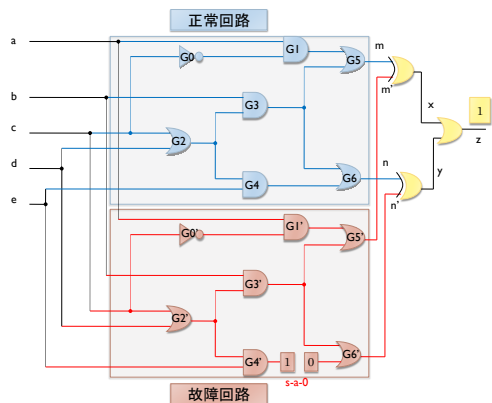


図 1. SAT を用いたテスト生成モデル回路例

3. 一意活性化を用いたテスト生成

回路中に D フロンティア[1]が唯一のとき，D フロンティアから外部出力へ伝搬する故障の影響が必ず通過しなければならないゲートに対して値を設定することを一意活性化[1]という．

一意活性化は必須割当てにより，解探索空間を削減することができ，早期の矛盾発見やバックトラック数の削減につながる．そのため，テスト生成モデルに対して作成した CNF に，一意活性化に必要な必須割当て情報を CNF に制約式として追加することで高速なテスト生成が可能となる．

一意活性化に必要な必須割当て情報を高速に抽出可能な手法が文献[5]で提案されている．一意活性化に必要な必須割当て情報は絶対的支配信号線と呼ばれている．

3.1. 絶対的支配信号線

絶対的支配信号線とは，縮退故障を持つ信号線に対して，故障影響が外部出力で検出できるまでに必ず通過する信号線のことを指す．

図 2 の回路において，信号線 a に縮退故障があると仮定すると，外部出力まで故障影響を伝搬するためには信号線 e と信号線 h を必ず通過しなければならない．すなわち，信号線 a の絶対的支配信号線は e と h となる．同様に信号線 b 以降の絶対的支配信号線を示したものが表 2 である．

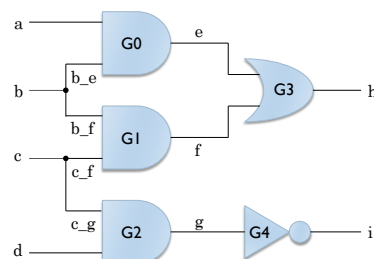


図 2. 組合せ回路例

表 2. 絶対的支配信号線の例

信号線	絶対的支配信号線
a	e, h
b	h
c	-
d	g, i
e	h
f	h
g	i

3.2. 絶対的支配信号線探索アルゴリズム

図 3 に絶対的支配信号線探索のアルゴリズムを示す。

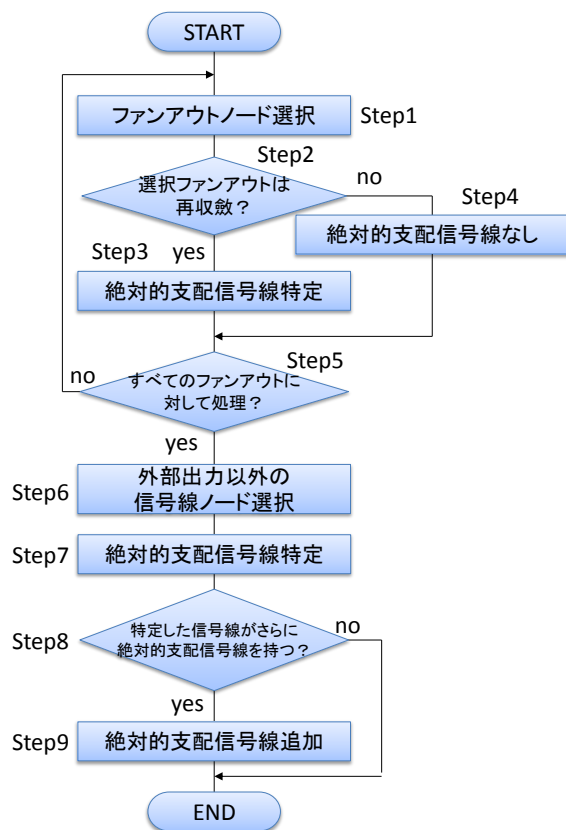


図 3. 絶対的支配信号線探索アルゴリズム

(Step1)

ファンアウトをレベルの高い方(外部出力に近い方)から一つ選択。

(Step2)

選択したファンアウトが再収斂していれば Step3 に進み、そうでなければ Step4 に進む。

(Step3)

再収斂した信号線が選択したファンアウトノードの絶対的支配信号線である。

(Step4)

選択したファンアウトノードは絶対的支配信号線を持たない。

(Step5)

すべてのファンアウトに対して処理をしたのであれば Step6 に進み、そうでなければ Step1 に戻る。

(Step6)

外部出力以外の信号線をレベルの高い方から一つ選択。

(Step7)

選択した信号線の出力先は、選択した信号線の絶対的支配信号線である。

(Step8)

絶対的支配信号線として特定した信号線が、さらに絶対的支配信号線を持つならば Step9 に進み、そうでなければ処理を終了する。

(Step9)

特定した信号線の持つ絶対的支配信号線を、選択した信号線ノードの絶対的支配信号線として追加する。

3.3. 絶対的支配信号線を用いた SAT テスト生成

ある縮退故障の影響を外部出力で検出するためには、縮退故障を起こしている信号線の絶対的支配信号線を持つゲートにおいて、絶対的支配信号線ではない入力(サイドインプット)に非制御値を割当てて必要がある。このサイドインプットに非制御値を割当てるという情報を制約式として CNF に追加することで高速なテスト生成が可能となる。

例として、図 2 の回路について考える。信号線 a に 0 縮退故障があると仮定すると、a の絶対的支配信号線は e と h であるので、そのサイドインプットである b_e と f にはそれぞれ非制御値である 1 と 0 を割当てて必要がある。これを制約式として CNF に追加する。

4. 故障解空間抽出

ある縮退故障に対し、その故障影響が伝搬可能な外部出力を探索する。その後、故障影響が伝搬する外部出力の値を決定する外部入力を探索し、発見した外部入力から外部出力までの解空間を抽出する。これにより、解空間の縮小ができ、CNF 生成時間と SAT-solver 実行時間が削減され、テスト生成時間の短縮が可能となる。

さらに、故障影響が伝搬する外部出力の中から一つ

ずつ選択し、その外部出力の値を決定し得る外部入力までの解空間を抽出することで、更なるテスト生成時間の短縮が目指せる。このとき、ある外部出力で抽出した解空間で充足可能であれば検出可能故障であり、充足不可能であればその外部出力では検出不可能であることを示す。すべての伝搬可能な外部出力に対して充足不可能となれば、その故障は冗長故障である。

故障伝搬可能な外部出力を1つずつ選択し、解空間を抽出して CNF を生成していく過程で、第3章で説明した絶対的支配信号線探索を行う。故障影響が伝搬する外部出力が1つに限定されるため、より多くの絶対的支配信号線が発見でき、テスト生成時間をさらに短縮することができる。

5. 実験結果

本論文では実験として、SAT に 3.3 章で説明した絶対的支配信号線探索による CNF への制約式を追加した場合と、制約式を追加しなかった場合とで比較をした。実験環境は、CPU は Intel Core2 DUO、メモリは 4GB、OS は Windows7 である。実験対象として ITC99 ベンチマーク回路の b01~b04, b06-b07, b11, b13 の組合せ回路版を使用。表 3 に実験結果を示す。

実験結果より絶対的支配信号線情報を用いることで、衝突回数が平均で 0.45 倍に縮小、最大で 0.34 倍に縮小。テスト生成時間が平均で 0.92 倍に短縮、最大で 0.46 倍に短縮された。

表 3. 実験結果

一意活性化なし			絶対的支配信号線探索		
回路名	衝突回数	テスト生成時間	回路名	衝突回数	テスト生成時間
b01.C	297	0.09	b01.C	169	0.10
b02.C	47	0.04	b02.C	26	0.03
b03.C	2467	0.71	b03.C	1100	0.78
b04.C	233005	57.82	b04.C	80426	27.14
b06.C	146	0.09	b06.C	69	0.11
b07.C	39493	11.53	b07.C	23876	9.83
b11.C	159217	47.67	b11.C	102357	37.53
b13.C	3320	2.19	b13.C	1680	2.49

6. おわりに

本論文では SAT を用いたテスト生成に対して絶対支配信号線の情報を抽出するアルゴリズムを用いて、一意活性化のための必須割当て条件式を CNF に追加することで処理の高速化を図った。実験結果から、小規模回路に対してはテスト生成時間の短縮はできなかったが増加分は微々たるものであり、大規模回路に対してはテスト生成時間の短縮に大幅に成功した。よって、絶対的支配信号線情報を用いる本手法は有効であるといえる。今後の予定として提案手法の更なる高速化、また、今回実験できなかった第4章の故障解空間抽出を実装し、実験・評価を行う予定である。

「参考文献」

- 1) 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, (2004), pp135~175.
- 2) Tracy Larrabee, “Test Pattern Generation Using Boolean Satisfiability”, IEEE TRANSACTION ON COMPUTER-AIDEDDESIGN, VOL.11, No.1, (1992), pp4~15.
- 3) Paul Stephan , Robert K.Brayton , Alberto.Sangiovanni-Vincentelli, “Combinational Test Generation Using Satisfiability” , IEEE.TRANSACTION.ON.COMPUTER-AIDEDDESIGN, VOL.15, No.9, (1996), pp1167~1176.
- 4) Paul Tafertshofer, Andreas Ganz, “SAT Based ATPG Using Fast Justification and Propagation in the Implication Graph”, IEEE/ACM international conference on Computer-aided design, (1999), pp139~146.
- 5) Xiaojing Hu, Zhengxiang Song, Jianhua Wang, Yingsan Geng, Wang Shen, “A fast and practical algorithm for absolute dominators searching” , IEEE Circuits and Systems International Conference on Testing and Diagnosis, (2009), pp1~4.