審査の結果の要旨

論文提出者氏名　サクンコンチャック タンヤパット

本論文は、「Formal Verification of Synchronization in System-Level Design」（システムレベル設計における形式的同期検証に関する研究）と題し、デジタル電子機器設計の上位設計段階であるシステムレベル設計において、その正しさを数学的に証明する手法である形式的検証手法、特に、複数の並列動作するプロセスの同期に関する検証技術について研究したもので７章より構成されている。

第１章は Introduction（序論）であり、研究の背景と目的を述べている。システム LSI に代表されるように近年のデジタル機器設計は、大規模・複雑化しており、設計期間の長期化が大きな問題となっている。特に、設計の正しさの確認作業である設計検証に要する期間が相対的に長くなっており、特に大規模設計の場合、全体の設計期間に占める設計検証期間の割合が 80%を超えるほどになっている。本論文では、その現状を分析し、従来のシミュレーションやエミュレーションだけでなく、設計の正しさを数学的に証明する形式的手法の必要性・有用性、特に、大規模・複雑な設計に対応するために、システムレベル設計と呼ばれる設計の初期段階での形式的検証の重要性を述べている。次に、従来の形式的検証技術のサーベイを簡単に行い、システムレベル設計における検証問題として、複数のプロセスが互いに同期しながら、正しく作業を進めることを検証する「同期検証」の重要性を説明し、システム設計における同期検証を形式的に行うフレームワークの構築と、それの上での効率的な形式的検証手法を本論文の目的としていることを述べている。

第２章は Preliminaries and Backgrounds（準備と背景）であり、本論文で使用する既存技術の説明を行っている。まず、デジタル機器設計におけるシステム設計とは何かについて説明し、システム設計記述に利用する SpecC 言語の要点を簡単に紹介している。本論文では、SpecC 言語で記述されたシステム設計記述に対して、形式的に検証を行う。さらに、本論文で利用する形式的検証技術に関する既存技術の説明を行っている。

第３章は Synchronization Verification（同期検証）であり、本論文が提案するシステム設計に対する同期検証手法を説明している。本論文で提案している同期検証のフレームワークでは、与えられた検証すべきシステム設計記述に対し、まず設計の抽象化を自動的に行う。この抽象化により大規模設計に対しても適用可能な検証手法となっている。次にその抽象化された設計記述に対して、複数の並列動作するプロセスが正しく時間軸上で同期しているかを調べるために、時間関数の式に展開し、それを解析する作業をモデルチェッキングと呼ばれる形式的検証手法を用いて行う。この解析に矛盾がなければ元の設計は正しいと言える。一方、設計に誤りがある場合には、解析の結果、反例が検出される。この反例は、抽象化された設計に対するものであるため、抽象化によっては、元の設計では成り立たない反例である場合がある。このため、検出された反例を元の設計で確かめる作業を行う。もし、元の設計でも反例となっている場合には、設計誤りが見つかったことになる。しかし、元の設計では反例となっていない場合には、検証のために行った設計の抽象化に問題がある（抽象化しすぎて

いる）ので、抽象化処理の改良を行う。以下、これらの作業を元の設計に対する反例が検出されるか、または、設計が正しいと証明されるまで、繰り返していく。本章では、この検証フレームワーク全体の処理の流れの説明を例題も利用して詳しく行っている。システムレベル設計の同期検証に対するこのような形式的検証のフレームワークは本論文によって、新規に提案されたものである。

第４章は Abstraction and Model Checking（設計の抽象化とモデルチェッキング）であり、３章で説明された検証フレームワークにおける、設計の抽象化と抽象化された設計のモデルチェッキング手法による解析に関し、そのアルゴリズムについて例題を使いながら、詳細に説明している。設計の抽象化では、設計記述の同期に関する部分以外を取り除く処理を行っており、元の設計が大規模な場合でも、大幅に設計規模を縮小できる。また、それに続くモデルチェッキングによる解析では、時間関数による条件式の形で定式化するという新しいアルゴリズムを提案しており、実時間処理に関する記述を含む設計記述の検証が可能となっている。このように、大規模・複雑な設計に適応可能な新規アルゴリズムが提案されている。

第５章は Abstraction Refinement（設計抽象化の改良）であり、４章で行った設計抽象化が、誤って反例を提示してしまう場合に、その誤りの反例を提示しないように設計の抽象化を自動的に改良する手法について、検討し、新規手法を提案している。問題の性質上、発見的手法となっているが、後の実験結果が示すように、効率的に作用する場合の多い手法が提案されている。

第６章は Implementation and Experimental Results（実装と実験結果）であり、３章、４章、５章で提案している同期検証のフレームワークの実装法の検討と実装結果の説明、それに、数種の例題に対して、実際に同期検証を行った結果について報告している。実装は、設計の抽象化とそのモデルチェッキングによる解析（４章で説明した部分）については、自動化が実現されている。また、設計の抽象化の改良については、効率的に検証を行うために一部対話的に処理する形でツールとして実現されている。実際に制御システムや通信プロトコルの検証に適用した結果、元の設計に対し、抽象化された設計が極めて小さくなることが示されており、本論文で提案する同期検証のフレームワークは十分実用的価値があることが、実験的に証明されている。

最後の第７章は、結論であり、本論文の研究成果をまとめるとともに、今後の方向について議論している。

以上、本論文は、大規模デジタル電子機器のシステム設計において、もっとも設計誤りを生じやすい並列プロセス間の同期に関し、大規模設計にも適用可能な検証フレームワークを提案、実装し、例題で評価することでその有効性を示したもので電子工学の発展に貢献するところが少なくない。

よって、本論文は博士（工学）の学位請求論文として合格と認められる。