科学研究費助成事業　　研究成果報告書

平成 ２６ 年　 ６ 月 １１ 日現在

機関番号：１２６０１
研究種目：若手研究(B)
研究期間：2011～2013
課題番号：２３７０００５１

研究課題名（和文）システムレベル設計に対する入出力タイミングを考慮した等価性検証手法に関する研究

研究課題名（英文）Equivalence Checking for System-Level Designs Having Different Input-Output Timings

研究代表者
松本　剛史（MATSUMOTO, Takeshi）

東京大学・大規模集積システム設計教育研究センター・助教

研究者番号：４０５３６１４０

交付決定額（研究期間全体）：（直接経費）　　2,800,000円、（間接経費）　　840,000円

研究成果の概要（和文）：組み込み機器やVLSIの設計記述の論理的な正しさを検証する手法について研究を行った。その中でも、研究の対象は、与えられた2つの設計記述が論理的に等価かどうかを調べる等価性検証である。また、検証対象は、近年、広く用いられるようになったシステムレベル設計と呼ばれる抽象度の高い設計記述である。本研究では、システムレベル設計記述に対して、その入出力タイミングを考慮した等価性検証手法、および、内部等価点の効率的な探索手法について研究を行い、システムレベル設計記述の等価性検証をより高性能化することを目指した。

研究成果の概要（英文）：In this work, design verification methods for embedded systems or VLSIs are studied. The purpose of design verification is to check whether a given design is correct or not and provide failing patterns if incorrect. We focus on equivalence checking of given two designs. Our target of verification is system-level design, which is a highly abstracted design level and has become widely applied recently. We proposed equivalence checking methods that can deal with different input/output timings between given two designs. In addition, we have developed a method to detect potentially equivalent internal variables in designs. The purposed of this work is to improve the ability of equivalence checking for system-level designs by those proposed methods.

研究分野：総合領域

科研費の分科・細目：情報学

キーワード：等価性検証　システムレベル設計　形式的検証

１．研究開始当初の背景

多くの組込み機器は、多数のハードウェアとソフトウェアを含むシステムであり、その設計の正しさの検証は非常に難しい問題になっている。様々な検証手法が適用されているが、その中で、等価性検証は強力なバグ検出手法の 1 つである。つまり、既に十分に検証された設計（ゴールデンモデル）と検証対象の設計の等価性を検証し、反例があれば検証対象の設計には設計誤りがあることが分かる。また、等価性を検証しながら設計を進める場合、どの部分で動作仕様の変更がなされたか、を明確にできるため、VLSI 製品を搭載した自動車・人工衛星などにおいて不具合があった際に、その VLSI 製品がどのような動作を行うことが設計段階で保証されているのか、を顧客に対して説明することができる。実用的には、論理合成前後の組合せ回路の等価性検証は確立された技術であり、動作合成前後の動作設計と RTL 設計の等価性検証においても商用ツールが存在する。一方で、システムレベル設計記述に対する等価性検証技術はほとんど存在しない。システムレベルとは、C 言語等で与えられたシステム全体の動作仕様を出発点として、並列性導入・並列動作間の同期設計・ソフト/ハード分割・IP 割当などを行う設計段階であり、多くの設計記述変更・詳細化が行われるため、等価性検証ツールの必要性は実際に非常に高いと考えられる。

２．研究の目的

（１）時間経過を含む 2 つの設計における等価性の定義と表現方法の考案

時間経過記述（例えば、SystemC 言語における wait 文や SpecC 言語における waitfor 文）を含むシステム設計記述において、計算に必要な入力値が与えられるタイミングと計算結果を表す出力値が現れるタイミングを数学的に指定する方法を考案する。単純な一定スループット・レイテンシによる指定法をベースにして、割込みなどのイベント駆動型動作や通信プロトコルによる動作に対しても、検証において考慮すべき入出力のタイミングを考慮した等価性指定方法を検討し、実際のシステムレベル設計記述における等価性が表現できるかを明らかにする。

（２）2 つの設計間で成り立つ可能性のあるタイミング条件の自動推定手法の考案

前項の等価性指定法を用いて、2 つの設計間で成立する可能性のある等価性の自動推定手法を考案する。手法は、いくつかの異なる前提条件に対して検討する。その前提条件としては、(i)入力信号の対応関係とタイミングが与えられた場合に、出力信号の対応関係とタイミングを推定する、(ii)入出力信号の対応関係が与えられたときに、その正しい入力・出力タイミングを推定する、を対象とする。手順としては、まず、設計記述のシミュレーションを行い、等価になる時刻の情報を持った内部等価点候補を抽出し、それらの間の依存関係を解析することにより、設計間で成立つ可能性のある時刻を考慮した等価性を導出する。具体的には、シミュレーションによって「時刻 t1 における変数 A と時刻 t2 における変数 B が等価」のような形で内部等価点候補を抽出し、設計記述を解析によって「候補 1 が成り立つならば、候補 2 が成り立つ」のような依存関係を構築する。

（３）システムレベル設計記述に対する等価性検証における検証性能の向上

（１）および（２）によって、システムレベル設計記述に対する等価性検証を行うことができるようになるが、全体の性能向上・速度向上のためには、検証アルゴリズムそのものの改良も不可欠である。特に、本研究で利用した記号シミュレーションに基づく等価性検証手法では、設計記述中の実行パス数に対して比例する計算量が必要となるため、設計記述中の条件分岐数に対して、検証時間が指数的に増加してしまう。これを改善することは必須である。本研究では、主に、記号シミュレーションにおける条件分岐の扱いについて工夫を加えることにより、検証性能の改善を行う。

３．研究の方法

（１）形式的等価性検証を行う環境の構築

研究を進めるために必要な検証環境の構築を行う。基本的な等価性検証の手法は、研究代表者がこれまで研究開発を進めてきた記号シミュレーション手法に基づく等価性検証ツールを利用する。この手法は、既にツールとして基本的な実装が済んでおり、SpecC 言語記述に対して検証を行うことができる。本研究では、この基本手法を拡張して、タイミングを考慮した等価性検証機能とボトムアップに等価性を検証する機能を追加する。前者は、等価性を指定するインタフェースの変更と既に証明された等価性を管理する内部データベースの改良によって実現可能である。後者については、目的の（２）で述べたような内部等価点を Uninterpreted 関数として表現し、これを等価性検証において利用することにより実現する。Uninterpreted 関数とは、関数名と全ての引数が等価であれば、その関数の戻り値を等価である、とする考え方であり、関数の内部論理を参照することなく等価性の判定を行うことができる。

（２）入出力タイミングを考慮した等価性の数学的定義の検討

2 つの設計間の等価性を数学的に定義し、その表現法を検討する。このとき、入出力タイミングも含めた定義にすることが本研究の特徴の一つである。この定義によって、

スループット・レイテンシ一定動作、イベント駆動型動作、ある通信プロトコルに従った動作等を行う幅広い設計に対して等価性を指定できるようにする。

（3）等価性検証アルゴリズムの改良
記号シミュレーションに基づく等価性検証手法では、分岐数に対して、検証時間が指数的に増加するという問題があった。そこで、各分岐が終了するポイントにおいて、そのポイントで再収斂する実行パスに関する記号シミュレーション結果をマージすることによって、記号シミュレーションに要する計算量を減らす手法を導入する。実装は、（1）の等価性検証ツール上で行い、企業から提供された例題を用いて評価を行う。

（4）内部等価点を効率的に求める手法の研究開発
内部等価点の利用は等価性検証の効率化において非常に重要であることが知られている。本研究では、GPGPU を用いて、効率的に内部等価点を求める手法を提案・評価する。

（5）等価性検証をシミュレーションによって検証する手法の検討
近年、設計中の一部が任意に論理変更した場合に、それを検出するためのテストパタン生成手法が研究されている。これは、別の見方をすれば、その「一部」の論理を同定するためのテストパタンが生成できるということである。そこで、この考え方を等価性検証に応用し、2 つの設計が等価でない場合、その 2 つの設計はどの程度「不等価」であるのか、を示す指標として、不等価となるテストパタン集合の利用を検討する。

4．研究成果

（1）等価性検証ツールの改善
従来から研究開発を行っている等価性検証ツールの機能改善・性能改善を行った。ツールは、これまでに約 10 の大学・企業（海外を含む）に対して、評価のために提供されており、評価結果を受けたバグ修正や改善がなされた。また、第 3 節（3）で示した性能改善を取り入れた結果、分岐数 20 の例題において、10 倍の検証時間高速化を実現することができた。

（2）タイミングを考慮した等価性検証ツールインタフェース
フィルタや信号処理、暗号化・復号化では、データが一定の時間間隔で到着し、ハードウェアで処理される場合が多い。これは、言い換えれば、一定のスループットでハードウェアが動作するということである。等価性検証では、等価であることを仮定する入力変数の対と等価であるかどうかを検証する出力変数の対をあらかじめ指定し、指定した仮定が成り立つ場合に、指定した等価性がどのような入力パタンに対しても成立するかどうかを調べている。本成果によって、この指定において、各入力変数・出力変数のスループットとレイテンシを指定できるようになった。また、入出力変数が有効であるための条件（例えば、リセット信号がオフであること、等）も併せて記述できるようなインタフェースとなっている。ベンチマーク例題や企業から提供された例題を通して、処理に使われる変数・処理結果を表す変数が特定のタイミングで入力・出力される場合においても、等価性検証が可能であることを示した。

（3）内部等価点候補の効率的な抽出手法
GPGPU を用いて、設計記述中の内部変数のうち、等価になる可能性がある候補を抽出する手法を実装した。この手法では、与えられたシミュレーション結果に対して、指定された時間幅の中で成り立つ変数間の等価性を網羅的に調べることができる。提案手法を GPGPU 上に実装して評価した結果、通常の CPU 上での実行と比べて、30 倍高速に内部等価点候補を抽出できることが確認できた。

（4）シミュレーションによる等価性検証の実現
第 3 節（4）で述べた考え方によって、シミュレーションによって等価性を検証する手法を検討し、そこで必要となるテストパタン生成に成功した。この手法では、まず、既に十分に検証がなされており正しいことが保証されている設計（または、正しいと信じるに足る設計）に対して、ある範囲の変更が行われることを想定する。その変更によって、変更前後の設計が不等価になる場合、その全てを検出するテストパタンを生成する。正しい（と信じられる）設計と検証対象の設計の等価性検証を行う際には、生成されたテストパタンを用いてシミュレーションを行うことにより、不等価（元の設計からの論理変更）を検出することができる。本研究では、初期実験として、10 行程度の C プログラムに対してテストパタン生成を行い、パタン数が数百程度であること、パタン生成時間が 10 分程度であることを確認した。今後は、より大規模な設計例題での評価を行っていく予定である。

（5）検証結果が不等価である場合の原因箇所の特定
等価であることが期待される 2 つの設計記述の等価性検証結果が等価でなかった場合には、設計中のどの部分が原因で不等価であるのか（言い換えれば、どこを修正すれば等価になるのか）を求める必要がある。本研究では、充足可能性判定ソルバーが出力する UNSAT コアと呼ばれる項集合や設計記述の依存関係を利用して、修正候補箇所を求める手法を提案した。

５．主な発表論文等

〔雑誌論文〕（計１件）

① S. Jo, T. Matsumoto, M. Fujita, SAT-based Automatic Rectification and Debugging of Combinational Circuits with LUT Insertions, IPSJ Transactions on System LSI Design Methodology, 査読有, Vol. 7, 2014, pp. 46-55, DOI: 10.2197/ipsjtsldm.7.46

〔学会発表〕（計６件）

①松本剛史，城怜史，藤田昌宏，プログラム可能データパスと SMT ソルバーを利用した高位設計デバッグ手法，組込み技術とネットワークに関するワークショップ ETNET2014, 2014 年 3 月 15 日，石垣，沖縄
②M. Fujita, T. Matsumoto, S. Jo, FOF: Functionally Observable Fault and its ATPG techniques, IFIP/IEEE 21st International Conference on Very Large Scale Integration and System-on-Chip, Oct. 6th-9th, 2013, Istanbul, Turkey
③K. Oshima, T. Matsumoto, M. Fujita, A debugging method for gate level circuit designs by introducing programmability, IFIP/IEEE 21st International Confer- ence on Very Large Scale Integration and System-on-Chip, Oct. 6th-9th, 2013, Istanbul, Turkey
④ S. Ono, T. Matsumoto, M. Fujita, Automatic Assertion Extraction in Gate-Level Simulation Using GPGPUs, IEEE 30th International Conference on Computer Design, Sep. 30th-Oct. 3rd, 2012, Montreal, Canada
⑤T. Matsumoto, S. Ono, M. Fujita, An Efficient Method to Localize Correct Bugs in High-Level Designs Using Counterexamples and Potential Dependence, IEEE/IFIP 20th International Symposium on Very Large Scale Integration, Oct. 7th-10th, 2012, Santa Cruz, USA
⑥李在城，松本剛史，藤田昌宏，論理関数の充足不可能性に注目した論理回路デバッグ手法の検討，組込み技術とネットワークに関するワークショップ ETNET2012, 2012 年 3 月 2 日，松島，宮城

〔図書〕（計０件）

〔産業財産権〕
○出願状況（計０件）

○取得状況（計０件）

〔その他〕
なし

６．研究組織

(1)研究代表者
松本　剛史（MATSUMOTO Takeshi）
東京大学・大規模集積システム設計教育研究センター・助教
研究者番号：４０５３６１４０

(2)研究分担者
なし

(3)連携研究者
なし