

誤り追跡入力の適用順序変更による真理値シミュレーションの効率化

Improving the Efficiency of Truth Variable Simulation
by Changing the Order of Input Patterns to Be Applied坂本 和輝[†] 黒木 修隆[†] 沼 昌宏[†]
Kazuki Sakamoto[†] Nobutaka Kuroki[†] Masahiro Numa[†]

1. はじめに

LSI 製造工程の様々な場面で、仕様変更やタイミング制約違反などにより、設計変更要求 (ECO: Engineering Change Order) が発生している。設計期間やコストの増加を抑制しつつ ECO に対応する手法として、回路に含まれる論理設計誤りを自動修正する論理診断に基づき、配線変更のみで修正を実現する手法が提案されている。

本稿の対象とする論理診断手法 [1] で利用される真理値シミュレーション [2] では、組合せ箇所に含まれる各修正箇所に割り当てられた真理値変数を用いる BDD (Binary Decision Diagram) で表現した共通一致関数に基づいて、各修正箇所の機能変更による回路修正の可能性を判断する。従来手法では、誤り追跡入力を生成された順序通りに適用するため、共通一致関数のノード数増加や、同関数の無効化による組合せ箇所削除のタイミングの遅れによって処理時間が増加する点が課題となっていた。

そこで本稿では、共通一致関数のノード数と、共通一致関数の無効化に基づいて予測した、各誤り追跡入力の組合せ箇所絞り込み効果をもとに、誤り追跡入力の真理値シミュレーションへの適用順序を変更する手法を提案する。

2. 提案手法

共通一致関数のノード数削減に有効な誤り追跡入力や、共通一致関数を無効化した誤り追跡入力を早期に適用するように順序を変更する手法を提案する。共通一致関数をより早期に無効化することで、誤り追跡入力が適用される合計回数を削減するとともに、早期に共通一致関数のノード数を削減することで、一つの誤り追跡入力に対する処理時間を短縮することを狙いとする。

2.1 共通一致関数のノード数に基づく適用順序変更手法

図 1 に、共通一致関数のノード数に基づく誤り追跡入力 α_q に対する適用順序変更手法の処理手順を示す。処理中の組合せ箇所に対して、 q 番目の誤り追跡入力 α_q を用いて共通一致関数 $c_{\alpha_{com}}^q = c_{\alpha_q} \cdot c_{\alpha_{com}}^{q-1}$ を求めた際に、 $c_{\alpha_{com}}^q$ のノード数が $c_{\alpha_{com}}^{q-1}$ よりも小さい場合、 α_q を適用順序リストの先頭に移動する。

ある組合せ箇所に対する処理においてノード数を削減した誤り追跡入力を、次の組合せ箇所に対する処理において、より早期に適用する。これにより、早期に共通一致関数のノード数を削減することで、一つの誤り追跡入力に対する処理時間を短縮することを狙いとする。

2.2 共通一致関数のノード数と無効化に基づく適用順序変更手法

図 2 に、共通一致関数のノード数と無効化に基づく、誤り追跡入力の適用順序変更手法の処理手順を示す。

まず、共通一致関数を無効化した誤り追跡入力を適用順序リストに追加し、その後共通一致関数のノード数を削減した誤り追跡入力を追加する。

共通一致関数の無効化によって、BDD のノード数を 0 とした場合は、処理対象とする組合せ箇所に対する機能変更を行っても機能仕様を満たすことは不可能であるので、その組合せ箇所を削除し、次の組合せ箇所に対する処理を行う。そのため、共通一致関数をより早期に無効化した場合は、修正解とは無関係な組合せ箇所に対する処理回数を削減することが可能である。

よって、共通一致関数を無効化した誤り追跡入力の方が、共通一致関数を無効化せず、ノード数を削減した誤り追跡入力よりも処理時間短縮の効果が高いと予想される。共通一致関数を

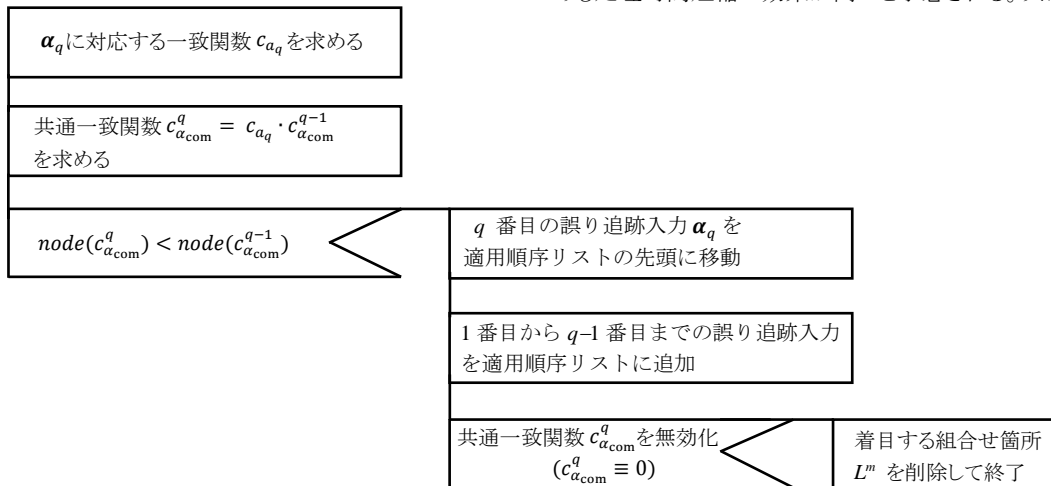


図 1 共通一致関数のノード数に基づく誤り追跡入力の適用順序変更手法の処理手順

[†] 神戸大学, Kobe University

表 1 各手法の処理時間 (b14_opt_C 回路, 多重度 5) [s]

手法	従来手法	提案手法 A	提案手法 B	
回路例	1	0.16	0.10	0.08
	2	118.10	26.17	25.93
	3	0.03	0.04	0.05
	4	0.15	0.06	0.07
	5	6.82	3.32	3.30
	6	85.85	90.03	49.55
	7	39.31	28.7	29.16
	8	2.54	0.55	0.34
	9	227.65	208.95	128.41
	10	0.07	0.05	0.33
幾何平均 (比)	3.09 (1.000)	1.77 (0.574)	1.46 (0.474)	

表 2 各手法の処理時間 (b20_C 回路, 多重度 4) [s]

手法	従来手法	提案手法 A	提案手法 B	
回路例	1	5.09	1.39	1.36
	2	1.54	0.89	0.86
	3	0.13	0.09	0.08
	4	1.26	0.26	0.26
	5	0.54	0.29	0.29
	6	476.52	331.07	257.95
	7	0.48	0.11	0.09
	8	16.25	16.1	16.67
	9	0.11	0.09	0.06
	10	1,944.58	1,307.47	992.65
幾何平均 (比)	3.75 (1.000)	1.89 (0.504)	1.66 (0.444)	

無効化した誤り追跡入力をより早期に適用することで、さらなる処理時間の短縮を図る。

3. 実験と評価

修正箇所シミュレーションに基づく論理診断手法のプログラムをもとに、提案した各手法を計算機 (CPU: Corei7-12700K 3.60 GHz, RAM: 128 GB) 上に実装した。b14_opt_C 回路に対して 5 箇所, b20_C 回路に対して 4 箇所の素子機能誤りを無作為に挿入した回路を各 10 例, 合計 20 例について、真理値シミュレーションの処理時間を評価項目として実験評価を行った。

誤り追跡入力の適用順序変更手法に関して以下の 3 種類の手法を実装し、適用する誤り追跡入力の相違による処理時間への影響を避けるため、各回路例については同一の誤り追跡入力を用いて実験を行った。

- i) 従来手法 : 適用順序を変更しない
- ii) 提案手法 A : 共通一致関数のノード数に基づく誤り追跡入力の適用順序変更手法
- iii) 提案手法 B : 共通一致関数のノード数と無効化に基づく誤り追跡入力の適用順序変更手法

表 1, 表 2 に、それぞれ b14_opt_C 回路, b20_C 回路に対する処理時間の結果を示す。処理時間の幾何平均値に関して、提案手法 A が b14_opt_C 回路に対して 42.6%, b20_C 回路に対して 49.6% 短縮する効果が得られた。提案手法 B では b14_opt_C 回路に対して 52.6%, b20_C 回路に対して 55.6% 短縮する効果が得られた。

提案手法 B で処理時間を最も短縮することができたことから、共通一致関数を無効化した誤り追跡入力と共通一致関数のノード数を削減した誤り追跡入力の双方を早期に適用することが、処理時間の短縮に有効であることが確認できた。

4. まとめ

本稿では、修正すべき箇所の組合せを絞り込む真理値シミュレーションの効率化による論理診断処理時間の短縮を目的として、共通一致関数に基づく誤り追跡入力の適用順序変更手法を提案した。

提案手法を計算機上に実装して評価実験を行った結果、4, 5 箇所の素子機能誤りを含む b14_opt_C, b20_C 回路に対して、共通一致関数のノード数と無効化の双方を考慮した誤り追跡入力の適用順序変更手法が、真理値シミュレーションの処理時間短縮に有効であることを確認した。

今後の課題としては、有効な誤り追跡入力を生成することで、処理時間のさらなる短縮を図ることが挙げられる。

参考文献

- [1] H. Tsuyama, A. Masamori, N. Kuroki, and M. Numa, "An error diagnosis technique based on location variable simulation employing implicit representation of error location sets," The 24th Workshop on Synthesis And System Integration of Mixed Information Technologies (SASIMI 2022), pp. 127-132, Oct. 2022.
- [2] 沼 昌宏, 井上 宏, 皆見利行, 黒木修隆, 山本啓輔, "真理値シミュレーションに基づく LUT 論理診断手法", 情報処理学会論文誌, vol. 43, no. 5, pp. 1252-1259, 2002.

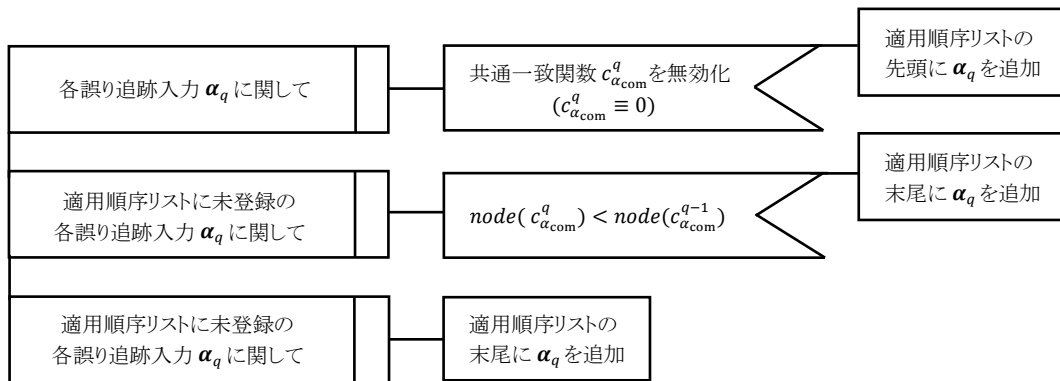


図 2 共通一致関数のノード数と無効化に基づく誤り追跡入力の適用順序変更手法の処理手順