

機関番号：17102  
 研究種目：基盤研究（B）  
 研究期間：2008～2010  
 課題番号：20300020  
 研究課題名（和文） TLM 検証を高速化するモデルの抽象化技術とテストパターン生成技術の研究  
 研究課題名（英文） Model abstraction for accelerating TLM verification and test pattern generation  
 研究代表者  
 松永 裕介（Yusuke Matsunaga）  
 九州大学・システム情報科学研究院・准教授  
 研究者番号：00336059

## 研究成果の概要（和文）：

ハードウェアだけでなく、マイクロプロセッサ上で動作するソフトウェアも含んだデジタルシステムの検証において用いられるトランザクションレベルモデリング（TLM）技術の効率化を目的として、その基盤要素技術の開発を行なった。具体的には、レジスタ転送レベルのハードウェア記述を TLM へ変換する TLM トランスレータの開発、および、SMT ソルバアルゴリズムの効率化、シミュレーション高速化を目的とした FPGA 用合成技術の開発を行なった。

## 研究成果の概要（英文）：

This research project is focusing to develop fundamental acceleration techniques for verification methodology on Transaction-Level Model. As outputs of the project, TLM translator, SMT solver, and logic synthesizer for FPGAs aiming to accelerate TLM simulation are developed.

## 交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2008年度	6400000	1920000	8320000
2009年度	3700000	1110000	4810000
2010年度	3800000	1140000	4940000
年度			
年度			
総計	13900000	4170000	18070000

研究分野：計算機科学

科研費の分科・細目：情報学・計算機システム・ネットワーク

キーワード：システムLSI，システムレベル検証，テストパターン生成

## 1. 研究開始当初の背景

現在のLSIは「システムLSI」と呼ばれ、1チップ内に複数のマイクロプロセッサ、メモリ、信号処理専用プロセッサ、画像処理回路、周辺回路などの多くの部品を含むようになり、規模や複雑さは増大している。また、ハードウェアだけでなく、マイクロプロセ

ッサ上で動作するソフトウェアも含んだシステムとして検証する必要があり、システムレベル検証がもっとも困難な作業のひとつとなっている。

そこで、近年注目を集めている手法がTLMとよばれるモデル化手法である。これは、あるまとまったデータをメモリに書き込む、と

か IO ポートに出力する、というデータのやりとりの単位で動作を抽象化したもので、RTL に比べて 100~1000 倍程度、高速にシミュレーションを行うことが可能となっている。システム LSI の設計を行う際に、まず回路の各部分をトランザクションレベルでモデル化を行い、その後はハードウェア設計者もソフトウェア設計者もこの TLM を見本として各担当分の実装設計を行えば良い。前述のように、TLM のシミュレーションは高速であり、また、数百 MHz のクロックで動作するシステム LSI の動作を数 GHz のプロセッサ(Pentium など)で模倣させることによってほぼ実機の数 10~100 分の 1 程度のスピードでシミュレーションを行うことができる。

TLM を用いたシステムレベル検証においてもっとも自動化されていない部分は TLM で記述された「仕様」を RTL の「実装」へ詳細化していくプロセスである。

一般にそのような処理を自動で行うことを「動作合成」もしくは「高位合成」と呼び、大学などの研究機関では古くから研究が行われているが、広く実用に用いられる状況にはなっていない。そのため、設計者が人手で RTL 記述を作成しているのが現状である。

この場合、例えば A という部分モジュールに対する TLM レベルの仕様と、RTL の実装が存在することになる。他のハードウェアモジュールの設計者やソフトウェアの開発者は A の RTL 実装が出来上がる前に TLM レベルの仕様に基づいて各自の開発を行うので、もしもこれらの仕様と実装の間に齟齬が生じていたらシステム全体の正しい設計が行えなくなる。

そこで、自動合成に代わる次善の策は、TLM レベルの仕様と RTL の実装が機能的に等価であるかどうかを検証することであるが、そのような検証を自動的に(特に数学的・形式的に)行う技術は確立しておらず、その問題の複雑さから考えて、広く汎用で使えるレベルの効率のよいアルゴリズムは存在しないと思われる。結局、現状の TLM を用いた検証フローでは、以下のような手順で「仕様」から「実装」への詳細化を行っている。

- (1) A という部分モジュールに対する TLM レベルの仕様(SPEC<sup>TLM<sub>A</sub></sup>)を作成し、それを TLM レベルのテストベンチを用いて検証する。他のモジュールの開発者やソフトウェア開発者はこの SPEC<sup>TLM<sub>A</sub></sup> をゴールデンモデルとして参照する。
- (2) A という部分モジュールに対する RTL の実装(IMPL<sup>RTL<sub>A</sub></sup>)を作成し、それに TLM レベルのラッパ

(wrapper)を被せたものを前述のテストベンチを用いて検証する。この際、SPEC<sup>TLM<sub>A</sub></sup>と挙動が異なっていないかどうかチェックする。通常は一気に RTL の実装レベルまで詳細化せずに段階的に詳細化を行って、元の仕様を満たした実装を作り上げる。

ここでいうテストベンチとは検証対象のモジュールへ入力するテストパターンを生成し、モジュールから出力されたパターンをチェックするプログラムを指す。

この手順の鍵は、SPEC<sup>TLM<sub>A</sub></sup>に対する「ネガ」としてのテストベンチを共用することで、IMPL<sup>RTL<sub>A</sub></sup>と SPEC<sup>TLM<sub>A</sub></sup>の等価性を暗に検証しているところにある。

この手順はシステムティックかつ工学的な手法ではあるが、2つの大きな問題があると思われる。1つは、結局、IMPL<sup>RTL<sub>A</sub></sup>の検証に関しては RTL シミュレーションを行わなければならないという事実。2つ目は、検証の質がテストベンチで用いるテストパターンに因っているということである。

## 2. 研究の目的

本研究課題ではこれら2つの問題点を解決すべく、以下のような技術の研究・開発を行う。

- RTL の実装を「抽象化」し、RTL の実装 + TLM ラッパと等価な TLM モデルを自動的に生成する技術
- 与えられたテストパターン集合をデータマイニング技術を用いて「学習」することによって、新規のパターンを自動的に生成する技術

「抽象化」はある意味、動作合成の逆プロセスと見ることが出来る。つまり、スケジュールされた演算実行のタイミングを解析して、データ依存性に関係しない動作の同期を取り去ることで無駄なシミュレーションサイクルの実行の低減を目指す。

もちろん、与えられた RTL 記述から仕様となる TLM 記述と等価なモデルを自動生成することはほとんど困難であるが、動作合成や等価検証と異なり、この「抽象化」を部分的に適用することができたとしても、シミュレーションの高速化に役立つので、この抽象化はインクリメンタルに適用することが可能であり、TLM 検証手法の効率化に対しては非常に実用的意義が高いと思われる。

従来、形式的検証の技術の一つとして「抽象化」の研究はなされているが、これは時相論理式などの形で記述された仕様の検証を行

上で不要となる情報を削ぎ落とすことで、検証モデルの状態爆発を防ぐ手法であり、本研究のようにシミュレーション速度の向上を目指すものではない。

また、TLM を用いた検証手法としては本研究で提案しているような「抽象化」技術の実現例は報告されていない。

2 番目のテストパターン生成に関しては、従来は人手で与えたテストパターン+ランダムパターンを用いるという手法が一般的で、ランダムパターンの生成に対して、検証項目のカバレッジなどのフィードバックをかけるという技術が提案されている。本研究では、人手で与えられたテストパターンをデータマイニング技術を用いて解析して、その特徴点を抽出し、それらを用いて半ばランダムに新規のパターンを生成する技術の開発を行う。単純なランダムパターンを用いる従来手法に比べて、より少ないパターン数で質の高い検証が行えることが期待できる。

以上のように、本研究提案では、システムレベル検証の主流になりつつある TLM 検証の枠組みのなかで、検証の高速化・高速化に役立つと思われる基盤技術の開発を行う。

### 3. 研究の方法

TLM 検証を高速化するための技術として、以下の項目の研究を行なう。

#### (1) トランスレータの開発

RTL 記述(Verilog)を入力として等価な TLM 記述を生成するトランスレータの開発を行う。

#### (2) 検証アルゴリズムの研究

抽象化の際の変換が正しいことを保障するための形式的検証アルゴリズムの研究を行なう。特に計算時間の短縮に注目して研究を行なう。

#### (3) シミュレーション高速化のための FPG A 用回路合成技術の研究

シミュレーションの高速化のために、対象回路(モデル)を FPG A で実装し、その FPG A を実際に動作させる方法がある。ここでは、シミュレーションの高速化を目的とした FPG A 用回路合成技術の研究を行なう。

### 4. 研究成果

(1) ハードウェア記述言語 Verilog-HDL を入力として等価な TLM 記述を生成するトランスレータの開発を行った。具体的には、まず Verilog-HDL 記述を構文解析した結果の

構文木のデータ構造を生成し、そこから個々の Verilog モジュールをインスタンス化した意味モデルの生成を行う。さらにこの意味モデルからシミュレーション用のモデルの生成を行う。

本来の Verilog-HDL の意味モデルではかなり自由なスタイルでハードウェアの記述を行えるが、実際に論理合成を行って回路を生成できる記述スタイルはかなり制限されている。具体的には、あらかじめ「クロック信号」と規定された信号線に同期して各記憶素子の値が切り替わる、というモデルである。そこで、論理合成対象の回路記述に対してはこのような単純化された意味モデルを用いてシミュレーションを行うことで基本となるシミュレーション速度の向上を図っている。

これらの技術そのものは学術的にはそれほど先進的なものではないが、シミュレーションモデルの抽象化という本研究の本題の技術を開発するための基礎となるものであり、今後の研究を進める上で不可欠のものとなっている。

#### (2) 検証アルゴリズムの研究

抽象化を行う際に用いる充足可能性判定問題(SAT) ソルバの開発を行った。既存のプログラムでトップクラスの MiniSat とほぼ同等の性能を達成している。これに述語論理の判定アルゴリズムを付加することで一階述語論理の検証が行なえる SMT ソルバを開発することができる。SMT は命題論理の証明を行なう SAT (Satisfiability) ソルバと、1階述語論理のいくつかの Theory の証明エンジンを組み合わせたもので、ここではハードウェアの検証に有効な、EUF (Equality with Uninterpreted Functions) と BV (Bit-Vectors) の証明エンジンを組み込んでいる。大まかには、トップレベルの論理論理の構造を充足する解を SAT ソルバで求め、その部分解に対応する各々の Theory が充足しているかを個々の証明エンジンで証明するというステップを実行する。証明を正しく行なうという意味ではすべての論理構造を一旦命題論理の形で表して、それを SAT ソルバで解けばよいが、その論理構造が充足不可能な場合の多くは、ごく一部の論理構造によって矛盾が起こっており、論理構造全体を命題論理の形で表す処理が非効率的である。そこで、論理構造を一旦 AIG (AND-INVERTER-GRAPH) と呼ばれる 2 入力 AND ゲートからなる論理回路の形式で表現し、AIG の構造に対して、論理最適化を行なうことで、簡単な矛盾の検出を行なうアルゴリズムの開発を行なった。局所的な

矛盾はSATソルバを起動することなく検出することができるため、効率化に役立っている。

検証問題そのものは、命題論理に限定してもNP完全問題であり、どのような問題でも効率よく解けるといえる保障はないが、実用上は正しく設計できていることの自明な確認を形式的検証ツールを用いて保障させるという使い方が多いと思われるので、上記の工夫は有効である。

### (3) シミュレーション高速化のためのFPGA用回路合成技術の研究

RTレベル(レジスタ転送レベル)あるいはゲートレベルで記述された回路を含むシミュレーションを高速化するためのひとつの方法として、その回路をFPGAで実現し、そのFPGAの回路を実際に動作させる手法がある。ここでは、RTレベルおよびゲートレベルの回路をFPGAに実装するためのテクノロジマッピングの研究を行なった。テーマは大きく分けて2つあり、一つ目は制御回路などのランダムロジックを最小段数のLUT(FPGAの基本論理素子)で構成するためのアルゴリズムの開発で、もうひとつは、算術演算回路のFPGA向け合成アルゴリズムの開発である。

論理回路を最小段数で合成するための既存アルゴリズムが存在するが、一般に、面積(論理素子の個数)削減を目的としたマッピング結果と比べてかなり面積の大きな回路を合成する傾向がある。一方、面積の増加を抑えた近似手法も提案されているが、すべてのマッピングの可能性を調べていないので、最小段数の保障がない。今回、最小段数を保障しつつ、面積オーバーヘッドを抑えたアルゴリズムを開発した。面積オーバーヘッドを抑える仕組みとして、一度マッピングした回路に対して、段数の増加がない範囲で再合成を行なうアルゴリズムを新規に提案している。

算術演算回路のFPGA向け合成アルゴリズムとしては、乗算回路の後半部分に現れる多入力加算回路を6入力LUTを用いて効率よく合成するアルゴリズムを開発した。従来の手法では、多入力加算を3入力2出力のCarry-Save-Adderを用いて構成しており、最近のFPGAで用いられている6入力LUTでは大きな無駄が生じる。そこで、3-to-2(3入力を2出力にまとめるもの)以外にも4-to-3, 5-to-3, 6-to-3の多入力加算器を組み合わせてよりLUT数の少ない回路を合成できるアルゴリズムを開発した。

### 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者に

は下線)

[雑誌論文] (計4件)

1. Taiga Takata and Yusuke Matsunaga, "Efficient Cut Enumeration Heuristics for Depth-Optimum Technology Mapping for LUT-based FPGAs", IEICE Trans. on Fundamentals, Vol. E92-A, No. 12, Dec. 2009, 査読有
2. Taiga Takata and Yusuke Matsunaga, "Area Recovery under Depth Constraint for Technology Mapping for LUT-based FPGAs", IPSJ Transactions on System LSI Design Methodology, Vol. 2, pp. 200-211, Feb. 2009, 査読有
3. Taeko Matsunaga, Shinji Kimura and Yusuke Matsunaga, "Framework for Parallel Prefix Adder Synthesis Considering Switching Activities", IPSJ Transactions on System LSI Design Methodology, Vol. 2, pp. 212-221, Feb. 2009, 査読有
4. Sho Kodama and Yusuke Matsunaga, "Binding Refinement for Multiplexer Reduction", Transactions on System LSI Design Methodology, Vol. 2, No. 2, pp. 43-52, Feb. 2009., 査読有

[学会発表] (計5件)

1. 松永 裕介, "高位合成における種々の最適化手法について", 第23回回路とシステム軽井沢ワークショップ, 2010年4月19日、長野県, 招待講演
2. Taiga Takata and Yusuke Matsunaga, "A Power-aware Post-processing under depth constraint for LUT-based FPGA Technology Mapping", Proc. of International Workshop on Logic and Synthesis 2009, pp. 332-339, Aug. 2, 2009., Berkeley, CA, USA, 査読有
3. Taeko Matsunaga, Shinji Kimura, and Yusuke Matsunaga, "Multi-Operand Adder Synthesis on FPGAs using Generalized Parallel Counters", Proc. of International Workshop on Logic and Synthesis 2009, pp. 222-228, Aug. 1, 2009., Berkeley, CA, USA, 査読有
4. Taiga Takata and Yusuke Matsunaga, "An efficient cut enumeration for depth-optimum technology mapping for LUT-based FPGAs", ACM Great Lakes Symposium on VLSI, pp. 351-356, May. 11, 2009., Boston, MA, USA, 査読有
5. Taeko Matsunaga, Sinji Kimura and

Yusuke Matsunaga, ``Synthesis of parallel prefix adders considering switching activities'', In proceedings of ICCD2008, pp.404-409, Oct. 14, 2008., Tahoe, CA, USA, 査読有

[その他]  
ホームページ等

## 6. 研究組織

### (1) 研究代表者

松永 裕介 (Yusuke Matsunaga)  
九州大学・システム情報科学研究院・准教授  
研究者番号：00336059

### (2) 研究分担者

馬場 謙介 (Kensuke Baba)  
九州大学・附属図書館研究開発室・准教授  
研究者番号：70380681

### (3) 研究分担者

吉村 正義 (Masayoshi Yoshimura)  
九州大学・システム情報科学研究院・助教  
研究者番号：90452820