

ARM Cortex-A9デュアルコア内蔵FPGAと16コア内蔵マルチコア・プロセッサを搭載 Parallella搭載のFPGAのカスタマイズ方法と iDCTアクセラレータの実装

鈴木 量三郎, 片岡 啓明 Ryouzaburo Suzuki, Hiroaki Kataoka

“Parallella”は、ARM Cortex-A9デュアルコア内蔵FPGAと、16コアを内蔵したマルチコア・プロセッサを1枚のカードに実装した並列処理に最適な評価ボードです。応用編の今回は行列演算が得意なGPUと比較しながら、Parallellaの使いどころを考察します。そしてParallellaの特徴でもあるFPGAをカスタマイズし、FPGA内にiDCTを高速化するアクセラレータを実装する事例を紹介します。

1 Parallellaはどんな用途に 適している？

● 依存関係がなければ分割しただけ並列処理が高速化

前回の記事(本書No.8, pp.78-89)ではParallellaのデモとしてマンデルブロを実行しました。並列計算の例としてよくあるデモで、その理由は分割した領域同士で依存関係がないため余分な通信が発生しないことと、入力データが必要ないことです。

通常のアプリケーションでは計算のために入力が必要です。また、場合によっては計算時に分割された隣の領域からデータをもらう必要があるかもしれません。

Parallellaで入力と出力のあるアプリケーションの例題として、画像のグレースケール変換を考えます。グレースケール変換は、分割した領域同士に依存関係はありません。そのため分割しただけ並列化が可能で、その分だけ高速化が期待できます。

このように、領域分割ができる問題は各コアに処理を分割して任せれば、理論的には分割しただけ速くなるはずですが、実際には分割に要した(余分な)処理の

コストや通信のコストも考慮する必要があります。

● マルチコア・プロセッサ Epiphany

ParallellaにはEpiphanyと呼ばれるマルチコア・プロセッサが搭載されています。図1にEpiphanyの内部構成を示します。eLink IOがあるだけで、DDR SDRAM上に格納した画像データへ直接アクセスはできません。

Epiphanyは、画像のデータをZynqにつながったDDR SDRAMからとってきています。また結果もメモリへ転送しています。Epiphanyの構成をもう一度見てみましょう。Epiphany側コアにはRAMがあります。このRAMはプログラムから高速にアクセスできます。一方でZynqへのメモリ・アクセスにはメッシュのネットワークを通るため高速にはアクセスできません。より高速に処理を実現しようとするなら、いったんDMAでEpiphanyのコア内にデータをもってきておいて、処理が終了したところでEpiphanyからDMA転送する必要があるでしょう。

● Epiphanyが得意な処理

Epiphanyは領域分割して並列処理が可能なもの、

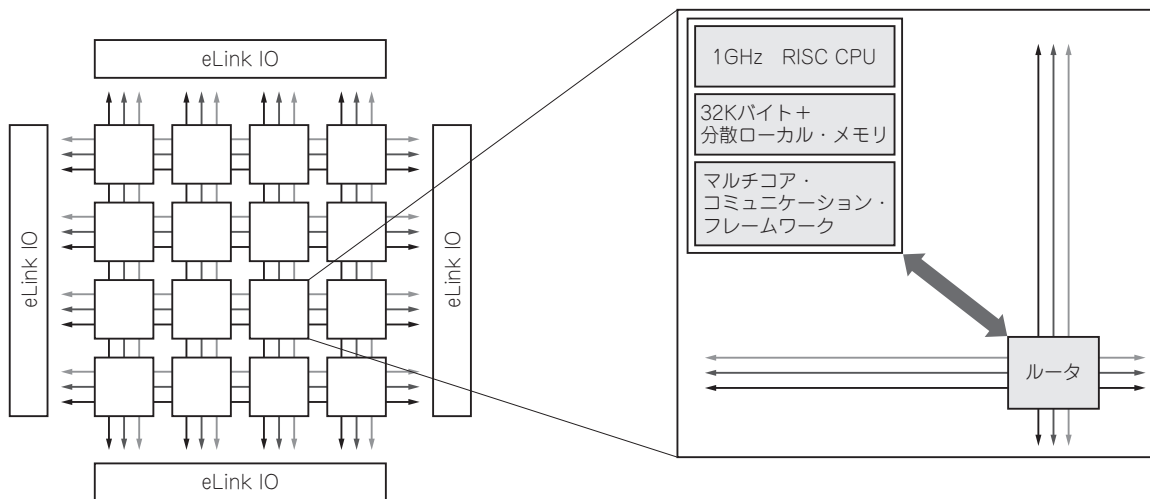


図1 Epiphanyの内部構成