

5N-2 論理検証システム EVERY7 (II)

山際 肇 田中 英俊 飛永 聡 向山 輝 若林 一敏 高橋 美希子
 NECソフトウェア北陸(株) NEC(株)

1. はじめに

レジスタトランスファ(以下RT)レベルの機能記述言語を用いたゲートレス設計においては、回路をRTレベルで記述したものと、ゲートレベルのネットリスト間の、論理の等価性を完全に検証することが求められている。しかし、従来当社で用いていたパタンシミュレーションによる論理検証手法では、より完全な検証を行うためには膨大なテストパタンやシミュレーション時間が必要であった。このパタンシミュレーションでの限界を打破するために、フォーマルベリフィケーションを取り入れた論理検証システム"EVERY7"を開発した。本システムを利用することにより、論理検証時間の大幅な短縮とより完全な検証が可能となった。

2. EVERY7概要

EVERY7は、回路の機能仕様をRTレベルの機能記述言語FDL(Function Description Language)[1]で記述したものと、ゲートレベルのネットリストで表現したものを入力し、両者の論理的等価性の検証を自動で行なうものである。

- 図1にEVERY7の概要を示す。EVERY7は、
- ① 組合せ回路化部
 - ② フォーマルベリフィケーション部
 - ③ パタンシミュレーション部

の3つの機能部分より構成される。

まず、①にて検証回路のFDL記述とネットリストは、レジスタ情報を基に組合せ回路化されたものへと変換される。

次に、②にて組合せ回路化されたFDL記述とネットリストから、対応する組合せ回路同士の論理照合を行い、論理の等価性を検証する。照合できなかった組合せ回路が存在した場合、または一致しなかった組合せ回路を発見した場合は、不可・不一致情報ならびにエラーを発見できるテストパタンを生成する。

最後に、③にて組合せ回路化されたFDL記述とネットリストの論理シミュレーションを、生成されたテストパタンにて行い、ユーザーインターフェイスとなるシミュレーション結果リストを出力する。

3. フォーマルベリフィケーション部

フォーマルベリフィケーション部は、当社で開発したフォーマルベリフィケーションシステムCONDOR[3]を使用している。

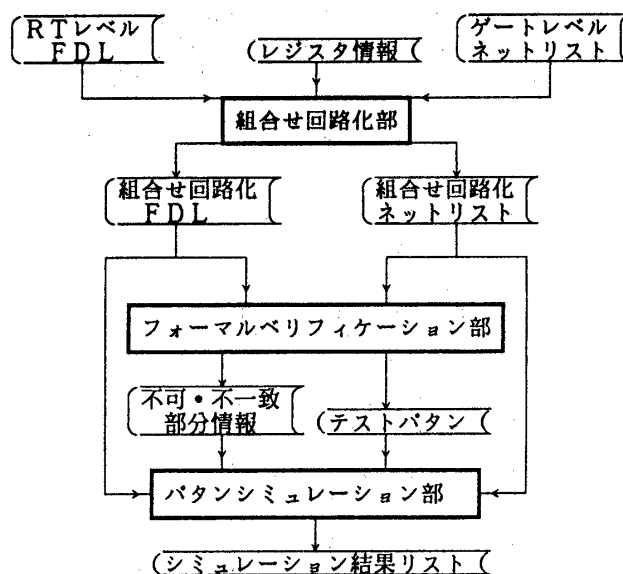


図1. EVERY7の概要

(1) CONDOR

FDL記述とネットリストの論理を二分決定グラフ(BDD)[4][5]によって表現し、両者の論理的等価性を証明することにより論理検証を行うシステムである。

BDDは、グラフによる関数の論理表現法であり、

- 比較的コンパクトな論理関数の表現
- グラフのパス上の入力変数の順序による、論理表現の一意的決定

という特長を持つ。これらの特長のために、論理関数をBDDで表すことにより、論理的等価性の判定を、高速かつ完全に行うことが可能となる。

しかし、大規模回路をBDDで表現した場合、記憶領域と処理時間の両面で限界に達するため、BDDの節点数をなるべく少なくすることが重要である。

CONDORは、ヒューリスティックを利用した入力変数の順序最適化や、回路の分割照合によってBDDの節点数の削減を図り、現実の大規模論理回路の検証を可能にしている。

図2にCONDORの概要を示す。

FDL記述ならびにネットリスト中の対応する組合せ回路単位で論理照合は行われる。双方の組合せ回路中の論理を、まずそれぞれ論理式(ブール代数式)化を行い、次にそれぞれBDDにより表現し、その差分を判定することにより論理的等価性を証明する。また、組合せ回路の規模が大きい場合には、さらに部分回路に分割し、部分回路毎にBDDで表現することにより論理照合を行う。部分回路への分割は、機能仕様から抽出された中間信号を利用して行う。

A Logic Verification System "EVERY7" (II)
 Hajime YAMAGIWA, Hidetoshi TANAKA, Satoshi TOBINAGA
 NEC Software Hokuriku, Ltd.
 Akira MUKAIYAMA, Kazutoshi WAKABAYASHI, Mikiko TAKAHASHI
 NEC Corporation.

(2) テストパタンの生成

論理的に等価でない2つのBDDの排他的論理和により生成されたBDDにおいては、その最下端にある定数値1を持つ定数節点に行き付くような各節点の割り当て値を求めれば、これは論理的な差異を検出するテストパタンになる。[6]

CONDORでは、この原理を利用して、論理的に等価でないと判断した組合せ回路に対しては、パタンシミュレーションを行なうと、FDL記述ならびにネットリスト双方のシミュレーション結果が異なるようなテストパタンを生成する。このテストパタンにより、次節で説明するパタンシミュレーションが行われる。

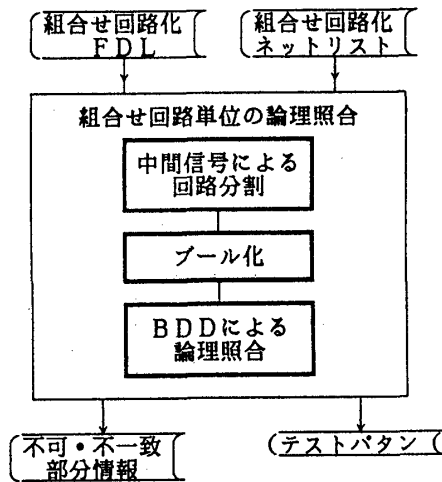


図2. CONDORの概要

4. パタンシミュレーション部

現状のフォーマルベリフィケーションには、幾つかの問題点が残っている。

- 乗算器のように、BDDに変換するとグラフの規模が急激に大きくなり論理照合が行えないものがある。
- 論理エラーの存在は確認できても、エラー箇所を特定するための情報を充分設計者に与えることができない。これらを補完するために、EVERY7ではフォーマルベリフィケーションにおいて論理照合が出来なかった組合せ回路、またはエラーを発見した組合せ回路については、パタンシミュレーションを継続して行っている。

図3にパタンシミュレーションの概要を示す。フォーマルベリフィケーションにおいて論理照合ができなかった組合せ回路に対しては、FDL記述より自動生成したテストパタンにより[7]、また論理エラーを発見した組合せ回路に対しは、フォーマルベリフィケーション部にて生成したテストパタンにより、FDL記述ならびにネットリスト双方の、これらの組合せ回路に対する論理シミュレーションを実行する。

論理エラーを解析するための情報として、ここではFDL記述中の全信号の状態値、ならびにネットリスト中の全ネットの状態値を表示したシミュレーション結果等を出力する。これらを利用して、設計者が論理エラー発見時の解析を迅速に行うことができるようになっている。

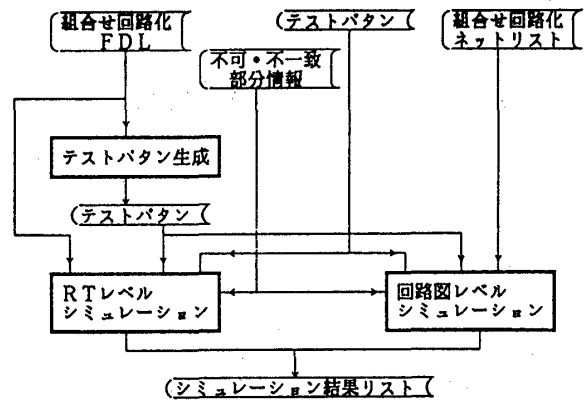


図3. パタンシミュレーション概要

5. 評価結果

本システムをEWS4800ワークステーション(30Mips)上に実現し、社内で設計された回路を対象に評価を行った。また、パタンシミュレーションを用いた既存の検証システムとの比較も行った。その結果を表1に示す。

パタンシミュレーションを用いた既存の検証システムに比べ本システムは検証時間を数十~百分の1程度に短縮できることが実証された。

また、従来のパタンシミュレーションでは検出できなかった論理エラーも検出できた。

6. むすび

現在本システムは、フォーマルベリフィケーションとパタンシミュレーションの2本立で検証を行っているが、今後はフォーマルベリフィケーションの処理能力の向上ならびにエラー解析情報提示能力の向上を図り、フォーマルベリフィケーションのみで全ての対象回路を処理できるようにしていく予定である。

表1. 評価結果

回路	A	B	C
回路規模(ゲート数)	15000	6027	5856
EVERY7による検証時間	1740s	275s	299s
パタンシミュレーションによる検証時間	125h	15831s	17304s
検証時間比(EVERY7:パタンシミュレーション)	250:1	58:1	58:1

参考文献

- [1] Kato, S. and Sasaki, T., "FDL: A Structural Behavior Description Language", CHDL83(1983), 137-152.
- [2] 下野, 田中, "EWS上の論理検証システム: EVERY4", 第20回 FTC研究会資料(1989).
- [3] 向山, 他 "論理検証システム CONDOR", 情処研報 Vol.92, No. 56, pp17-22, 1992.
- [4] S.B.Akers. "Binary Decision Diagrams", IEEE Transaction on Computers, Vol.C-27, No.6, pp509-516, Jun. 1978.
- [5] R. E. Bryant. "Graph-Based Algorithms for Boolean Function Manipulation", IEEE Transaction on Computers, Vol. C-35, No.8, pp677-691, Aug. 1986.
- [6] 石浦, "二分決定グラフとは", 第5回回路とシステム 軽井沢ワークショップ, pp155-159, Apr. 1992.
- [7] M.Kawai et al., "A High Level Test Pattern Generation Algorithm", Proc.1983 ITC, pp346-352, 1983.