

動作合成された束データ方式による非同期式回路とその動作仕様に対する等価性検証手法

田川 貴聡[†] 松本 剛史[‡] 藤田 昌宏^{†,§}

[†] 東京大学大学院工学系研究科電気系工学専攻 [‡] 東京大学大規模集積システム設計教育研究センター

[§] 科学技術振興機構 戦略的創造研究推進事業 CREST

1 はじめに

非同期式回路には、低消費電力・低電磁放射などの利点があるが、ハザードのない回路設計が求められるうえ、利用できる CAD ツールも限られており、その設計は困難である。そのため、非同期式回路設計のための CAD ツールが求められており、特に動作合成手法は必要とされている。

動作合成とは、C 言語や SystemC 言語などによって記述された動作記述から RTL モデルを合成する手法である。動作合成では、スケジューリング、リソースアロケーション、バインディング、制御回路合成を行い、最適な RTL モデルを合成する。システムレベル設計の要求から、同期式回路では多くの動作合成手法が研究されてきた。

動作合成を用いた設計フローでは、早期にバグを漏れなく発見し、取り除くことが重要である。そのための有効な手法の一つが、等価性検証手法である。動作合成における等価性検証では、動作合成後の RTL モデルと、動作合成前の動作記述が等価であることを証明する。これによって、元の動作記述が正しければ、合成後の RTL モデルの動作の正しさが保証され、設計誤りの混入を防ぐことができる。

このような等価性検証手法も、同期式回路では多く研究がなされているが、非同期式回路では、必要性が高いにも関わらず、あまり研究されていない。本稿では、束データ方式による非同期式回路の動作合成前後における等価性検証手法を提案する。

2 関連研究: 束データ方式による非同期式回路の動作合成手法 [1] [2]

動作記述から、束データ方式の非同期式回路を合成する手法を紹介する。この手法では、まず入力動作記述から DFG(Data Flow Graph) を生成し、図 1 のようにスケジューリングを行い、最適な演算開始のタイミングを決定する。それから、アロケーション、制御回路合成などを行い、図 2 のような束データ方式の回路を生成する。

Equivalence Checking between a Synthesized Asynchronous Circuit with Bundled-data Implementation and its Behavior Specification

Takaaki TAGAWA[†] Takeshi MATSUMOTO[‡]
Masahiro FUJITA^{†,§}

[†] Department of Electrical Engineering and Information Systems, The University of Tokyo

[‡] VLSI Design and Education Center, The University of Tokyo

[§] CREST, JST

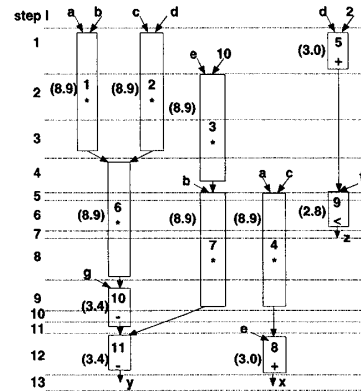


図 1: スケジューリング後の DFG([1] より引用)

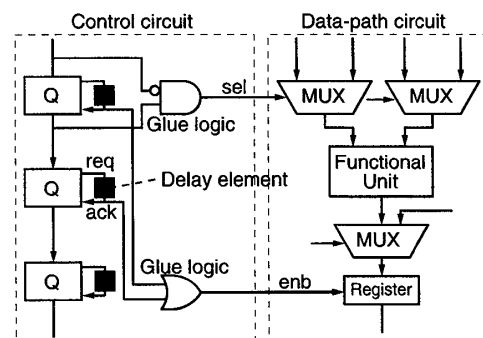


図 2: 束データ方式の回路モデル

束データ方式とは、非同期式回路のデータエンコーディングのひとつである。要求信号 *req* と応答信号 *ack* からなるハンドシェイク信号によって動作を制御する。演算の終了を保証するために、要求信号線上に遅延素子を挿入する。実行時間は、遅延素子遅延の総和に等しくなる。スケジューリングにおける一つのコントロールステップは、一つの Q 素子にマッピングされる。この Q 素子によって、制御は順次移っていく。制御回路で生成された信号によって、データパス回路におけるマルチプレクサの選択信号やレジスタの書き込み信号を制御する。

3 提案する等価性検証手法

提案する等価性検証手法では、データフローとタイミング制約の検証を行う。束データ方式による非同期

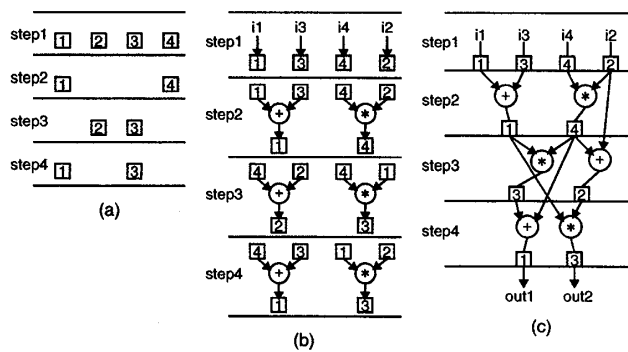


図 3: 束データ方式の回路モデル

式回路の RTL モデルと、対応する動作記述を入力とし、検証結果 (PASS/FAIL) を出力する。

合成後の RTL モデルから DFG を生成し、動作記述から生成した DFG と比較することでデータフローの検証を行う。また、RTL モデルから DFG を生成する際、コントロールステップを考慮したものを生成する。つまり、スケジューリング後の DFG を再現することになる。この DFG と、遅延素子の遅延時間を参照することで、タイミング制約の検証を行う。

3.1 RTL モデルからの DFG の生成

本手法では、各コントロールステップにおけるマルチプレクサの選択信号やレジスタの書き込み信号の対応は、シミュレーションなどによってわかっているものと仮定する。この仮定のもとで、次のような手順で DFG を生成する。

1. 各コントロールステップにおいて書き込みのあるレジスタを抽出する。これらのレジスタを出力レジスタと呼ぶことにする。(図 3 (a))
2. 入力レジスタまたは入力ポートから出力レジスタまでのパスを抽出する。(図 3 (b))
3. 入力レジスタを、前方の最も近い出力レジスタと統合する。(図 3 (c))

3.2 データフローの検証

第 3.1 節の方法で生成した DFG と動作記述から生成した DFG を比較して、等価性を調べる。DFG の等価性の判定には、記号シミュレーションを用いて充足可能性判定問題に帰着させる従来手法 [3] を適用することができる。

3.3 タイミング制約の検証

第 3.1 節の手法で生成した DFG の各コントロールステップについて、最も遅延の大きなパスを探索する。各演算に対応する演算器の最大遅延時間が与えられているものとする。そのパスの遅延が、対応する遅延素子の遅延時間より小さいかどうか調べることで、タイミング制約を満たすことを保証できる。

3.4 ケーススタディ

演算子の数が 34 個の動作記述を例題として、提案手法のうちデータフローの検証を行った。提案した手順に従って、手動で実験を行った結果、データフローが正しいことが確認できた。タイミング制約の検証については、例題のリソースライブラリや遅延素子の遅延時間の情報を入手していないため、行っていない。今後、これらの情報を入手して、実験を行う予定である。

4 まとめと今後の課題

本稿では、束データ方式による非同期回路の動作合成前後における等価性検証手法を提案した。提案手法では、データフローとタイミング制約について検証を行う。

今後の課題として、次の 2 点を挙げる。

- 提案手法をプログラムで実装し、例題に適用して、その有効性や問題点について検討する。
- 現在は演算器の最大遅延のみを考えて検証を行っているので、ショートパス問題については検証できていない。今後は、最小遅延時間も考慮した検証手法についても検討する。

参考文献

- [1] Naohiro Hamada, Yuuki Shiga, Hiroshi Saito, Tomohiro Yoneda, Chris Myers, and Takashi Nanya, "A Behavioral Synthesis Method for Asynchronous Circuits with Bundled-data Implementation (Tool Paper)", *Proc. International Conference on Application of Concurrency to System Design*, pages 50-55, 2008.
- [2] Naohiro Hamada, Yuuki Shiga, Takao Konishi, Hiroshi Saito, Tomohiro Yoneda, Cris Myers, and Takashi Nanya, "A Behavioral Synthesis System for Asynchronous Circuits with Bundled-data Implementation", *IPSJ Transactions on System LSI Design Methodology*, No.2, pages 67-79, February 2009.
- [3] E. Clarke, D. Kroening, and K. Yorav, "Behavioral consistency of C and verilog programs using bounded model checking", *In Proc. of the Design Automation Conference*, pages 368-371, June 2003.