

データストリームを利用した高度組み込みシステム向け協調設計に関する研究 (032109004)

A Co-design Method for Advanced Embedded Systems using Data Stream

大森 洋一 国立大学法人九州大学 大学院システム情報科学研究院
Yoichi Omori Graduate School of Information Science and Electrical Engineering,
Kyushu University

研究期間 平成 15 年度～平成 17 年度

概要

ソフトウェア・ハードウェア協調設計は、組み込みシステム開発効率化の本命と目されている。しかし、広範な普及のためにはシステム仕様の獲得法、仕様の実現可能性を判定する定量評価法など未解決の課題がある。本研究ではデータストリームを利用したプロトタイピング開発環境の構築により、仕様の可視化、設計性能の定量評価を行ない、これらの課題を解決する。具体的には、データフロー記述エディタを実装し、命令追加などによりカスタマイズしたプロセッサ用のプログラムを出力し、シミュレータとエミュレータのシームレスな統合により定量的な性能評価を行なうシステムを構築し、ハードウェアとソフトウェアを一体とした最適化を可能とした。

Abstract

Software-hardware codesign is regarded as a promised method for embedded system development. There are, however, unsolved subjects, such as a method of acquiring system specification and a quantitative method which judges the realization of specifications. In this research, these problems are resolved by visualization of specification and quantitative evaluation of performance on a prototyping environment using data stream. A data flow description editor was developed, which generates a program for customized processors with new operations. The program is quantitatively evaluated by integration of a simulator and an emulator. The system enables the optimization of hardware and software as unified way.

1. まえがき

特定用途向けのシステムを 1 チップに集積する System-on-Chip (SoC) 手法は、ULSI 集積技術の飛躍的な発展に伴い、信号遅延の短縮、実装面積の縮小、電力消費量の減少などの利点により応用範囲を広げつつある。SoC に適した設計技法としてハードウェア・ソフトウェア協調設計 (コデザイン) が本命視されており、実問題への応用も始まっている。

しかしながら、コデザインの効率的な実現法は未だに確立されておらず、

- 1) クライアントの要求定義、獲得手段
 - 2) データフローグラフの定量的な実行性能評価
 - 3) タスク分割方針と定量的なリソース見積り
- といった課題が残されている。

本提案では、チップ実装技術や計算機アーキテクチャに関する深い知識を必要としない使いやすいコデザイン実現を目的として、ストリーム処理に注目したシステム設計法を提案し、これらの課題を解消するツールの開発と検証を行なう。

久世らにより提案されたストリームは、複数の機能間で次々とデータを受け渡しつつ、一連の処理を行なうという演算形態である。ストリームは本質的にはデータフロー記述であり、システム仕様を人間にとって直観的に表現できる。本研究では、仮想機械上で実行可能な拡張データフローグラフ (EDFD) を定義し、システム構築の際に、抽象度に依存しない計算モデルとして利用する。EDFD という一貫したモデルで継続的に設計を詳細化することで、複数の抽象度に渡る正当性検証が容易になる。

2. 研究内容及び成果

本提案システムの全体構成を図 1 に示す。今回提案する開発環境は、主として組み込みシステム設計者が使用する

ことを想定している。クライアントの要求を拡張データフローグラフ (Enhanced Data Flow Graph、EDFD) としてプロトタイピングし、挙動を確認した要求を仕様として、ストリームの各機能として分割・生成するところまでを対象とする。

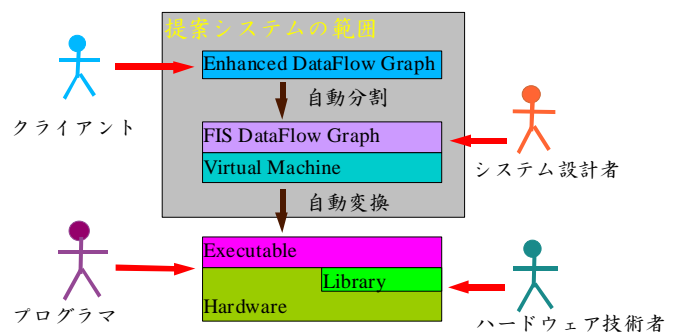


図 1 開発環境概念図

2.1 クライアントの要求定義、獲得手段

必ずしも専門家ではないクライアントの要求を実装可能な仕様としてまとめるのは、困難である。どのような要求があるのかをクライアント自身も曖昧にしか認識していなかったり、設計中に変更したりする可能性もある。EDFD は、従来のデータフローに対して、コデザイン向けの拡張を施しており、システム設計仕様の中間表現として EDFD を利用することにより、仮想的な環境においてシステムの挙動を随時実行可能なプロトタイプが実現でき、クライアントとの意思疎通が容易になる。

実行可能な仕様記述を可能にするために、提案者らは、一般にプロセッサに用意されている命令群に対して実プログラムで用いられる命令の種類はそれほど多くない点、

また複雑な命令のほとんどは単純な命令列で置き換えられる点に注目し、仮想データ駆動プロセッサの命令セット **Fundamental Instruction Set (FIS)** を提案した。FISにより、特定のハードウェアに依存しない機能量の定量評価が可能となる。既存のプロセッサを利用する場合はその命令セットを用いばよい。

2.2 データフローグラフの定量的な実行性能評価

EDFD において確定した機能は、ストリームの順に適当な大きさに分割され、ソフトウェアとハードウェアのいずれかまたは両方により実装される。本提案では入力された EDFD をその時点での仕様として扱い、プロトタイプによる随時の仕様変更を想定する柔軟な戦略をとる。このために、機能の最適化における制約条件と目的関数を交換し、時間的制約条件をみたすように機能分割を行なう。すなわち、EDFD で表現された処理にかかる時間を正確に見積もるのは難しいので、その機能がみたすべき時間的制約を満たすようにハードウェア資源を投入する。投機実行や先読みまで考慮すれば、ほとんどの機能は対象システム中の最小処理時間で実現できる。あとは許される遅延に応じて、ハードウェアの共有やソフトウェアによる実現により、必要なハードウェア量を最適化すればよい。

2.3 タスク分割方針と定量的なリソース見積り

抽象的な機能をハードウェアに変換する作業を安直に行なうと、次々とデータが流れてくるような一部のアプリケーションを除き、機能の重複が多く、ハードウェア利用効率の非常に低いシステムになってしまう。これはハードウェア・コストの上昇や配線遅延の増大につながり、コデザインが普及しない大きな要因となっている。さらに、ハードウェア量の制約を考慮した現実的な応用では、時間的な制約を満たすための解は複数あるのが普通であり、どの選択肢が最適なのか決めるのは、熟練したシステム設計者にとっても容易ではない。

本提案では、ハードウェア・パラメータのうち、同時に実行される処理数である並列度に注目して、2段階のタスク生成の指標として利用する方法を採用する。図2にEDFDの記述例を示す。

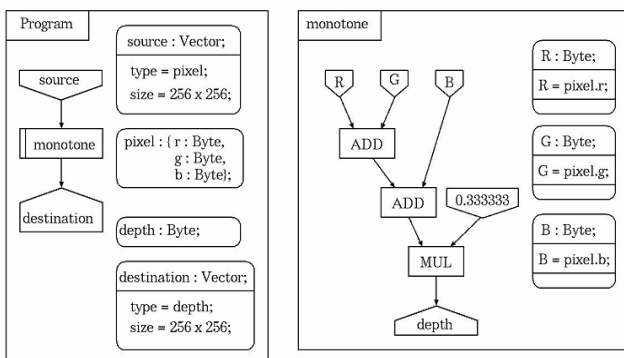


図2 EDFD の記述例

3. むすび

本研究の研究成果として、構造化エディタとシミュレータ/エミュレータの混合評価が挙げられる。

EDFD は図的な表現であり、習熟していない設計者にも分かりやすいとはいえ、慣れは必要である。そこで、構造化エディタにより、入力時に適切なストリーム表現になるようにノードの接続検査、ストリームの型検査といった

サポートを行なう。このプログラミングの視覚化により、システムの挙動の直観的な表現を行ない、システム設計者のより正確な設計仕様の確認、より複雑な設計の把握を可能とし、正確な仕様獲得を容易にする。正確な仕様記述が得られたならば、その仕様から実際に動作するシステムを設計するための手段が次の課題となる。なぜなら、ソフトウェアまたはハードウェアにおける高級言語記述は、既に確立した技術であるが、システムの仕様をソフトウェアまたはハードウェアに切り分ける手法などが残されているからである。

プロトタイプの実現に関しては、提案済みの仮想命令セット FIS の利用に加えて、マクロフローモデルの提案およびハードウェアを利用した高速シミュレーション手法の開発を行ない、精度・速度の両面を満足する基本的な技術開発に成功した。

マクロフローモデルは、パケットの流れる密度であるフローレートと処理速度の関係を定式化することにより、個々のパイプラインステージにおける遅延評価を不要とし、計算量を大幅に削減した性能見積りモデルである。さらに、仮想命令セット FIS およびハードウェアを利用した高速シミュレータの開発を行ない、精度・速度の両面を満足する基本的な技術開発に成功した。

既存ハードウェアを利用した高速シミュレーション技術は、シミュレーション速度を劇的に向上させることが知られている。そこで、トレース駆動シミュレーションと組み合わせることにより性能見積り精度とシミュレーション速度の両立を図るなどの工夫を行なった。実際のハードウェアとの組み合わせによる評価を行ない、現在も性能面の改良を行なっている。このようなシミュレーションをサポートするハードウェア機能の追加の提案を行った。

時間制約をみたす設計の保障に関しては、モデリングの段階における制約条件記述法に関する提案を行った。また、設計の自動検証や再利用性の向上を目的として、機能モデルと実効も出る文理による、より一般的な非機能的な制約の記述についても提案した。こうした性能見積りとの組み合わせにより、モデル段階での保障を行う。

【誌上発表リスト】

- [1] Shuji Sannomiya, Yoichi Omori, and Makoto Iwata, "A Macroscopic Behavior Model for Self-Timed Pipeline Systems", Proceedings of the 17th International Workshop on Parallel and Distributed Simulation pp133-140 (2003. 7.13)
- [2] Yoichi Omori, "Stream: A System Design Method Based on Dataflow", Proceedings of the 7th Joint Workshop on System Development pp37-48 (2004.4.5)
- [3] Yoichi Omori, "Exhaustive Formalization of First-step Requirements", Proceedings of the 11th International Workshop on Future Software Technologies 2005 Vol NF pp33-40 (2005.11.9)