

|      |    |   |   |
|------|----|---|---|
| 報告番号 | ※甲 | 第 | 号 |
|------|----|---|---|

## 主 論 文 の 要 旨

論文題目 Efficient System-Level Design Space Exploration for Large-Scale Embedded Systems  
(大規模組込みシステムにおける効率的なシステムレベル設計空間探索)

氏 名 安藤友樹

### 論 文 内 容 の 要 旨

近年、複雑さが増す組込みシステムを実現するため、組込みシステムの機能は増加している。組込みシステムの機能が増加することで、組込みシステムはより高度な計算と通信の処理能力が要求されている。これらの処理能力の要求に対処するため、組み込みシステムでは、専用ハードウェアやマルチプロセッサなどが演算処理装置として利用されるようになった。

一般に、組み込みシステムは限られた環境で使用されるため、実行時間、ハードウェア面積や消費電力といったシステムの性能は、厳しい性能要件を満たす必要がある。複数の演算処理装置で構成される組込みシステムを設計する場合、設計者は演算処理装置に対するシステムの機能割当てを示すマッピングを決定する必要があり、このマッピングに依存してシステムの性能が決まる。そのため、設計者が大規模な設計空間から効率的に適切なマッピングを見つけることが、システムが厳しい性能要件を満たす上で重要となる。

複雑な組込みシステムを設計するため、システムレベル設計が提案された。システムレベル設計で重要な概念は、高い抽象度にてシステムを設計することと、システムレベルで設計空間を探索することである。システムレベル設計では、まず設計者がシステムを抽象度の高いモデルで記述する。その後、システムの性能を評価するために、モデルはシミュレーション記述やターゲット用の実装に変換される。システムの性能評価結果が性能要件を満たさない場合、設計者はモデルを変更し、再び変更後のモデルの評価を行う。設計者は、モデルの修正と評価を繰り返すことで、システムが性能要件を満たすように設計空間を探索する。

システムレベル設計を実現するために、いくつものシステムレベル設計ツールが開発されてきた。システムレベル設計ツールの一つとして SystemBuilder がある。SystemBuilder は、設計者が決定したマッピングに従い、モデルから自動的にター

ゲット実装を生成する。生成された実装は、シミュレーションツール及び FPGA (Field-Programmable Gate Array) の両方で実行可能である。SystemBuilder が自動的にターゲット実装やソフトウェアとハードウェア間の通信インタフェースを合成するため、設計者は、容易に異なるマッピングを評価できる。

本論文は、まず、AES 暗号化システムの設計事例により SystemBuilder を評価する。設計事例による評価により、SystemBuilder はパイプラインシステムの設計に有効であることを明らかにする。さらに、システム設計を効率化するためには、1) 実装の制限、2) 長い評価時間、3) 不十分なシステムの性能向上支援、の3つの問題があることを明らかにする。一つ目の問題は、実装の制限である。多くのシステムレベル設計ツールは、機能をソフトウェアかハードウェアのいずれかに割当て可能である。しかしながら、現在のシステムレベル設計ツールは、ハードウェアの共有など他の割当て方に対応していない。二つ目の問題は、長い評価時間である。たとえ SystemBuilder が実装を自動生成したとしても、実装の生成と性能評価に非常に長い時間が必要である。従って、SystemBuilder のみで多くのマッピングを評価することは困難である。三つ目の問題は、不十分なシステムの性能向上支援である。SystemBuilder のプロファイル機能により、実行時間のボトルネックは明らかにできる。しかしながら、プロファイル機能のみでは、現在の複雑な組み込みシステムの性能改善方法を検討することは困難である。これらは SystemBuilder のみの問題ではなく、他のシステムレベル設計ツールの問題でもある。そのため、より効率的なシステムレベル設計を実現するためには、これらの問題を解決することが重要である。

本論文は、上記の問題を解決するため、3つのツールを提案する。3つのツールは Extended SystemBuilder, Mapping Explorer, Improvement Analyzer である。Extended SystemBuilder は SystemBuilder の拡張であり、複数のアプリケーション間でハードウェアを共有するモジュールを生成できる。ハードウェアを共有するモジュールの生成が可能になることで、一つ目の問題が解決し、設計者はより性能の良い実装を得ることができるようになる。Mapping Explorer は効率的なマッピングの探索を実現するツールである。効率的な探索アルゴリズムである pareto-update search を利用することで、マッピングの探索数が大幅に削減できる。そのため、Mapping Explorer はマッピングの探索数を削減することで、探索時の性能評価時間を短縮でき、二つ目の問題を解決する。たとえ Mapping Explorer が適切なマッピングを発見できない場合でも、Improvement Analyzer が設計者によるボトルネックの特定を支援する。さらに、Improvement Analyzer は複数の性能改善方法の候補を一覧として出力することで、設計者によるモデル修正の支援を行う。ボトルネックの特定および、性能改善方法の候補を出力することで、Improvement Analyzer は三つ目の課題を解決する。

上記3つのツールが連携することで、マッピングの探索とボトルネック解析は自動化され、効率的な設計空間探索が実現される。そのため、提案手法を用いることで、システム設計者はマルチプロセッサと専用ハードウェアモジュールを搭載した大規模な組み込みシステムを、効率的に設計可能である。

本論文は3つのツールの詳細を述べ、設計事例を用いてツールを評価する。さらに、設計事例により設計空間探索の効率化効果を示す。