

メディアアプリケーションを用いた並列化コンパイラ協調型 ヘテロジニアスマルチコアアーキテクチャのシミュレーション評価

神山 輝壮† 和田 康孝† 林 明宏† 間瀬 正啓† 中野 啓史†
渡辺 岳志† 木村 啓二† 笠原 博徳†

† 早稲田大学 理工学術院 基幹理工学部 情報理工学科 〒169-8555 東京都新宿区大久保 3-4-1

E-mail: †{kamiyama,yasutaka,akihiro,mase,hnakano,watanabe,kimura,kasahara}@oscar.elec.waseda.ac.jp

あらまし 本稿では、汎用プロセッサコアに加え複数のアクセラレータを1チップ上に集積したヘテロジニアスマルチコアアーキテクチャと、それに協調する自動並列化コンパイラの性能について述べる。コンパイラによる並列性の抽出を考慮して記述されたマルチメディアアプリケーションを用いて、汎用CPUコアを2基、FE-GAを想定したアクセラレータコアを2基搭載したヘテロジニアスマルチコアアーキテクチャ構成で評価したところ、MP3エンコーダでは1つの汎用CPUコアに対して9.82倍、JPEG2000エンコーダでは14.64倍の速度向上率が得られた。

キーワード ヘテロジニアスマルチコア OSCAR コンパイラ

Performance Evaluation of Parallelizing Compiler Cooperated Heterogeneous Multicore Architecture Using Media Applications

Teruo KAMIYAMA†, Yasutaka WADA†, Akihiro HAYASHI†, Masayoshi MASE†,
Hirohumi NAKANO†, Takeshi WATANABE†, Keiji KIMURA†, and Hironori KASAHARA†

† Department of Computer Science and Engineering, Waseda University Okubo 3-4-1, Shinjuku-ku, Tokyo,
169-8555 Japan

E-mail: †{kamiyama,yasutaka,akihiro,mase,hnakano,watanabe,kimura,kasahara}@oscar.elec.waseda.ac.jp

Abstract This paper describes a heterogeneous multicore architecture having accelerator cores in addition to general purpose cores, an automatic parallelizing compiler that cooperatively works with the heterogeneous multicore, a heterogeneous multicore architecture simulation environment, and performance evaluation results with the simulation environment. For the performance evaluation, multimedia applications written in C or Fortran, considered with parallelization by the compiler, are used. As a result, the evaluated heterogeneous multicore having two general purpose cores and two accelerator cores achieves 9.82 times speedup from MP3 encoder. This architecture also achieves 14.64 times speedup from JPEG2000 encoder.

Key words Heterogeneous multicore, OSCAR Compiler

1. はじめに

汎用CPUのみでは得られない高い演算性能を達成するために、ハイパフォーマンスコンピューティングの分野から組み込みシステムの分野まで、ヘテロジニアスマルチコア構成のシステムが大きな注目を集めている。特に、携帯電話やゲーム機、DVDレコーダ、デジタルテレビ、カーナビ等といった情報家電分野においては、高い処理性能を高実装密度、低消費電力で実現可能なヘテロジニアスマルチコアアーキテクチャに対する期待が

大きい[1],[2]しかしながら、各コアが得意とする処理を考慮しながらプログラムを分割して複数のコアに処理を割り当てたり、コア間あるいはメインメモリとのデータ転送オーバーヘッドを最小化するプログラミングは容易ではない。そのため、プログラムの持つ並列性を自動的に解析し、ヘテロジニアスマルチコア用に最適化されたコードを生成する自動並列化コンパイラの開発が必要不可欠となる。

このような問題を解決するために、従来からホモジニアスマルチプロセッサあるいはマルチコアプロセッサ用自動並列化

コンパイラとして、OSCAR マルチグレイン自動並列化コンパイラが開発されている [3], [4]。さらに、OSCAR コンパイラにおけるヘテロジニアスマルチコア用の並列化手法、ならびに OSCAR コンパイラと協調動作する OSCAR ヘテロジニアスマルチコアアーキテクチャの提案も行われている [8]。

本稿では、コンパイラによる並列化を考慮して記述された C [13] あるいは Fortran によるマルチメディアアプリケーションに対して OSCAR コンパイラによる並列化を適用し、OSCAR ヘテロジニアスマルチコアアーキテクチャで評価した結果について述べる。本評価は、従来からのサイクルレベルの詳細なマルチコアアーキテクチャシミュレータに対して、ヘテロジニアスマルチコアアーキテクチャの概略シミュレーションを容易かつ高速に行うことが可能な拡張を施した評価環境により行った。

本稿の構成は以下の通りとなる。第 2 章では本論文が対象とする OSCAR ヘテロジニアスマルチコアアーキテクチャの概略を、第 3 章ではヘテロジニアスマルチコア用の並列化を行う OSCAR コンパイラの概略をそれぞれ説明する。第 4 章では評価対象のアプリケーションと評価環境について述べ、第 5 章では評価結果について述べる。最後に第 6 章で本稿のまとめを述べる。

2. ヘテロジニアスマルチコアアーキテクチャ

本章では本稿が対象とする OSCAR ヘテロジニアスマルチコアアーキテクチャの概略について述べる。OSCAR ヘテロジニアスマルチコアアーキテクチャとは、OSCAR 型メモリアーキテクチャ [3], [4] に対して、動的再構成可能プロセッサ (Dynamically Reconfigurable Processor: DRP) や DSP 等のアクセラレータを付加したヘテロジニアスマルチコア構成のマルチコアアーキテクチャである。

2.1 プロセッサの構成

OSCAR ヘテロジニアスマルチコアプロセッサはローカルメモリやデータ転送ユニットと接続されている CPU や DRP 等のアクセラレータコアからなるプロセッサエレメント (PE) をチップ内外の集中共有メモリ (CSM) と複数バスやクロスバ等の結合網で接続したものである。CSM は各 PE で共有するデータを保持するメモリである。OSCAR ヘテロジニアスマルチコアアーキテクチャのブロック図を図 1 に示す。

2.2 プロセッサエレメント (PE) の構成

各プロセッサエレメントは汎用コアとローカルプログラムメモリ (LPM) やローカルデータメモリ (LDM)、分散共有メモリ (DSM) といった各種のメモリ、データ転送ユニット (DTU) で構成されている。LPM は各 PE が実行するプログラムコードを格納するメモリ、LDM は各 PE のプライベートなデータを格納するメモリである。一方 DSM は自 PE から他 PE から同時にアクセス可能なデュアルポートメモリであり、タスク間データ転送や同期フラグの送受信時に使用される。DTU は高機能 DMAC であり、各プロセッサ処理とは独立して PE 間あるいは PE-CSM 間データ転送を行うことができる。これにより各々のタスク処理とデータ転送のオーバーラップを可能とする。

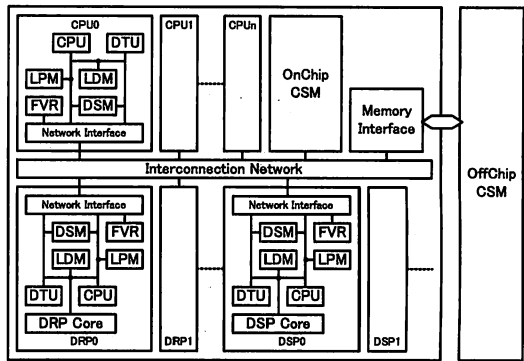


図 1 OSCAR ヘテロジニアスマルチコアプロセッサ

2.3 アクセラレータプロセッサエレメント

アクセラレータプロセッサエレメントとは図 1 中の DRP0 か DSP0 のように、アクセラレータが付加されている PE である。またアクセラレータ PE は動的再構成可能プロセッサの機能書き換えやタスクのスケジューリング情報の処理、アクセラレータコアの起動処理といった一定の計算処理が可能な簡素なコントローラ CPU を持つ。

3. ヘテロジニアスマルチコアマルチコア上の階層的粗粒度タスク並列処理

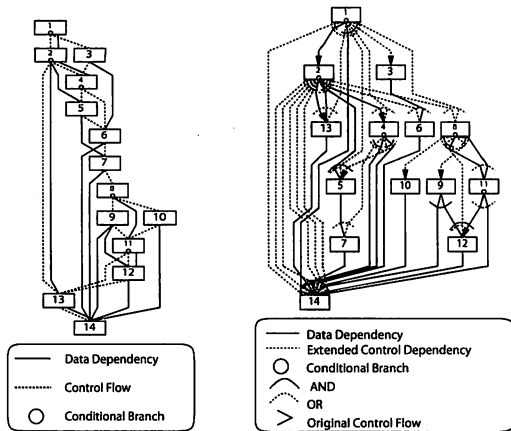
本章ではヘテロジニアスマルチコアプロセッサ上での階層的粗粒度タスク並列処理の概要を述べる。

3.1 粗粒度タスク並列処理

粗粒度タスク並列処理手法では、まずソースとなるプログラムを擬似代入文ブロック (BPA)、繰り返しブロック (RB) およびサブルーチンブロック (SB) の 3 種類のマクロタスク (MT) に分割する。ここで擬似代入文ブロックとは基本ブロック、および複数の小基本ブロックを融合したブロック、または 1 つの基本ブロックが分割されてできるブロックとして定義したものである。コンパイラはこれら 3 種類の MT 間のコントロールフローとデータ依存関係を解析し、その関係を表すマクロフローグラフ (MFG) を生成する。さらにこの MFG から MT 間の並列性を抽出するために、最早実行可能条件解析を実行してマクロタスクグラフ (MTG) を作成する [5], [6]。マクロフローグラフとマクロタスクグラフの例を図 2 に示す。RB または SB の内部ボディ部に粗粒度並列性が存在する場合は、その RB や SB 内部で階層的に MTG の生成を行う [7]。その後、各階層の MT は 1 つ以上のプロセッサエレメント (PE) をグループ化したプロセッサグループ (PG) に対して割り当てられる。この時に MTG 内に条件分岐等がなければ、プロセッサ間の同期やデータ転送などのオーバーヘッドを最小化するために静的に MT を割り当てるスタティックスケジューリング手法 [8] が適用される。

3.2 アクセラレータコアを考慮したプロセッサグルーピング

本稿で対象とするヘテロジニアスマルチコアにおいては、汎用コア PE は全ての種類の MT を実行することが可能である



(a) Macro Flow Graph (MFG)

(b) Macro Task Graph (MTG)

図2 マクロフローグラフ (MFG) とマクロタスクグラフ (MTG) の例

が、アクセラレータコア PE は実行可能な MT が限られている。すなわち、アクセラレータコア PE のプロセッサグループと汎用コア PE のプロセッサグループは分けて考えることが必要である。汎用コア PE と同様にアクセラレータ PE もプログラムの階層構造に沿ってグルーピングすると、粗粒度タスク並列処理を行う MTG のネストが深くなるに従ってアクセラレータの数が足りなくなってしまうことや、各 MTG がアクセラレータを独占してしまい、全体のアクセラレータ利用率が下がってしまうことが考えられる。そのためアクセラレータ PE は階層的なグルーピングからは独立して扱い、必要に応じてあらゆる階層の MTG から MT が割り当てられるものとする。図3にヘテロジニアスマルチコアを考慮した階層的なプロセッサグルーピングの例を示す。図3では、汎用コアが4基、DRPもしくはDSPを2基もつヘテロジニアスマルチコアの例であるが、汎用コアはプログラムの第1階層でそれぞれ2PEからなる2つのPG(PG1.0, PG1.1)にグルーピングされており、各PGにこの階層のMTG内のMTを割り当てて実行することで粗粒度タスク並列処理が行われている。またPG1.0内に粗粒度並列処理が可能な場合、PG1.0はさらに第2階層で1プロセッサをもつ2つのPG(PG1.2.0, PG1.2.1)にグルーピングできる。一方DRPおよびDSPといったアクセラレータPEは汎用コアの階層グルーピングとは違い、実行可能なMTがどの階層のMTGにどのように分散しているかに関わらず、全汎用コアグループからタスク処理依頼を受け付け可能である。

OSCARコンパイラによるヘテロジニアスマルチコア用マクロタスクスケジューリングは、このようにグルーピングされたプロセッサに対し割り当てられる。この時、アクセラレータ用のマクロタスクは、アクセラレータに対して優先的に割り当てられるが、全体の処理時間を考慮して、アクセラレータが混み合っていると必要に応じて汎用コアにも割り当てられることがある。また、データ転送もなるべくマクロタスク処理の裏側で行うためのスケジューリングが行われる。

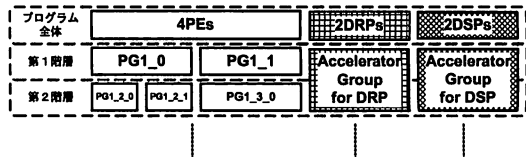


図3 ヘテロジニアスマルチコアを考慮した階層的なプロセッサグルーピングの例

表1 各メモリのアクセスコスト

分散共有メモリ (2Port)	1clock
分散共有メモリ (Remote)	4clock
ローカルメモリ	1clock
集中共有メモリ	16clock

4. 性能評価環境

本章では、ヘテロジニアスマルチコアの概略シミュレーション環境および評価に使用したアプリケーションプログラムについて述べる。

4.1 評価条件

本評価では1チップ上に汎用コアPEまたはアクセラレータPEを合計4つまで搭載するシステムを想定し、ヘテロジニアスマルチコアプロセッサをシミュレートするアーキテクチャシミュレータを用いて評価した。本評価では、従来からのクロックサイクルレベルの詳細なマルチコアアーキテクチャシミュレータに対して、ヘテロジニアスマルチコアアーキテクチャの概略シミュレーションを容易かつ高速に行う拡張を行ったシミュレータ上で評価を行った。本シミュレータではアクセラレータコアの機能を汎用コアの機能レベルシミュレータにより実現している。またアクセラレータコアによる実行クロック数は、アクセラレータで実行部分に対する実行クロック数をあらかじめ与えておくことで計測する。このようにすることで、ホモジニアスマルチコア用実行オブジェクトに対して、アクセラレータによる実行コストのヒント情報を付加するだけで、容易にヘテロジニアスマルチコアの概略シミュレーションを行うことができる。本方式は同等のシミュレーションを周波数ヘテロ構成の詳細シミュレーションにより行う場合に比べて2倍以上高速である。PE間結合網は3本バスとし、CSMの構成は4バンク構成とした。各種メモリのアクセスコストは表1に示す通り、自PE内部の分散共有メモリおよびローカルメモリへのアクセスは1clock、集中共有メモリへのアクセスは16clockとしている。

シミュレーションするアクセラレータコアのモデルとして動的再構成可能プロセッサである株式会社日立製作所のFE-GA [9]を想定している。FE-GAの扱うことのできるMTに含まれる処理のうち、FE-GAにより速度向上が得られる処理に関しては、CPUコアと比べて10倍の速度向上ができると想定し [11]、シミュレーションを行った。また本評価ではCPUコアとしてSPARC V9命令セットを持つシングルイシューの簡素なプロセッサコアを使用している。本評価に汎用コアとしてSPARCコアを用いたのは、前述のシミュレータのヘテロジニアスマル

チコア対応を SPARC コアを優先して行ったためである。

4.2 評価アプリケーション

本評価では MP3 エンコーダと JPEG2000 エンコーダを用いた。MP3 エンコーダはコンパイラによる自動並列化を考慮した制約つき C に記述されている。

4.2.1 自動並列化のための制約つき C 言語の使用

情報家電分野では主として C 言語がソフトウェア開発に使用されるが、一般的に C 言語はポインタなどの言語仕様から自動並列化することは困難である。そのため、本稿では並列化コンパイラの並列化性能を C においても最大限発揮できるよう、MP3 エンコーダの記述に以下の制約を課した [13]。

- 関数のポインタ引数以外のポインタと構造体を使用しない
- 複数のポインタ引数を用いた参照先が重ならない
- 関数の再帰呼び出しを使用しない

4.2.2 MP3 エンコーダ

本稿では UZURA MPEG1 / LayerIII encoder in FORTRAN90 [10] を参照実装した MP3 エンコーダを用いる。ただし通常オプションとして与えられるパラメータは定数として表記した。図 4 (a) に示す通り、MP3 エンコーディングはサブバンド解析、MDCT (変形離散コサイン変換)、心理聴覚分析、非線形量子化、ハフマン符号化、ビットストリーム生成の 5 つのステージから成り立つ。各ステージは一度に入力フレーム数分処理を行い、これらの処理を各々の入力データに対して順に適用する。これら 5 つのステージと初期化の一部のループにループ整合分割 (LAD) [12] を適用し、フレーム間の粗粒度タスク並列性を抽出した。そのイメージを図 4 の (b) に示す。図の例ではループを 4 つの粗粒度タスクに分割して並列性を抽出している。ただし MDCT 部と心理聴覚分析部は前フレームの演算結果を使用するのでフレーム間の依存が存在する。しかし、それ以外の箇所では各フレームに対する処理を並列に行うことができる。本評価では PCM データ 32 フレーム分を入力データとして使用し、ループ整合分割で 32 分割し、並列実行した。FE-GA で実行可能な MT に関しては、プログラムソース上で指示文を用いてサブバンド分析の一部、心理聴覚分析、MDCT、非線形量子化の一部を指定した。

4.2.3 JPEG2000 エンコーダ

JPEG2000 エンコードの処理は図 5 に示す通り、DC レベル変換、離散ウェーブレット変換、スカラ量子化、EBCOT(Embedded Block Coding with Optimized Truncation) 符号化の 4 ステージからなる。本評価では EBCOT 符号化において、64x64 ピクセルのコードブロック単位の並列性を抽出し並列実行した。またスカラ量子化は各サブバンドに対して行う処理であるが、コードブロックがサブバンド境界を超えて生成されないため、コードブロック単位で処理を行うことが可能であり並列性を抽出できる。以上 4 ステージ全てを FE-GA で実行可能であると想定し、評価を行った。

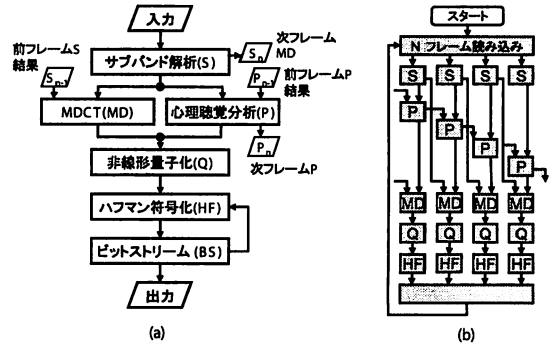


図 4 MP3 エンコーダの処理フローとフレーム間並列性利用を考慮したプログラム構造

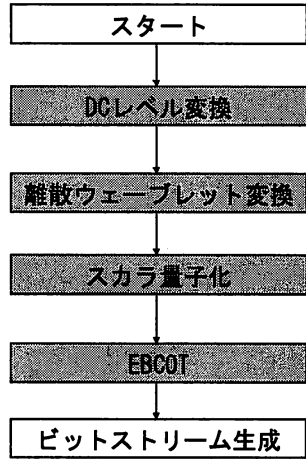


図 5 JPEG2000 エンコーダの処理フロー

5. 評価結果

5.1 コア構成を変化させての性能評価結果

図 6 と図 7 に MP3 エンコーダと JPEG2000 エンコーダを用いたときの評価結果をそれぞれ示す。図の横軸はプロセッサのコア構成であり、図中の n CPU + m DRP とは n 基の汎用コア PE と m 基のアクセラレータ PE によるアーキテクチャ構成を意味する。また縦軸には 1CPU による逐次実行時に対する速度向上率が示されている。図 6 が示すように MP3 エンコーダでは 1CPU+1DRP 構成において 1CPU に比べ 4.72 倍、2CPU+1DRP 構成では 5.57 倍、2CPU+2DRP では 9.82 倍の速度向上率を得ることができた。4CPU 構成に比べ 2CPU+2DRP 構成は 2.46 倍速度向上し、性能の点でホモジニアスマルチコアと比べて、ヘテロジニアスマルチコアアーキテクチャの優位性が示された。

図 7 に JPEG2000 エンコーダを用いたときの評価結果を示す。図 6 と同様に横軸はプロセッサのコア構成であり、縦軸は 1CPU による逐次実行時に対する速度向上率が示されている。同様に図 7 の JPEG2000 エンコーダでは、1CPU+1DRP 構成では 1CPU に比べ 9.38 倍、2CPU+1DRP 構成では 8.61 倍、

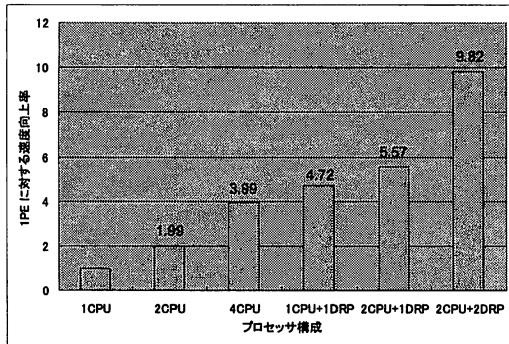


図 6 MP3 エンコーダの評価結果

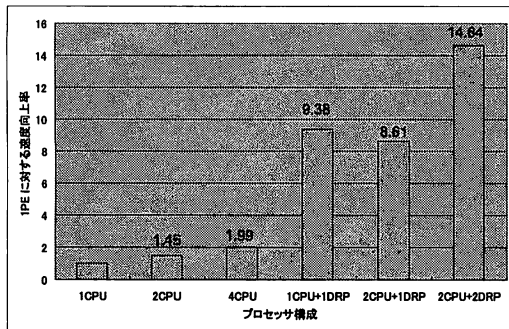


図 7 JPEG2000 エンコーダの評価結果

2CPU+2DRP 構成では 14.64 倍の速度向上率を得ることができた。2CPU 構成と比べて 1CPU+1DRP 構成は 6.47 倍速度が向上し、4CPU 構成と比べて 2CPU+2DRP 構成で 7.34 倍の速度向上が得られた。JPEG2000 エンコーダにおいてもヘテロジニアスマルチコアの優位性が示された。

5.2 評価結果の考察

本章では各アプリケーションを用いたときの評価結果について考察を述べる。

5.2.1 MP3 エンコーダ

図 8 に MP3 エンコーダに 2CPU+2DRP 構成時の実行トレース結果を示す。図中の CPU0, CPU1 は汎用コア PE を表し、DRP0, DRP1 はアクセラレータ PE を表す。MT の右側の数字は各 MT の番号が記されている。この図に載っている枠はサブバンド解析処理と非線形量子化処理を行う MT を示す。サブバンド解析処理の MT は CPU が DRP に比べて 10 倍実行クロック数がかかることを考慮して、バランスよく CPU と DRP にタスクが割り当てられている。一方、非線形量子化処理は、サブバンド解析処理ほど CPU と DRP にバランスよく割り当てられていない。これはスケジューリング時には CPU は DRP に対し 10 倍実行クロックがかかることを考慮して計算しているが、実際には非線形量子化部にはサブバンド解析処理に比べて DRP 内のアクセラレータコア以外にコントローラ CPU で処理を行うことが多いため、DRP に割り当てられたタスクの実行時間がスケジューリングで想定された時間より長く

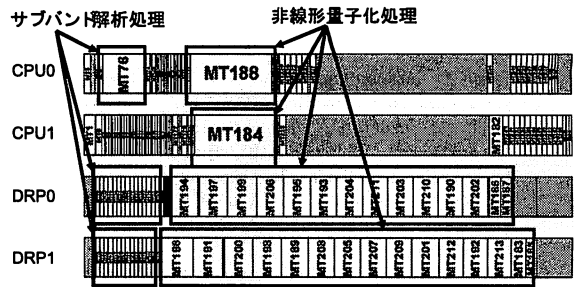


図 8 2CPU+2DRP 構成時の MP3 エンコーダ実行トレース結果

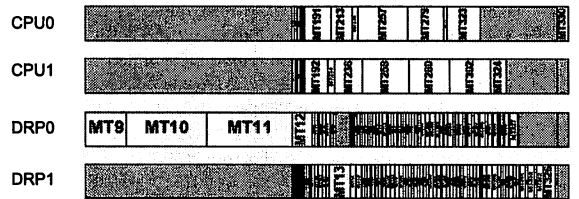


図 9 2CPU+2DRP 構成時の JPEG2000 エンコーダ実行トレース結果

なる。これによって CPU 側のアイドル時間が長く発生することとなった。

5.2.2 JPEG2000 エンコーダ

図 9 に 2CPU+2DRP 構成時の JPEG2000 エンコーダの実行トレース結果を示す。図中の MT9 にあたる処理は DC レベル変換で、その後に行われる MT10, MT11 は離散ウェーブレット変換処理にあたる。これらの MT は逐次処理されるため、DRP が 1 基のみ使用され、その他の PE はアイドル状態となっている。その後のスカラ量子化、EBCOT 符号化処理は、並列に実行されるため、各 CPU と DRP にバランスよく割り当てられていることが示されている。離散ウェーブレット変換で並列性が抽出できたとしても、データ転送オーバーヘッドの大きい処理であり、本結果はアプリケーションの性質を最大限に活かすことができたと考えられる。

6. まとめ

本稿では汎用 CPU に加え、複数のアクセラレータを 1 チップ上に集積したヘテロジニアスマルチコアアーキテクチャとそれに協調する自動並列化コンパイラの性能評価を行った。ヘテロジニアスマルチコアの概略シミュレーション環境を構築し、コンパイラによる並列性の抽出を考慮して記述したマルチメディアアプリケーションを用いて評価を行った。その結果汎用コア PE を 2 基、FE-GA を想定したアクセラレータ PE を 2 基搭載したヘテロジニアスマルチコア構成において、MP3 エンコーダの場合 1 つの汎用 CPU コアに対して 9.82 倍、JPEG2000 エンコーダの場合では 14.64 倍の速度向上が得られた。

謝辞 本研究の一部は NEDO「リアルタイム情報家電用マルチコア技術」、「情報家電用ヘテロジニアスマルチコア」プロジェクト、早稲田大学グローバル COE「アンビエント SoC」の

支援により行なわれた。

文 献

- [1] Pham, D., Asano, S. and et al., M.B.: The Design and Implementation of a First- Generation CELL Processor (2005).
- [2] Torii, S., Suzuki, S., Tomonaga, H., Tokue, T., Sakai, J., Suzuki, N., Murakami, K., Hiraga, T., Shigemoto, K., Tatebe, Y., Obuchi, E., Kayama, N., Edohiro, M., Kusano, T. and Nishi, N.: A 600MIPS 120mW 70mA Leakage Triple-CPU Mobile Application Processor Chip, ISSCC(2005)
- [3] 笠原、成田、橋本: “OSCAR(Optinally Scheduled Advanced Multiprocessor) のアーキテクチャ”, 電子情報通信学会論文誌, D 分冊, Vol.J71-D, No.8(1988).
- [4] Kimura, K., Wada, Y., Nakano, H., Kodaka, T., Shirako, J., Ishizaka, K. and Kasahara, H.: Multigrain Parallel Processing on Compiler Cooperative Chip Multiprocessor, Proc. of 9th Workshop on Interaction between Compilers and Computer Architectures (INTERACT-9) (2005).
- [5] 本田, 岩田, 笠原: Fortran プログラム粗粒度タスク間の並列性検出法, 信学論 (D-1), Vol.J73-D-I, No.12, pp.951-960(1990).
- [6] 笠原, 合田, 吉田, 岡本, 本多: Fortran マクロデータフロー処理のマクロタスク生成手法, 信学論, Vol.J75-D-I, No.8, pp.511-525(1992). Vol.J73-D-I, No.12, pp.951-960(1990).
- [7] 白子, 神長, 近藤, 小幡, 笠原: 並列処理階層自動決定手法を用いた粗粒度タスク並列処理, Arc2002-148-4, 情報処理学会 (2002)
- [8] 和田, 押山, 鈴木, 内藤, 白子, 中野, 鹿野, 木村, 笠原: ヘテロジニアスチップマルチプロセッサにおける粗粒度タスクスタティックスケジューリング手法, Shining2006, 情報処理学会 (2002).
- [9] 津野田, 高田, 秋田, 田中, 佐藤, 伊藤: デジタルメディア向け再構成型プロセッサ FE-GA の概要, 信学技報 RECONF2005-65(2005).
- [10] UZURA3: MPEG1/LayerIII encoder in FORTRAN90, http://members.at.infoseek.co.jp/kitaurawa/index_e.html.
- [11] 田中 他: 再構成プロセッサ FE-GA のオーディオ処理への応用, 信学技報 RECONF2005-67.
- [12] 吉田, 越塚, 岡本, 笠原: 階層型粗粒度並列処理における同一階層内ループ間データローカライゼーション手法, 情報処理学会論文誌, Vol. 40, No. 5, pp. 2054-2063 (1999).
- [13] 間瀬, 馬場, 長山, 田野, 益浦, 深津, 宮本, 白子, 中野, 木村啓二, 笠原博徳, “OSCAR コンパイラにおける制約付き C プログラムの自動並列化”, 情報処理学会研究会報告 2006-ARC-170-01