

FPGA を用いたメニーコア・アーキテクチャ SMYLEref の評価環境の構築

ゲン チュオン ソン^{†1} レイ ジャオ^{†1} 近藤 正章^{†1}
平尾 智也^{†2} 井上 弘士^{†2}

我々は、組み込みシステム向けの高性能かつ低消費電力メニーコアプロセッサ実現のために、独立行政法人新エネルギー・産業技術総合開発機構 (NEDO) のプロジェクト「極低電力 回路・システム技術開発 (グリーン IT プロジェクト)」の中で、「低消費電力メニーコア用アーキテクチャとコンパイラ技術」の研究課題を実施している。本プロジェクトでは、1) 組み込みシステム応用を意識した高効率な超並列処理の実現、2) 大幅な動作時消費電力の削減、3) ソフトウェア生産性の向上、を組み込みシステム向けメニーコアプロセッサの重要な研究課題と位置付け、それらを解決するための一つの方策として、仮想アクセラレータ実行プラットフォームとしてのメニーコアアーキテクチャ SMYLEref とそのプログラム開発環境の構築を行っている。本稿では、SMYLEref のアーキテクチャについて述べるとともに、FPGA を用いた SMYLEref の評価環境について紹介し、その上での並列ベンチマークプログラムの初期評価結果を示す。

1. はじめに

現在のマイクロプロセッサは、数個から十数個程度のプロセッサコア (以降、コアと略す) を 1 チップに搭載したマルチコア構成が主流となっているが、将来的には数十個から数百個というさらに多数のコアを搭載したメニーコアが有望視されている。我々は、高性能・低消費電力なメニーコアプロセッサの実現と、メニーコアプロセッサの組み込みシステムへの応用展開を目的として、独立行政法人新エネルギー・産業技術総合開発機構 (NEDO) のプロジェクト「極低電力 回路・システム技術開発 (グリーン IT プロジェクト)」の中で、「低消費電力メニーコア用アーキテクチャとコンパイラ技術」の研究課題を九州大学、立命館大学、電気通信大学、株式会社フィックスターズ、株式会社トプスシステムズの連携体

^{†1} 電気通信大学 大学院情報システム学研究所
Graduate School of Information Systems, The University of Electro-Communications
^{†2} 九州大学 大学院システム情報科学研究院
Faculty of Information Science and Electrical Engineering, Kyushu University

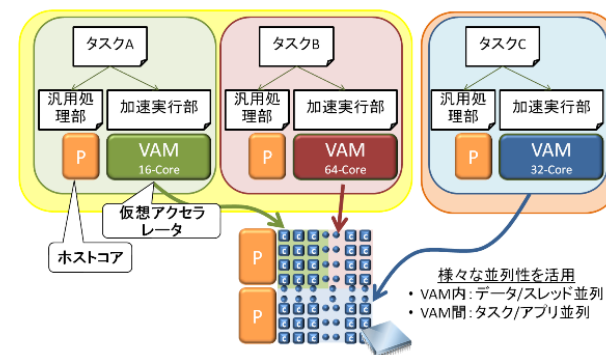


図 1 仮想アクセラレータ (VAM: Virtual Accelerator on Many-core) の概要

制で実施している。

我々は、メニーコアの本質は低性能ではあるが小面積かつ低消費電力なコアを多数搭載して極めて高い並列処理性能を得ることであると考え、この本質に基づくメニーコアならではの研究開発を行うことで、高性能・低消費電力なメニーコアシステムの実現を目指している。そこで、1) 組み込みシステム応用を意識した高効率な超並列処理の実現、2) 大幅な動作時消費電力の削減、3) ソフトウェア生産性の向上、を重要な課題と位置付け、それらを解決するための一つの方策として、仮想アクセラレータ実行プラットフォームとしてのメニーコアアーキテクチャとそのプログラム開発環境の構築を行っている。本研究のポイントは今までにない新しいアプローチとして「仮想アクセラレータ (VAM: Virtual Accelerator on Many-core)」の概念を導入する点にある。多数の小規模コアで構成されるメニーコアを仮想アクセラレータ実現のためのハードウェアプラットフォームとして活用し、一方でコンパイラ、あるいは並列プログラムはアプリケーション特性に応じて自らが仮想アクセラレータの構成を決定しつつ、それに基づきコード生成やアプリケーションの実行を行う (図 1 参照)。このように、ソフトウェアに対してハードウェアアーキテクチャの決定権を与えることで、高い並列処理性能を得ることを狙う。

VAM ではマルチスレッド・マルチプログラム環境での効率的な実行を目的として、小規模コア群に対して複数の仮想アクセラレータを同時にマッピング可能とすることで、コア数にスケール可能な性能向上を実現する。一方、各コアは 0.5V といった極低電圧でのプログラム実行を可能にする。この際には、メニーコアの豊富なハードウェア資源を有効活用する

ことで、極低電圧領域において生じるばらつきといった諸問題の解決を図る。また、仮想アクセラレータ実行を前提としたきめ細かな電源電圧/動作周波数制御など、メニーコアならではの低消費電力化技術を確認する。これにより、コア数にスケール可能な低消費電力化を可能にする。さらに、プログラムの開発から実行までのシームレスな最適化を実現するために、各種 API を策定してそれに基づくソフトウェア開発環境を構築する。これらを通して、先に述べた課題の解決を図る。

上述した仮想アクセラレータ VAM 実現のためのハードウェアプラットフォームの 1 つとして、我々は *SMYLEref* メニーコアアーキテクチャを開発している。また、その評価・検証を行う環境として FPGA を用いた評価を構築している。ソフトウェアによるシミュレーションだけでなく、HDL により設計を行い、かつ FPGA 上でマッピングして実際にメニーコアプロセッサとして動作させ、高速・正確な評価を行うことが目的である。さらに、本評価環境をできる限り公開し、多くのメニーコアプロセッサアーキテクチャやソフトウェアの開発・評価環境として広く利用してもらうことを想定している。これにより、将来的な組み込みシステムにおけるメニーコアプロセッサの普及と、本分野の産業力強化に貢献したいと考えている。

本稿では、*SMYLEref* メニーコアのアーキテクチャについて述べるとともに、開発中の FPGA によるメニーコアプロセッサ評価環境を紹介する。また、本評価環境を用いたの並列ベンチマークプログラムによる初期性能評価結果を示す。

2. メニーコア・アーキテクチャ *SMYLEref*

図 2 にメニーコア・アーキテクチャ *SMYLEref* の概要を示す。*SMYLEref* はシンプルな数個 (図の例では 8 個) のプロセッサコアをバスで結合したクラスタと呼ぶブロックを、2 次元メッシュのオンチップネットワーク (NoC) で結合したアーキテクチャである。本章では、*SMYLEref* の構成要素であるプロセッサコア、クラスタ構成と NoC、そして VAM 用の拡張について説明する。

2.1 プロセッサコア

本研究では、*SMYLEref* の各コアとして、科学技術振興機構 CREST の研究領域「情報システムの超低消費電力化を目指した技術革新と統合化技術」中のプロジェクトである「革新的電源制御による次世代超低電力高性能システム LSI の研究 (代表: 中村宏東京大学教授)」¹⁾ で開発された *Geyser*²⁾ プロセッサコアを用いた。*Geyser* は MIPS R3000 アーキテクチャをベースとしたシンプルなプロセッサコアであり、実 LSI チップへの実装実績もあ

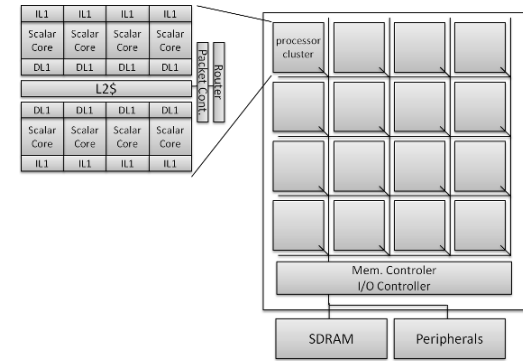


図 2 *SMYLEref* のアーキテクチャ

る。さらに、*Geyser* を FPGA 上に実装しての Linux 動作も確認されたコアであることから³⁾、*SMYLEref* におけるベースのプロセッサコアとして採用した。なお、*Geyser* は最粒度パワーゲーティングの研究用途に開発されたプロセッサコアであるが、本研究では最粒度パワーゲーティングの機能は用いない。

Geyser コアは 8KB (2Way セットアソシアティブ、64B ラインサイズ) の L1 命令、および L1 データキャッシュ、また命令により制御可能な 16 エントリの TLB を持つ。なお、データキャッシュはライトバック方式を採用している。

2.2 クラスタ構成と NoC

クラスタには、コアや L2 キャッシュを備えるプロセッサクラスタと、チップ外部とのインターフェースの役割を持つペリフェラルクラスタがある。1 つのプロセッサクラスタは、複数個の *Geyser* コアが Cluster バスを通して接続される。また、L2 キャッシュ、およびルータもこのバスに接続される。ペリフェラルクラスタは SDRAM コントローラや I/O コントローラを持つ。主記憶や I/O アクセスの際にはプロセッサクラスタとペリフェラルクラスタ間でデータ転送が行われる。

プロセッサクラスタ間、およびプロセッサクラスタ・ペリフェラルクラスタ間はルータを通してネットワークオンチップにより接続され、パケットスイッチング方式でデータ転送が行われる。複数コアでクラスタを構築し、さらにクラスタ間をネットワークで接続するという階層的な構造をとることにより、通信を自クラスタのキャッシュにできる限り閉じ込めることで、通信路の混雑による性能低下の回避を狙う。

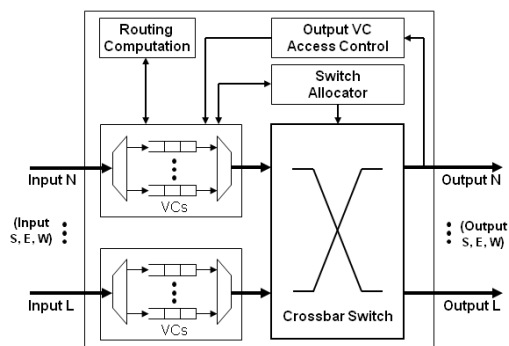


図 3 ルータアーキテクチャの概要

クラスタ間を接続するオンチップネットワークポロジは、スケーラビリティの確保と拡張性を考慮して 2 次元メッシュを採用している。この際、デッドロックを防止するために XY ルーティングプロトコルを用い、次元順ワームホールルーティングを行う。SMYLEref ので採用するルータアーキテクチャの概要を図 3 に示す。隣接ノードと Cluster パスとの通信のために 5 本の入出力ポート (N, S, E, W, L) を持つ。なお、データのビット幅は 32 ビットであり、各ポートに 2 個の VC を持つ。本ルータは仮想チャンネル (VC) ルータアーキテクチャをベースとし、低コスト・高性能なデータ転送を行うことができる On-the-fly ルータ⁴⁾ を利用する。通常、ルータは RC (Routing Computation), VA (Virtual Channel Allocation), SA (Switch Allocation), ST (Switch Traversal) の 4 段階の処理を行い、パケットを次のホップ先に転送する。On-the-fly ルータは、SA の確定をしながら VA を行うことで、ST ステージをクリティカルパスより除去し、さらにルックahead・ルーティング技術を利用して RC ステージと SA ステージを並列に行う。これにより、全ての処理を 1 サイクルで行うことが可能となり、各ルータで通信パケットを単一サイクルで転送することが可能になる。

2.3 VAM 向けの拡張

メニーコアでは、如何にして搭載コア数に見合った性能向上を実現できるかが最も重要である。1 個のチップに集積された多数のコアの稼働率を最大に引き上げ、コア数に対してスケール可能な性能向上を実現しなければならない。そのためには、極めて高い並列性の確保が必要不可欠となる。一般に、HPC (High Performance Computing) 分野における科学技

術計算や画像処理などでは高い並列性が内在する 경우가多く、メニーコア向けの有望なアプリケーションとして期待されている。しかしながら、例えば 100 個以上のコアの搭載を想定した場合、このような並列性の高いプログラムにおいてさえも高いスケーラビリティを実現することは容易でない。また、組込みシステム分野へのメニーコア応用を見据えた場合、比較的並列性の低いアプリケーションも多く存在するため、これらへの対応も考慮する必要がある。そこで SMYLEref では、第 1 章で述べたように、VAM の導入によりマルチスレッド・マルチプログラム実行を可能にし、プログラムレベルの並列性を有効に活用することで高いスケーラビリティを実現する。VAM でのプログラム実効効率を高めることを目的とした SMYLEref の主な特徴を以下に示す。

- 再構成可能 L1 キャッシュの搭載：各コアに搭載される L1 キャッシュは、キャッシュ領域もしくは SPM (Scratch Pad Memory) 領域として利用することができる。全ての領域をキャッシュもしくは SPM として使用するだけでなく、これらの混在も可能である (つまり、一部の領域をキャッシュ、残りの領域を SPM として使用可能)。キャッシュ構成に関しては、VAM 上で実行されるアプリケーションの特性に応じてコンパイラが決定することができる。
- 分散共有 L2 キャッシュ・メモリの拡張：SMYLEref では LLC (Last Level Cache) として分散共有 L2 キャッシュを搭載している。各クラスタには L2 キャッシュ・スライスが配置されており、各 VAM に割り当てられたクラスタに属する L2 キャッシュ・スライス群で LLC を構成する。VAM 間での L2 キャッシュ競合を回避するための特別なアドレス変換機構を有しており、分散共有キャッシュ・アクセスにおける空間局所性を高めることができる。
- グループ・ハードウェア・バリアのサポート：SMYLEref では (基本的に) 任意の数のコアを VAM に割り当てることができる。そこで、柔軟かつ高速なバリア同期を実現するため、グループ指定可能なメニーコア向けハードウェア・バリアをサポートする。バリア同期に参加するコアを指定することができ、それぞれの VAM に割り当てられたコアのみを対象とした同期が可能である。

3. FPGA における評価環境の構築

メニーコアのアーキテクチャやソフトウェアを検討・開発していくにあたり、その評価環境の構築は重要な課題である。これまで、多くのアーキテクチャの研究では機能レベル、あるいはサイクルレベルのソフトウェアシミュレータが評価環境として使用されてきた。しか

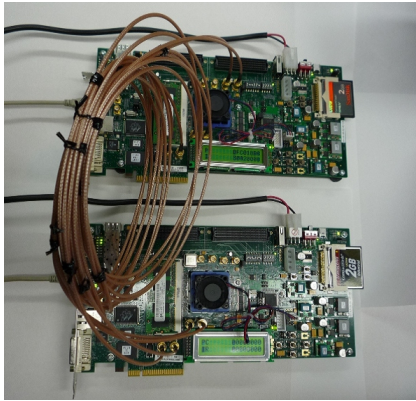


図 4 評価環境の外観

表 1 ML605 ボードと Virtex-6 チップの仕様

ML605 評価ボード	
FPGA デバイス	Virtex-6 XC6VLX240T
SDRAM	DDR3 SO-DIMM
搭載 I/O ポート	UART, USB, DVI 出力, CF, SMA 等
クロック入力	200MHz & 66 MHz オシレータ
Virtex-6 (XC6VLX240T)	
テクノロジー	65nm CMOS, 1.0V
Logic Cells	241,152
CLB Slices	37,680
Block RAM	14,975 Kbit
ユーザー I/O 数	720

し、一般的に多くのソフトウェアレベルシミュレータは、複数プロセッサ上で並列あるいは並行プログラムをシミュレーションする場合でも、逐次的にシミュレーションをすることが多く、コア数が数十や数百にもなるメニーコアプロセッサを評価する場合、プログラムの一部を評価するだけでも長時間を要する。さらに、OS などのシステムソフトウェアの動作も評価を行うとなると、より長いシミュレーション時間がかかってしまう。

そこで、我々は FPGA を利用して SMYLEref の評価環境を構築することにした。実際の複数のコアを FPGA 上に構築し、それらを並列に動作させることで、評価に要する時間の短縮が期待できる他、OS やシステムソフトウェアの実行もそのまま評価・検証を行うことができる。また、実際に HDL により設計することで、正確かつ詳細な評価・検証を行うことが可能になる。そのため、ソフトウェアレベルのシミュレータでは動作や実行が抽象化され、その結果隠蔽されてしまうような性能上のボトルネック（例えばコア間の同期など）も正確に評価を行うことができ、特にメニーコアプロセッサの評価環境として有用であると考えている。

3.1 ハードウェア、および開発環境

今回の実装では、Xilinx 社製の FPGA チップである Virtex-6 を搭載する ML605 評価ボードを、評価環境のプラットフォームとして利用した。ML605 評価ボード、および搭載する Virtex-6 チップの仕様を表 1 に示す。図 4 は本評価環境の外観である。なお、回路設計には Verilog HDL を用い、論理合成、マッピング、配置配線には Xilinx 社の ISE を利

用した。

3.2 コア/ペリフェラルクラスタの実装

クラスタとして、コアクラスタとペリフェラルクラスタの 2 種類を構築する必要がある。各コアクラスタは最大で 8 つの Geyser コアを持ち、2.2 節で述べたルータを搭載する。なお、現在 L2 キャッシュやコヒーレンス制御、バリア同期サポートのためのハードウェアなどは開発中である。

ペリフェラルクラスタは、SDRAM や UART, SysACE 等のコントローラを備えており、それらは Xilinx 社が提供する PLB (Processor Local Bus) バスに結合される。ペリフェラルクラスタ上のルータも PLB と接続されており、これによりオンチップネットワークを介して、各コアとペリフェラルが接続されることになる。

3.3 クラスタ間の通信

評価環境として利用する FPGA (Virtex-6 XC6VLX240T) は、Geyser コアを 8 個程度しか搭載することができず、1 つの ML605 ボードだけでは数十、あるいは数百のコアを持つようなメニーコアプロセッサの評価を行うことができない。そこで、複数のボードを結合し、ルータ間の通信をボード間に跨って行うことでメニーコアの評価環境を実現することにした。

ボード間の通信には高速シリアル通信インターフェースである rocket I/O を使用する。このために、ルータモジュール内にボードを跨って通信を行う際に使用する通信モジュールを開発した。本モジュールは、ISE の IP コアとして提供されている aurora と呼ばれる高速シリアル通信プロトコルを利用している。なお、物理的なボード間通信インターフェースとして SMA 規格の通信インターフェースを利用する。ML605 評価ボード上は 3 リンク分

の SMA コネクタしかないが、FMC の拡張ボードを利用することにより、1 ボードあたり追加で 8 リンク分の SMA コネクタを追加することが可能である。2 次元メッシュネットワークであれば、各方向あたり 2 リンク (双方向で通信のため) の合計 8 リンクがあれば一つのクラスタを構築し、隣接クラスタと結合しつつネットワークを形成することが可能である。したがって、上記の通信インターフェースを用いることで、理論的には無限個のコアを持つメニーコアの評価環境を構築することが可能である。

本評価環境では、プロセッサコアのクロック周波数として数十 MHz 程度を想定している。rocket I/O によるルータ間は 5Gbps でシリアルデータの通信が可能であるため、制御情報を含めた 1 パケット (現在は 38bit) を転送するバンド幅は、コアクロックに対して十分なバンド幅を確保できる。一方で、aurora モジュール中でシリアル/パラレルデータ変換をする際に、クロックの同期を行う必要性から通信レイテンシはルータ同士をボード内で直接接続する場合に比べて約 180ns 程度余分にかかる。そのため、プロセッサコアから見ると通信レイテンシは数サイクル増加してしまう。ただし、例えば I/O アクセスはさらに長いレイテンシがかかるため、この数サイクルのレイテンシ増加は評価結果に大きな影響は与えないと考えられる。

4. 評価

4.1 評価の仮定

構築した SMYLEref の FPGA 評価環境を用いて、いくつかの並列プログラムに関して性能評価を行う。本評価には ML605 ボードを 1 枚使用し、1 つのボード上に 4 つの Geysers コアをもつクラスタを 2 個、ペリフェラルクラスタを 1 個持つ SMYLEref を構築し、合計 8 コアの環境で評価を行った。L2 キャッシュは現在実装中であり、今回の評価では各コアの L1 命令、およびデータキャッシュのみを用いる。各機能ブロックへ供給する周波数は以下の設定を用いた。

- Geysers コア: 10MHz
- クラスタ内部バス・ルータ・ペリフェラルバス (PLB): 5MHz
- DDR3-SDRAM: 100MHz

SMYLEref を評価するにあたり、基本動作の確認と基礎データの取得を目的に、まずは通常の並列アプリケーションベンチマークを用いて評価を行う。評価には共有メモリ用のベンチマークプログラムである SPLASH2⁵⁾ の中から FFT, LU, Cholesky のプログラムを用いた。SPLASH2 ベンチマークは並列処理のためのプリミティブが m4 マクロにより記述

されており、処理系に合わせてマクロを展開することで、実際の並列処理 API などを含むプログラムコードを生成する。本評価では、SPLASH2 ベンチマークに付属されているバリア同期を含む pthread コードを生成するマクロを利用した。

コンパイラは MIPS 用のコードを生成するよう構築した gcc 4.4.6 を用いる。また、Geysers は浮動小数点演算器を持たないが、今回評価に用いたプログラムは浮動小数点演算を含むプログラムであるため、gcc の浮動小数点演算のソフトウェアエミュレーション (Soft Float) 機能を利用して実行する。

4.2 並列処理 API

SMYLEref の基礎評価を目的に、SMYLEref 評価環境向けの最低限の機能を持つ簡易版 pthread ライブラリを実装した。本簡易版 pthread ライブラリを用いることで、pthread を用いて並列化された SPLASH2 のコードをほぼそのままコンパイル・実行することが可能である。実装した pthread 関数はスレッド生成・終了を制御する pthread_create, pthread_join などの他、他排他制御用の関数 (pthread_mutex_lock, pthread_mutex_unlock など) やバリア同期用の関数 (pthread_barrier_wait など) である。

現在の評価環境ではまだキャッシュコヒーレンス機構を実装していない。一方、SPLASH2 はキャッシュコヒーレントな共有メモリ型の並列計算機が前提であるため、正しい実行結果を得るためにはいくつかの工夫が必要となる。本評価では、共有データ領域 (ヒープ領域) をキャッシュ経由でアクセスしない uncachable 領域とし、またバリア同期、および排他制御の度に毎回各コアで L1 データキャッシュをフラッシュすることにした。これにより、プログラム自体を変更することなく、キャッシュされたデータの一貫性を保証することができる。

なお、printf など、C ライブラリに含まれる基本的な関数については、Geysers の評価環境に付属されているものを拡張して利用した。

4.3 評価結果

まず、表 2 に、実 LSI チップへ実装した場合のある程度の参考になると考え、FPGA 上に実装した際の Geysers コア、ルータ等の各モジュールの回路規模を示す。表より、コアの面積が他のモジュールに比べて大きいことがわかる。8 コアで 1 つのプロセッサクラスタを形成することを想定すると、ルータなどの他の機構の面積はほとんどチップ面積には影響しない。

次に、図 5 に、各プログラムを 1 コアのみを用いて実行した場合に対する複数コアで実行した場合のスピードアップの結果を示す。図より、全プログラムともにコア数を増やすことで性能が向上していることがわかる。特に fft では、コア数にほぼ比例したスピードアップ

表 2 回路規模

	スライス数 (%)	Flip-Flops (%)	LUTs (%)
コアクラス			
Geysler コア (1 コア)	2,897 (7.69%)	5,836 (1.94%)	9,641 (6.40%)
ルータ	1,170 (3.11%)	838 (0.28%)	3,400 (2.26%)
パケットコントローラ	80 (0.21%)	79 (0.03%)	189 (0.13%)
ボード間通信機構	220 (0.58%)	568 (0.19%)	407 (0.27%)
ペリフェラルクラス			
I/O コントローラ	1,596 (4.24%)	6,007 (1.99%)	2,577 (1.71%)
ルータ	1,170 (3.11%)	838 (0.28%)	3,400 (2.26%)
パケットコントローラ	112 (0.30%)	94 (0.03%)	230 (0.15%)

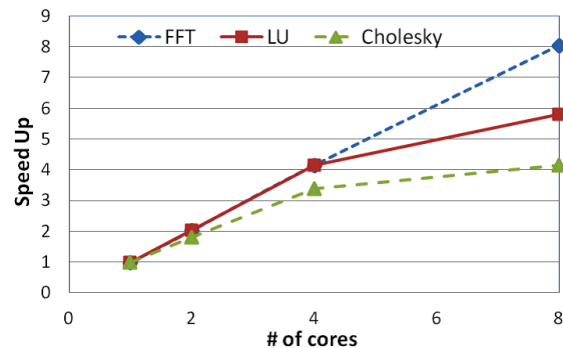


図 5 評価結果

プを得ることができている。一方で、LU や Cholesky では、コア数を増加させただけの性能向上が得られていない。これは、今回の評価環境の実装では、同期や排他制御の度にキャッシュをフラッシュしてしまっていること、またバリア同期には排他制御を利用したソフトウェア実装を用いており、同期にかかる時間が長いことが理由としてあげられる。今後、キャッシュコヒーレンスの実装やバリア同期のハードウェアサポートを行うことで、並列処理効率は改善すると考えられる。

5. 関連研究

近年、マルチコア、メニーコアのアーキテクチャやコンパイラについて多くの研究や開発が行われている。文献 6)、7) では、OSCAR マルチコアプロセッサ、および自動並列化コンパイラについて述べられている。また、実際のマルチコアチップも開発され⁸⁾、コンパイラによる自動並列化や消費電力削減の評価も行われている。最近ではメニーコアプロセッサ向けに、コンパイラによるコヒーレンス制御に関する研究も行われている⁹⁾。

文献 10) では小容量の FPGA を搭載したボードを複数接続し、スケラビリティを確保しつつ柔軟かつ高速なメニーコアプロセッサのシミュレーション環境の実現を目指す ScalableCore システムが提案されている。また、その他にも RAMP¹¹⁾ に代表されるように、マルチコアやメニーコアプロセッサの研究用途に FPGA を用いて評価環境を構築する研究も多く行われている。

近年では、実際数十コアを持つメニーコアプロセッサの試作や、商用の製品も登場してきている。Intel は 80 コアを持つプロセッサを試作し 1 チップで 1 テラフロップス以上を達成することを示している¹²⁾。このプロセッサは浮動小数点コアを 8×10 の 2 次元メッシュで構築している。ATAC プロセッサ¹³⁾ は、オンチップ光ネットワークを利用して高速なグローバルブロードキャストネットワークを提供し、1024 コアを持つメニーコアプロセッサを構築している。また、上記の光ネットワークを利用した ACKwise というスケラブルな Directory-based キャッシュコヒーレンスプロトコルも提案されている。さらには、Tilera 社の TILE64 のように商用のメニーコアプロセッサも登場している¹⁴⁾。TILE64 は、 8×8 のホモジニアスコアを Mesh オンチップネットワークを利用して結合したものである。

6. まとめと今後の課題

本稿では、組み込みシステム向けの高性能・低消費電力メニーコアプロセッサ実現に向けて導入した仮想アクセラレータの概念を実現する一つのアーキテクチャである SMYLEref と、その評価・検証を行う環境として開発している FPGA による評価構築について述べた。また、開発中の評価環境を用いて SPLASH2 ベンチマークプログラムによる初期評価実験を行い、並列処理効率を評価した。評価の結果、まだ改善点があるものの、評価した全プログラムにおいてスピードアップを確認した。

今後は、L2 キャッシュの実装やコヒーレンス機構など基本機能の実装を行うほか、VAM 向けの拡張を実装し、種々のアプリケーションで評価を行い、メニーコアプロセッサのあるべきアーキテクチャと、ソフトウェア開発環境を構築してゆく予定である。さらに、構築した評価環境に関する設計データはできる限り公開することで、多くのメニーコアプロセッサのアーキテクチャやソフトウェアの開発・評価環境として広く利用してもらうことを考えている。これにより、将来的な組み込みシステムにおけるメニーコアプロセッサの普及と、本分野の産業力強化に貢献したいと考えている。

謝辞 本研究を行うにあたり、ご議論・ご助言頂いた東京農工大学の並木美太郎教授、ならびに並木研究室の皆様へ感謝いたします。なお、本研究の一部は、独立行政法人新エネルギー・産業技術総合開発機構（NEDO）のプロジェクト「極低電力 回路・システム技術開発（グリーン IT プロジェクト）」の支援により行われたものである。

参 考 文 献

- 1) 中村 宏, 天野 英晴, 宇佐美 公良, 並木 美太郎, 今井 雅, 近藤 正章: 革新的電源制御による超低消費電力高性能システム LSI の構想, 情報処理学会研究報告 ARC-173-14, pp.79-84, 2007.
- 2) 関 直臣, ほか: MIPS R3000 プロセッサにおける細粒度動的スリープ制御の実装と評価電子情報通信学会論文誌 D, Vol. J93-D, No. 6, pp. 920-930, 2010.
- 3) 茂木 勇, 木村 一樹, 砂田 徹也, 並木 美太郎: 省電力 MIPS プロセッサ Geysler の FPGA 版評価ボードへの Linux の移植, 情報処理学会研究報告, OS-114-9, 2010.
- 4) S.T. Nguyen, and S. Oyanag: A High-throughput Router Architecture with On-the-fly Virtual Channel Allocation for On-chip Networks, IPSJ Trans. on ACS, Vol. 4, No. 2, pp.84-93, 2011.
- 5) S.C. Woo, M. Ohara, E. Torrie, J.P. Singh, A. Gupta: The SPLASH-2 programs: characterization and methodological considerations, Proc. of the 22nd International Symposium on Computer Architecture (ISCA-95), pp.24-36, 1995.
- 6) H. Kasahara, et.al.: A Multi-grain Paralliz-ing Compilation fScheme on OSCAR, Proc. 4th Workshop on Language an dCompilers for Parallel Computing, 1991.
- 7) K. Kimura, Y. Wada, H. Nakano, T. Kodaka, J. Shirako, K. Ishizaka and H. Kasahara: Multigrain Parallel Processing on Compiler Cooperative Chip Multiprocessor, Proc. of 9th Workshop on Interaction between Compilers and Computer Architectures (INTERACT-9), 2005.
- 8) M. Ito, et. al.: An 8640 MIPS SoC with Independent Power-off Control of 8 CPU and 8 RAMS by an Automatic Parallelizing Compiler, Proc. of IEEE International Solid State Circuits Conference (ISSCC2008), 2008.
- 9) 間瀬正啓, 木村啓二, 笠原博徳: 並列化コンパイラによるソフトウェアコヒーレンシ制御, 情報処理学会研究報告 Vol.2010-ARC-189-7, Apr. 2010.
- 10) 高前田伸也, 佐藤真平, 藤枝直輝, 三好健文, 吉瀬謙二: メニーコアアーキテクチャの HW 評価環境 ScalableCore システム, 情報処理学会論文誌コンピューティングシステム, Vol.4, No.1, pp.24-42, 2011.
- 11) RAMP - Research Accelerator for Multiple Processors,
<http://ramp.eecs.berkeley.edu/>
- 12) Vangal, S. R., et al.: An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS, IEEE Journal of Solid-State Circuits, Vol. 43, No. 1, pp.29-41, 2008.
- 13) Kurian, G., et al.: ATAC: A 1000-Core Cache-Coherent Processor with On-Chip Optical Network, The 19th International Conference on Parallel Architectures and Compilation Techniques (PACT'10), pp.477-488, 2010.
- 14) S. Bell, et. al.: TILE64 - Processor: A 64-Core SoC with Mesh Interconnect, Proc. of IEEE International Solid State Circuits Conference (ISSCC2008), 2008.