

Title	動的リコンフィギャラブルプロセッサにおける入出と演算のオーバーラップを用いたループネストの高速化に関する研究
Author(s)	荒木, 光一
Citation	
Issue Date	2008-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/4295
Rights	
Description	Supervisor:井口 寧, 情報科学研究科, 修士

動的リコンフィギャラブルプロセッサにおける 入出力と演算のオーバーラップを用いた ループネストの高速化に関する研究

荒木 光一 (610004)

北陸先端科学技術大学院大学 情報科学研究科

2003年2月7日

キーワード: 動的リコンフィギャラブルプロセッサ、動的部分再構成、データ入出力、コンテキスト。

1 はじめに

マルチコンテキスト型動的リコンフィギャラブルプロセッサは、任意の回路を何度でも再構成可能である。また、デバイス内に回路情報であるコンテキストを複数保存できるので、瞬時に再構成を実行することが可能である。本研究では、NECエレクトロニクス社の Dynamically Reconfigurable Processor(DRP)を対象デバイスとした。DRPは、Processing Element(PE)がALUベースで構成されているので、LUTベースで構成されている Field Programmable Gate Array(FPGA)よりストリーム処理を高速に処理することが可能である。DRP内にはTileと呼ばれるDRPのコアが複数あり、Tileは、他のTileに依存することなく独自に再構成を行う動的部分再構成を実行することができる。このような特徴を活かして、CPUでボトルネックとなるループネストをDRPでハードウェア化し並列に処理することで高速に処理できる可能性がある。

しかし、DRPのI/Oビット幅が非常に小さいため、並列処理を行うために必要なデータの入力と演算結果の出力が各1クロックでは不可能である。従って、DRPの典型的なストリーム処理の流れは、1) 複数クロックでデータを入力しデバイス内のメモリに保存する。2) 並列に構成された演算器で処理、3) 複数クロックで演算結果を出力となる。この手法では入出力データが多い処理の場合、データ入出力の実行時間がボトルネックになってしまう。従って、データ入出力時間を削減することが課題となる。また、DRPには16個の回路データを保存することが可能だが、ある処理が要求する回路データ数が16個以上である場合、実行中にホストメモリから保存できない回路データをダウンロードする必要がある。この結果、ダウンロード時間が実行時間のボトルネックとなる。従って、マル

チコンテキスト型動的リコンフィギャラブルプロセッサにとってデバイス内に保存する回路データ数を削減することもまた重要な課題である。

2 提案手法

本研究では、データ入出力時間の削減とデバイス内に保存する回路データ数を削減することを目的とし、2つの手法を提案する。1つ目の手法は、データ入出力と演算をオーバーラップさせることによってデータ入出力を隠蔽する。ループ内の処理は、同じ処理をループカウント回行うので、各 Tile にループ内の処理を並列処理のように割り当てる。そして、DRP の動的部分再構成を利用して、ある Tile が演算を行っている間に、他の Tile にデータ入力を行う。これにより、データ入出力クロックを隠蔽でき、全体の実行クロック数を削減することが可能である。2つ目の手法は、1クロックで入力可能なデータからデータ駆動型の回路データを生成することで、回路データ数を削減する。DRP のコンテキストは、Control Data Flow Graph(CDFG) を分割することで生成される。つまり、ある時点で DRP 上に実現している回路は、CDFG の一部であるということである。従って、CDFG を細かく分割すれば、動作周波数は増加するが、回路データ数は増加し、逆に CDFG を粗く分割すれば、動作周波数は低下するが、回路データ数を減少させることが可能である。本提案では、1クロックで入力されるデータを基に CDFG の分割を行う。入力されたデータで実行可能な CDFG 中の演算子を探索し、その演算子の集合を1つの回路データとして生成する。この結果、動作周波数の低下が考えられるが、1つ目の提案でデータ入出力クロックを隠蔽し全体のクロック数を削減できるので、従来の DRP 処理法より高速に処理できる可能性がある。

3 評価

提案手法の有効性を示すために画像処理のラプラシアン・フィルタを用いて、実行クロック数の削減数、実行時間の比較と回路データ数の削減率の評価を行った。比較を行うための従来手法は、1. で示した従来の DRP のストリーム処理を用いた。実行時間の比較で一般的な比較を行うために Intel 社 Pentium4 2.8GHz を用い、コンパイラには gcc3.2.2 を用いた。コンパイルオプションとして最適化オプションの O3 を付属してコンパイルを行った。DRP の実装では、理論上最大5つの Tile でオーバーラップすることが可能だが、実際には配線が不可能なため3パターン (Tile 数 2,3,4) でオーバーラップ処理を行った。

実行クロック数の削減数は、最大で従来手法の全クロック数の55%に削減することができた。CPU との実行時間の比較では、従来手法が151%、提案手法が最大で230%高速に処理することができた。回路データ数の削減率において提案手法は、従来手法の回路データ数を約50%にまで削減することができた。

4 まとめ

本研究では、データ入出力時間の削減とデバイス内に保存する回路データ数を削減することを目的とした。これらの問題に対して、データ入出力と演算をオーバーラップさせることによってデータ入出力を隠蔽する手法と1クロックで入力可能なデータからデータ駆動型の回路データを生成する手法を提案した。画像処理のラプラシアン・フィルタで従来のDRPのストリーム処理法と比較した結果、本研究の提案手法は、目的を達成できることが分かった。