OpenCLはGPUだけじゃない

並列処理の記述が得意なOpenCLをFPGA開発に活かす
Altera OpenCLを試す
～実行環境構築までの設定方法とその実力～

山際 伸一 Shinichi Yamagiwa

ムーアの法則が限界を迎え，マルチコアやマルチプロセッサ，メニーコア・プロセッサ構成のCPUやGPU（Graphics Processing Unit）が次々登場しています．これらのプロセッサの性能を引き出すには，並列動作を効率良く記述できる言語が必要です．そこでOpenCLが登場します．ここではAltera SDK for OpenCLを，ユーザの立場で実際に動作させてみた様子について解説します．

1 プロセッサの性能追求の歴史

OpenCLの話を始める前に，それが登場する背景ともなる，プロセッサの性能追求の歴史について少し説明します．

● 命令レベル並列性からスレッドレベル並列性へ

ムーアの法則により，プロセッサの性能が向上してきた時代は終わりを迎えています．ハイエンドPC向けプロセッサの動作クロック周波数も，ここ数年は3GHz前後で停滞気味となっています．しかしそれでも，より性能の高いプロセッサが要求され続けています．

今までのプロセッサは，命令レベルでの並列性を抽出する技術を中心に研究開発が行われてきました．**図1**に示すようなアセンブリ・コードの場合，依存性のない命令がプロセッサ内部でハードウェアで解析され，並列実行されています．

このような方式をスーパースカラと呼びます．つまり，これまでのプロセッサは，幾つの命令をどれだけ並行実行できるかが性能を左右しており，その複雑な機構をどれだけ詰め込めるかが勝負の分かれ目でした．最近のプロセッサでは4～8命令が同時実行されており，さらに，コンパイラがそれらをサポートするために，並列性を抽出する仕組みが取り入れられています．

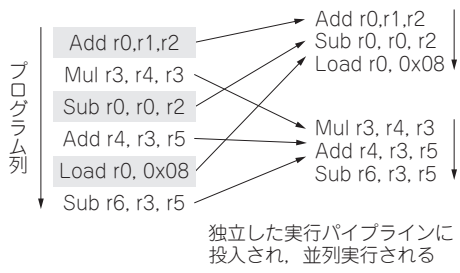

図1 スーパースカラ・プロセッサの仕組み

● マルチコア・プロセッサの登場

しかし，このような命令レベルの並列性を抽出するには，とても複雑で，大量のハードウェア・リソースを必要とします．そのため，回路規模が大きくなるに従い，クロック・スピードを向上させていくスケーラブルな改良が頭打ちになり，さらに回路規模の増加による歩留まり（不良品の発生確率）の悪化が伴い，命令レベルの並列性を重視することに限界が見えはじめました．そこで命令レベルの並列性ではなく，処理の単位（スレッド）での並列性を重視する方向に転向しているのが現代のプロセッサ技術のコンセプトです．すなわち，マルチコア・プロセッサです．

最近のPC向けプロセッサでは，4～12個程度のコアを集積したものが主流となっています．**図2**に示すように，このプロセッサの構成としては全てのプロセッサ・コアから共通のアドレスでアクセス可能なメモリ空間が存在することを前提とした共有メモリ型のアーキテクチャをとり，複数のスレッドがプロセッサに割り当てられ，並列実行されています．

例えば，Javaのようなプログラムを考えてみましょう．**図3**に示すように，GUI，計算処理，イベント処理，描画など，複数の処理が一つのプログラムにパックされていますが，それらはスレッドとして実装され

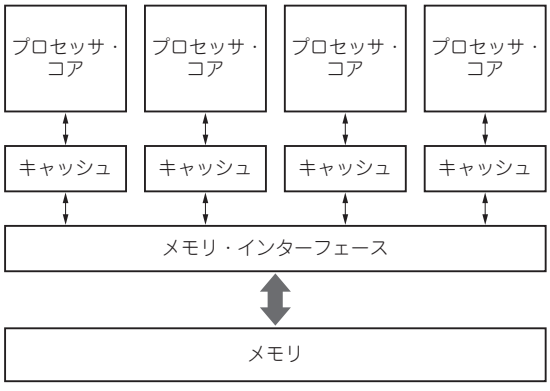

図2 マルチコア・プロセッサ