

FPGAによるCPUアクセラレータ

1H-1

宮田耕自（筑波大学）

丸山 勉（同）、星野 力（同）

1. はじめに

近年の半導体の高集積化によって汎用マイクロプロセッサには様々な機能を付加することが可能となってきた。しかしトランジスタの集積度の上昇ほどには演算速度が上がらないのが現状である。従って既存の汎用マイクロプロセッサとは違ったアーキテクチャを持つFPGA (Field Programmable Gate Array) に注目する。

FPGAは書換可能なハードウェアのことであり、現在さまざまな分野での研究がなされている。FPGAは、問題毎に回路構成を最適化でき、さらに外部からの要求に応じてFPGAの内部機能を変更できる。これらの特徴を利用し、命令レベルの並列性が十分ある、パイプライン化が可能である、また並列化が可能である等の問題においては汎用マイクロプロセッサの演算速度をはるかに凌ぐことが可能である。

現在高速化が望まれるほとんどの処理はこの条件を満たしていると考えており、このような問題に対して汎用マイクロプロセッサの代わりにFPGAを用いることは非常に有効である。

本論文では、FPGAの特徴と問題点、高速演算用FPGAアーキテクチャの構成、そしてFPGAシミュレータの概要について述べる。

2. FPGAの特徴と問題点

既存のFPGAは、ランダムロジックの実現を主目的としている。このため、これらのFPGAの基本演算子はN入力1出力のLookup Tableで構成されており、任意のN入力に対する論理演算を非常に高速に行うことができる。このような構造を持つFPGAをアクセラレータとして用いることを考えた場合、以下のような問題点がある。

- ① 32bit程度のデータ幅の演算速度が非常に遅い
キャリー伝播のための専用ハードウェアを持たないため汎用マイクロプロセッサの10分の1程度の速度が上限である。
- ② 入出力pinの動作速度が遅い
全てのI/O pinがプログラマブルであり、その動作

An accelerator with a new FPGA architecture

Kouji Miyata (University of Tsukuba)

Tsutomu Maruyama, Tsutomu Hosino

周波数は、汎用マイクロプロセッサのキャッシュメモリの10分の1程度である。

③書き換え速度が遅い

基本演算子（1bit演算を行う）毎に書き換えデータが必要であるため、書き換えデータ量が大きい。

このように既存のFPGAをCPUアクセラレータとして使うには問題が多い。従って高速演算用FPGAアーキテクチャを考える必要がある。

3. 高速演算用FPGAアーキテクチャ

高速演算のためのFPGAアーキテクチャを考えるに当たり、ソフトウェアで記述されたアルゴリズムを高速に実行することを考える。ハードウェアに特化したアルゴリズムを用いることにより、より高速な処理が可能になることもあるが、アルゴリズムの開発が困難であることが多く、ソフトウェアのアルゴリズムに則した計算を高速化することで、より広い範囲の問題を容易に扱うことが可能となる。

ソフトウェアで記述されたアルゴリズムのハードウェア化による高速化を考える場合、

- ①命令レベルの並列処理
- ②パイプライン処理
- ③並列処理

が有効な手段である。

図1に高速演算用FPGAのアーキテクチャを示す。

図1において、ALUのデータ幅は32bitであり、32bit演算を基本単位とする。これに対応して、ネットワークの切り換えも32bit単位でのみ行う。ただしALUは、ビット分割可能な構成を考えており（例えば8bit×4等）、SIMD的な動作に対応する。ALUの演算はコンテキストメモリによって指定される。このコンテキストメモリのアドレスを変えることにより、ALUで実行される演算を動的に変えることができる。

ロジックブロックは、内部レジスタとPLAから構成される。入力はALUで行われる条件判断結果等であり、ALUに付属するレジスタのEnable、セクタの切り換え、コンテキストメモリのアドレス等の動的な制御を実現する。

このような構成を採ることにより、FPGAの書き換えは32bit演算単位での指定となるため、書き換えデ

ータ量を著しく削減することができる。また、動的に書き換え可能な部分を導入することにより、ALUのより有効な利用(再利用)が可能となり、限られたハードウェア量の中で、より大規模な処理を実現することが可能となる。

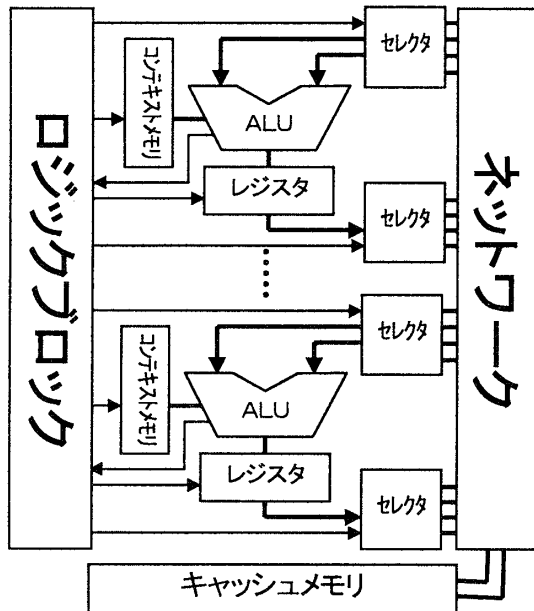


図1 高速演算用FPGAの構成

入出力に関しては、メモリに対してのみ行うので入出力 pin はプログラマブルである必要がない。このためメモリアクセスに専用化し高速なメモリアクセスを実現することができる。さらにキャッシュメモリを用いることで、アクセス速度は汎用マイクロプロセッサと同程度が実現可能である。

このようなアーキテクチャにより、命令レベルでの最大限の並列性を引き出し、最適なパイプライン処理、並列処理による高速処理が実現できる。

4. FPGAシミュレータ

図1のアーキテクチャ上で問題を実行することを考えた場合、データ依存関係のある演算が割り振られたALU間でのみ、ネットワークでのデータ転送、及びロジックブロックでの制御が可能であれば十分である。つまり全てのALUの出力が、全てのALUの入力と接続可能である必要はなく、またロジックブロックにおいても、あるALUの条件判断の結果に応じて、全てのALUの制御が必要となるわけではない。従って、ネットワーク、ロジックブロックは完全結合である必要はなく、より簡潔な構成を採ることができる。また、セレクトの数もすべて均一である必要はない。

どの程度まで、ネットワーク、ロジックブロック、セレクト等を簡略化できるかに関しては、トイプログラ

ムを用いた評価ではなく、より多くの現実的な問題による評価が必要となる。

このためには、シミュレータによる評価が必要である。シミュレータの必要とするパラメータにはALUの数、ネットワークの配線パターン、ロジックブロックの構成、セレクトの数、コンテキストメモリの大きさ等がある。

ここで、使用可能なハードウェア量が決まっていると仮定した場合、ALUの数を優先すると、ネットワーク、ロジックブロックに使われるハードウェア量は減少し、データ転送経路が限定されるためALUの使用効率は落ち性能は低下する。逆にネットワーク、ロジックブロックを充実させるとデータ転送経路の選択の自由度が高くなるためALUの使用効率は上がるが、ALUの数そのものは減少し性能は低下する。

従って、限られたハードウェア量の中で演算速度が最大となる構成を見つける必要がある。このバランスは各問題で異なると考えられるため、より多くの問題に対する評価が必要である。

このためには、ソフトウェアで記述されたアルゴリズムのHDLへのコンパイラ(但しソフトウェアには制約が必要)と、そのHDLからアーキテクチャへのマッピングを行うソフトウェアが必要であり、現在これらの開発を進めている。

5. 終わりに

以上、FPGAの特徴と問題点、高速演算用FPGAアーキテクチャ、及びFPGAシミュレータについて述べた。現在、FPGAシミュレータを作成中であり、今後、高速演算用FPGAアーキテクチャの評価を行う予定である。

参考文献

- [1] ALTERA 「FLEX10K Data Sheet」
- [2] XILINX 「XC6200 Product Description」
- [3] FCCM 1997
「Garp: A MIPS Processor with a Reconfigurable Coprocessor」
John R. Hauser and John Wawrzynek
- [4] 並列処理シンポジウム JSPP'98
「DRAM混載型仮想ハードウェアHOSMIIの構成と性能」
柴田裕一郎 宮崎英倫 高山篤志 凌曉萍 天野英晴
- [5] CQ出版社
「パソコンによるプログラマブルロジックの設計」
トランジスタ技術編集部編