ISSN 2186-5647

ルックバック手法を用いた組合せテスト生成の高速化

日大生産工　○鈴木　悠介　　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 Integratied circuits: LSI）が大規模化し，テスト生成時間の増加や故障検出率の低下が問題となっている．

この問題を解決するためには，スキャン設計のようなテスト容易化設計を用いて論理設計を行うとともに，さらに効率の良い高速なテスト生成アルゴリズムを考案することが必要である．

アルゴリズムが単純でかつ効率的なテスト生成アルゴリズム[1]の一つとして SPOP [2]が提案されている．しかしながら，回路の大規模化に伴い，現実的な時間で 100%の故障検出効率を得るテスト生成をするのは困難である．

テスト生成の高速化の手法として静的学習[3]，衝突学習[4]，再帰学習[4]などの導入が提案されている．しかしながら，これらの学習処理はオーバヘッドの増加が問題となっている．

そこで本論文では，バックトラックに注目する．従来手法の順序式バックトラックでは矛盾が発生した場合，一つ前の選択肢に戻って処理を続ける．しかしその近傍に解が存在しない場合には無駄なバックトラックが増加し，テスト生成時間の増加や故障検出率の低下につながる．このような場合に対して効果が高いルックバック手法[5]を SPOP に導入し，テスト生成時間の削減，故障検出率の増加を図る．

2.　SPOP アルゴリズム

図 1 に SPOP のアルゴリズムを示す．

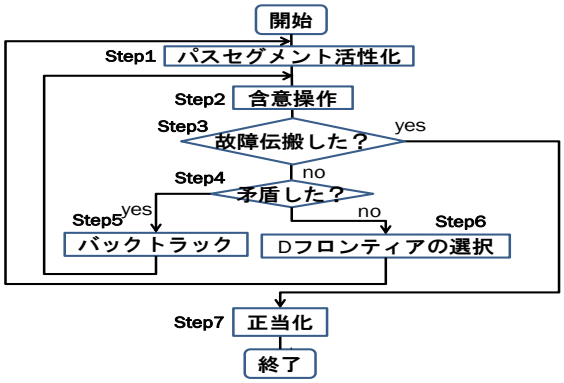

図 1.SPOP アルゴリズム

(Step1)

故障から出力側へ最も近い分枝元信号線までパスセグメント[2]を活性化する．

(Step2)

活性化された経路を基に含意操作[1]を行う．

(Step3)

故障の影響が外部出力へ伝搬しなければ Step4,していれば Step7 の処理を行う．

(Step4)

含意操作の結果に矛盾があれば Step5，なければ Step6 の処理を行う．

(Step5)

直前の選択にバックトラックする．

(Step6)

D フロンティアの選択[2]を行う．どの D フロンティアの選択を行ったかを判定木に情報として積む．

(Step7)

未正当化信号線を正当化し，処理を終了する．

On the Accleration of Combinational Test Generation Usinga Look-Back Technique

Yusuke SUZUKI and Toshinori HOSOKAWA

3. SPOP の故障伝搬処理の判定木

本章では SPOP の判定木について説明する. 図 2 に SPOP の故障伝搬処理の判定木の例を示す. ここで決定レベルは D フロンティアの選択した順番を示している.

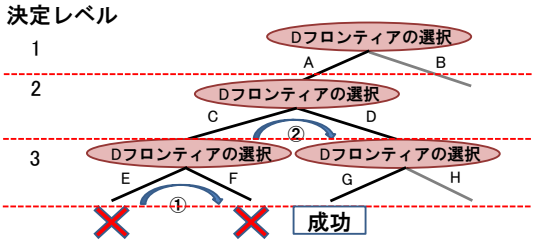

図 2.判定木の例

ある故障の D フロンティア選択において, A→C→E の順で D フロンティア選択して故障伝搬と含意操作を行い, 矛盾が発生した場合を考える. 選択した D フロンティアの故障伝搬経路でテスト生成が成功しなければ 1 つ前の D フロンティア選択肢までバックトラックし, F の経路を選択する. F を選択してもテスト生成が成功しなければ C の D フロンティア選択までバックトラックし, A→D と選択する.

4. テスト生成の解空間

図 3 はテスト生成における判定木の解空間をについて示した図である.

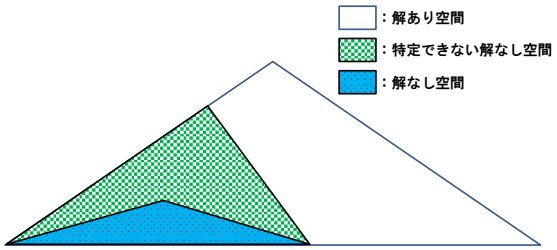

図 3.テスト生成における解空間

図 3 に示すように解空間[6]はテスト生成が成功する「解あり空間」, 選択した時点では矛盾が発生しない「特定できない解なし空間」, 矛盾が発生してテスト生成に失敗する「解なし空間」の 3 つの空間によって構成される.

テスト生成において打ち切り故障を削減するには探索アルゴリズムにおいて解なし空間を避け, バックトラックの回数を減らすことが重要となる. 具体的には「特定できない解なし空間」での無駄なバックトラックの回し, これらの空間を避けるようにバックトラックする必要がある.

5. ルックバック手法

文献[5]で提案されたルックバック手法は図 1 の Step5 においてバックトラック制限に達している場合に実行される.

5.1 ルックバック手法アルゴリズム

ルックバック手法のアルゴリズムを図 4 に示す.

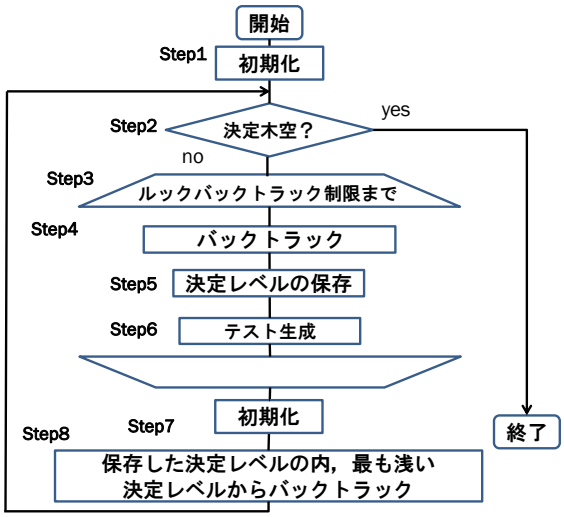

図 4.ルックバック手法アルゴリズム

(Step1)

バックトラック数を 0 に初期化する.

(Step2)

決定木が空ならば打ち切り故障として処理を終了する.

(Step3)

ルックバックトラック制限の値に達する

まで Step4，5，6，の処理を繰り返す．

(Step4)

直前の選択にバックトラックする．

(Step5)

バックトラックの処理を行った決定レベルを保存する．

(Step6)

テスト生成処理を行う．

(Step7)

バックトラック数を 0 に初期化する．

(Step8)

保存した決定レベルの内，最も浅い決定レベルからバックトラックする．

5.2　ルックバック手法の例

図 5,6 を用いて従来の順序式バックトラック手法とルックバック手法の例を示す．

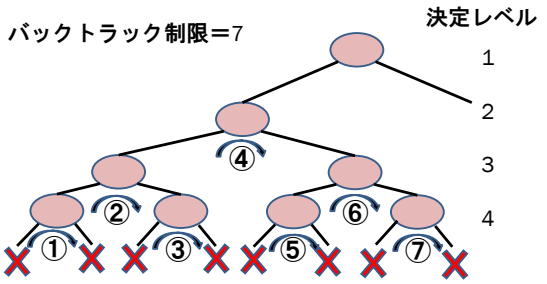

図 5.順序式バックトラック手法

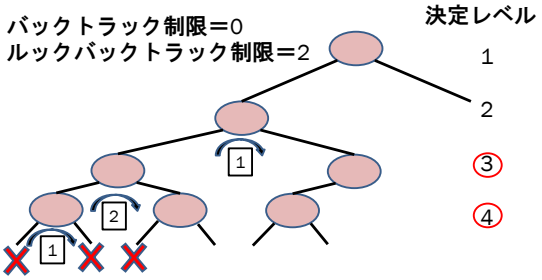

図 6.ルックバック手法

図 5 は従来手法の順序式バックトラックでバックトラック制限=7 とした場合の例である．従来手法ではバックトラック制限=7 に達している場合に矛盾が発生しているのでテスト生成を打ち切る．

一方，図 6 は提案手法を導入し，バックトラック制限=0, ルックバックトラック制限=2 とした場合の例である．まずルックバックトラック制限に達するまで決定レベルを保存しながら従来のバックトラックを行う．またルックバックトラック制限に達している場合に矛盾が発生したら，まずバックトラック数を 0 に初期化する．次に保存した決定レベルの内，最も浅いレベルからバックトラックを行う．例では最初にルックバックトラック制限に達した場合に保存されている決定レベルは 3,4 なので決定レベル 3 から再びバックトラックを開始する．提案手法を用いることで解なし空間での無駄なバックトラックを避ける可能性があり，解空間を広く探索することが可能となる．

6.　実験結果

テスト生成アルゴリズム SPOP のバックトラック部にルックバック手法を組み込んで実装し，故障シミュレーションありで ISCAS'85 ベンチマーク回路に対して実験を行った．

表 1 は従来手法がバックトラック制限=5, 提案手法がバックトラック制限=5, ルックバック制限=1 としてテスト生成を行った結果を示している．実験結果から提案手法は従来手法と比較して打ち切り故障数を削減し，故障検出率を増加させることができた．

表 2 は c1355 と c2670 の回路に対して，バックトラック制限とルックバックトラック制限の値を変えた結果の比較を示している．c1355 ではルックバック手法を導入することで打ち切り故障数を 0 にし，故障検出率を増加させることができた．しかし，c2670 ではバックトラック制限=100 のみの場合とルックバック手法を導入したものを比較すると，打ち切り故障を 0 にできず，打ち切り故障数を増加した．これはルックバック手法

表 1.打ち切り故障数の比較(1)

		バックトラック制限=5					バックトラック制限=5,ルックバックトラック制限=1				
回路名	故障数	検出数	打ち切り故障数	冗長故障数	総バックトラック数	故障検出率	検出数	打ち切り故障数	冗長故障数	総バックトラック数（ルックバック数）	故障検出率
c1355	1574	1419	172	8	1182	88.56	1566	0	8	1571 (431)	99.49
c1908	1879	1856	11	9	69	98.78	1867	0	9	31 (18)	99.36
c2670	2747	2620	5	112	157	95.38	2620	1	116	157 (5)	95.38
c3540	3428	3290	0	137	107	95.97	3290	0	137	107 (0)	95.97
c5315	5350	5278	14	58	514	98..65	5287	5	58	557 (105)	98.82
c6288	7744	7479	231	34	2711	96.58	7611	99	34	23337 (2000)	98.28
c7552	7550	7283	186	81	1889	96.46	7391	72	87	2566 (1490)	97.89

表 2.打ち切り故障数の比較(2)

	c1355						c2670					
	検出数	打ち切り故障数	冗長故障数	総バックトラック数（ルックバック数）	故障検出率	時間(秒)	検出数	打ち切り故障数	冗長故障数	総バックトラック数（ルックバック数）	故障検出率	時間(秒)
バックトラック制限=100 ルックバックトラック制限=0	1419	147	8	18907 (0)	90.15	13.35	2620	0	117	171 (0)	95.38	5.07
バックトラック制限=0 ルックバックトラック制限=1	1566	0	8	2054 (2054)	99.49	20.28	2602	46	89	182 (182)	94.72	4.96
バックトラック制限=0 ルックバックトラック制限=5	1566	0	8	3245 (3245)	99.49	9.78	2600	28	89	165 (165)	95.38	4.99
バックトラック制限=5 ルックバックトラック制限=0	1394	172	8	1182 (0)	88.56	10.31	2620	5	112	157 (0)	95.38	6.5
バックトラック制限=5 ルックバックトラック制限=1	1566	0	8	1571 (2002)	99.49	20.63	2620	1	116	157 (5)	95.38	6.07
バックトラック制限=5 ルックバックトラック制限=5	1566	0	8	4469 (4873)	99.49	15.16	2620	1	116	169 (17)	95.38	6.49

がルックバックトラック制限に達している場合に矛盾が発生したら，その近傍に「解あり空間」があった場合にでもそれを避けるようにバックトラックしてしまったと考えられる．

7. おわりに

本論文では，SPOP のバックトラック部に過去に提案されたルックバック手法を組み込み，実験を行った．その結果，ルックバック手法を導入する場合，回路ごとにバックトラック制限とルックバックトラック制限を設定する必要があることが分かった．今後は回路に依存しない非順序式バックトラック手法を実装する必要がある．

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”，高額図書株式会社(2004)，p135～p175

[2]小澤真由美，” 組合せ回路における再収斂ブロック分割による一意割り当て信号線数評価に関する研究”， 平成 18 年度日本大学生産工学部数理情報工学科卒業研究概要集(2006)

[3] M.Schulz, “SOCRATES：A Highly Efficient Automatic Test Generation System”, IEE TRANSACTIONS ON COMPUTER –AIDED DESIGN,VOL 7 NO 1, JANUARY(1988) , p130

[4]Chen Wang, Sudehakar M.Reddy, Irith Pomeranz, Xijiang Lin, Janusz Rajski, “Conflict Driven Techniques for Improving Deterministic Test Pattern Generation”, International Conference on Computer-Aided Design (2002)

[5] Ilker Hamzaoglu, Janak H, “New Techniques for Deterministic Test Pattern Generation”, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on (1999)

[6]大森悠翔，“静的学習による組合せテスト生成の高速化アルゴリズムに関する研究，平成 17 年度日本大学生産工学部数理情報工学科卒業研究概要集(2005)