

情報家電用マルチコア上における マルチメディア処理のコンパイラによる並列化

宮本孝道[†] 浅香沙織[†] 見神広紀[†]
間瀬正啓[†] 木村啓二[†] 笠原博徳[†]

半導体集積度向上に伴う消費電力の増大，プロセッサ動作クロック周波数向上の鈍化，ハードウェア・ソフトウェア開発期間の増大といった問題に対処すべく，一つのチップ上に複数のプロセッサコアを集積するマルチコアプロセッサが注目を集めている．また，携帯電話，デジタルTV，ゲーム機等の情報家電機器ではコーデック，グラフィックス等のマルチメディア処理の高速化が非常に重要となっている．本論文ではMPEG2エンコード・デコード，MP3エンコード，JPEG 2000エンコードのマルチメディア処理をOSCAR並列化コンパイラにより，新規開発したAPIを用いて並列化し，VLIWコアを4基集積の富士通FR1000マルチコア及びルネサステクノロジ，日立製作所，早稲田大学共同開発SH-4Aコアを4基集積のRP1マルチコア上での並列性能評価を行った．FR1000マルチコアでは4プロセッサ時に1プロセッサ時に比べ上記4アプリケーションの平均で3.27倍，RP1マルチコアでは4プロセッサ時に1プロセッサ時に比べ上記4アプリケーションの平均で3.31倍という並列処理性能が得られた．

Parallelization of Multimedia Applications by Compiler on Multicores for Consumer Electronics

TAKAMICHI MIYAMOTO,[†] SAORI ASAKA,[†] HIROKI MIKAMI,[†]
MASAYOSHI MASE,[†] KEIJI KIMURA[†] and HIRONORI KASAHARA[†]

Multicore processors have attracted much attention because there are much opportunity to overcome the increase of power consumption, the difficulty of improvement of processor clock speed, and the increase of hardware/software developing period. Also, speeding up multimedia applications is required with the progress of consumer electronics like mobile phones, digital TV and games. This paper evaluates parallel processing performances of multimedia applications such as MPEG2 encode and decode, MP3 encode and JPEG 2000 encode using newly developed multicore API by OSCAR parallelizing compiler on the FR1000 4 VLIW cores multicore processor developed by Fujitsu Ltd, and the RP1 4 SH-4A cores multicore processor jointly-developed by Renesas Technology Corp., Hitachi Ltd. and Waseda University. As the results, the OSCAR compiler gave us 3.27 speedup in average using 4 cores on FR1000 multicore, 3.31 speedup in average using 4 cores on RP1 multicore.

1. はじめに

従来，マイクロプロセッサにおける性能向上の牽引力になっていた命令レベル並列性の利用と周波数の向上は，半導体集積度の向上につれて並列性抽出の限界，消費電力の増大等が顕在化し，今後の性能向上への寄与が難しくなっている．これらの問題に対処するためマルチコアプロセッサが注目を集めている．マルチコアプロセッサでは複数のプロセッサコアを一つのチップ上に集積することにより，プロセッサコア間で命令

レベル並列性よりも並列性の大きいループレベル，タスクレベルの並列処理の実現が可能となる．また，各プロセッサコアをコンパイラやOSと協調し低周波数・低電圧で動作させ，適切な並列処理をすることで，高性能・低消費電力が実現可能なアーキテクチャとしても期待されている．特にデスクトップ及びサーバ向けのインテル Quad コア Xeon や Core 2 Quad，AMD の Quad コア Phenom，ワークステーション，ハイエンドサーバ用の IBM Power6，Sun SPARC T2 等が発表されている．また，情報家電向けの組み込み分野においても，SCE/IBM/東芝の Cell¹⁾，NEC エレクトロニクス/ARM の MPCore，MP211²⁾，富士通の FR1000³⁾，パナソニックの UniPhier⁴⁾，ルネサステ

[†] 早稲田大学
Waseda University

クノロジの SH-X3⁵⁾ が発表され、ゲーム機や携帯電話、カーナビ等に利用され始めている。

情報家電機器上では動画や音声等のマルチメディア処理が広く行われ、アルゴリズムの複雑化、扱うデータ量の増大に伴い高い処理性能が要求されてきている。従来、このような要求に応えるためには DSP、マルチメディアプロセッサの利用や Intel SSE 等のマルチメディア拡張命令セットの追加による SIMD 最適化が行われてきた。しかし、今後より複雑な処理に短期間で柔軟に対処するためには組み込み用マルチコアプロセッサ及びこれらの性能を引き出す自動並列化コンパイラ技術が重要になる。

NEDO “リアルタイム情報家電用マルチコア技術の研究開発” プロジェクトで開発するコンパイラ技術を各社マルチコア上で利用できるように本プロジェクトにおける早稲田大学委託事業内マルチコアアーキテクチャ・API 検討委員会（委員会に参加、(株)日立製作所、(株)ルネサステクノロジ、(株)富士通研究所、(株)東芝、(株)松下電器産業、(株)日本電気）で情報家電用マルチコア並列化 API を開発した。このような API 開発は、The Multicore ASSOCIATION⁶⁾ でも行われており、これは主に分散メモリ環境を想定した通信・デバッグ用の API となっている。これに対し今回開発された API はローカルメモリ、分散共有メモリ、オンチップ集中共有メモリ等の複数種類のメモリを持つマルチコアアーキテクチャを対象とし、リアルタイム性、低消費電力制御も可能とする新しいタイプの API となっている。この API を用いることによって、各社のプロセッサコアが異なるマルチコア上でのコンパイラを用いた並列処理が実現可能となる。

本論文ではマルチメディアアプリケーションを対象として OSCAR 並列化コンパイラにより上記の API 入りの並列化コードを生成した。これらの並列化されたプログラムを富士通 FR1000 マルチコアとルネサステクノロジ、日立製作所、早稲田大学共同開発 RP1 マルチコアという二種類の異なるマルチコアにおいて並列性能評価を行う。

以下本論文では、第 2 章で対象とするマルチコアアーキテクチャについて、第 3 章で情報家電用マルチコア並列化 API と本 API を用いた各社マルチコア上での評価手順について、第 4 章で OSCAR 並列化コンパイラで扱う粗粒度タスク並列処理手法について、第 5 章でマルチメディアアプリケーションの富士通 FR1000 マルチコア、ルネサス 日立 早大 RP1 マルチコア上での性能評価結果について、第 6 章では本論文のまとめを述べる。

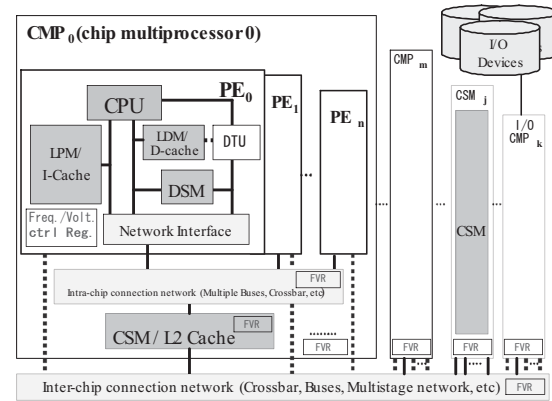


図 1 OSCAR マルチコアアーキテクチャ
Fig. 1 OSCAR Multicore Architecture

2. 対象マルチコアアーキテクチャ

NEDO “リアルタイム情報家電用マルチコア技術の研究開発” プロジェクトのマルチコアアーキテクチャ・API 検討委員会において、既存マルチコアである FR1000, MP211, CELL, UniPhier のメモリアーキテクチャを検討し、API を定める対象としてこれらの既存メモリアーキテクチャを包含するアーキテクチャである OSCAR マルチコアアーキテクチャが採択された。本章ではこの OSCAR マルチコアアーキテクチャと共に、本論文での並列性能評価に用いた富士通 FR1000 マルチコア、及び OSCAR マルチコアアーキテクチャに準拠して上記プロジェクトで試作されたルネサステクノロジ、日立製作所、早稲田大学共同開発 RP1 マルチコアについて述べる。

2.1 OSCAR マルチコアアーキテクチャ

OSCAR マルチコアアーキテクチャは OSCAR 並列化コンパイラとの協調動作により、高実効性能・低消費電力化の実現を目指したアーキテクチャである^{7),8)}。

図 1 に示すように OSCAR マルチコアアーキテクチャは 1 つのチップ上に複数のプロセッサエレメント (PE) を持ち、各 PE はプロセッサプライベートなデータを保持するローカルデータメモリ (LDM)、共有データや同期変数を保持する分散共有メモリ (DSM)、プログラムコードを保持するローカルプログラムメモリ (LPM)、及び CPU と非同期にデータ転送が可能なデータ転送ユニット (DTU) を持つ。チップ上の全ての PE はバスやクロスバ等の Interconnection Network によってオンチップあるいはオフチップ集中共有メモリ (CSM) に接続されるメモリ構成となっている。

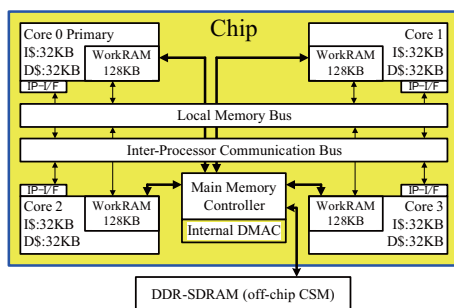


図 2 FR1000 マルチコアブロック図概要

Fig. 2 Block Diagram of FR1000

2.2 富士通 FR1000 マルチコア

FR1000 マルチコアは図 2 に示すように、8way VLIW の FR550 プロセッサコアを 4 基集積のホモジニアスマルチコアであり、各コアは 32KB のインコヒーレントなデータキャッシュ、32KB の命令キャッシュ及び OSCAR マルチコアアーキテクチャの DSM にあたる 128KB の WorkRAM を持ち、チップ上には 2 チャンネルのメインメモリコントローラ、DTU として DMAC が存在する。また、off-chip CSM は最大 2 バンク構成で利用でき、今回用いた環境では 1GB × 2 バンクの構成を取っている。

2.3 ルネサステクノロジ 日立製作所 早稲田大学 共同開発 RP1 マルチコア

RP1 マルチコアは図 3 に示すように、SH-4A (SH-X3) コアを 4 基集積のホモジニアスマルチコアであり、各コアは 32KB のデータキャッシュ、32KB の命令キャッシュ及び OSCAR マルチコアアーキテクチャの LDM にあたる 16KB の OLRAM, LPM にあたる 8KB の ILRAM, DSM にあたる 128KB の URAM をそれぞれ持ち、チップには on-chip CSM にあたる 128KB の CSM が存在する。従来の共有メモリモデルによるプログラミングが容易な SMP モード、リアルタイム制約が保証しやすい AMP モード、及びそのハイブリッドモードでの利用が可能である。SMP モードではスヌープコントローラが専用のスヌープバスを介して各コアのデータキャッシュの一貫性を保証する。

3. 情報家電用マルチコア並列化 API を用いたマルチコア上での性能評価手順

本章では、FR1000 マルチコアと RP1 マルチコアのような異なるマルチコア上で、プロジェクトで開発された並列化コンパイラ技術による並列化を可能とするために開発された情報家電用マルチコア並列化 API とそれを用いたマルチコアの性能評価手順について述べる。

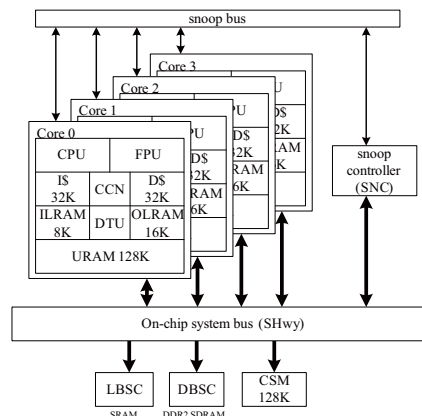


図 3 RP1 マルチコアブロック図概要

Fig. 3 Block Diagram of RP1

情報家電用マルチコア並列化 API は OpenMP API⁹⁾ 互換形式の API (C と FORTRAN に対応) であり、SMP 上では OpenMP コンパイラでそのままコンパイルし動作することが可能である。本 API は、並列処理、メモリオペレーション順序保証に加え、リアルタイム処理に利用するローカルメモリ、分散共有メモリ、オンチップ集中共有メモリ等のメモリ配置を指定することができる。さらに本 API では、データ転送ユニット、消費電力制御にも対応している。

情報家電用マルチコア並列化 API を用いた各社マルチコアの性能評価手順を図 4 に示す。図 4 に示すように、入力として逐次の C あるいは FORTRAN プログラムを用い、OSCAR 並列化コンパイラにより各種マルチコア固有のメモリアーキテクチャやメモリパラメータを考慮したデータ分割、マルチプロセッサスケジューリング、適切なメモリ上へのデータ配置等が施され、API が挿入された C あるいは FORTRAN の並列化プログラムが出力される。出力された本 API 付き並列化プログラムは各社の逐次コンパイラの前に API 解釈系をつけるだけの簡単な作業により、各社マルチコア用の実行バイナリ生成と、各社マルチコアの性能評価が可能となる。

4. OSCAR 並列化コンパイラによる粗粒度タスク並列処理

本章では OSCAR 並列化コンパイラによる粗粒度タスク並列性の抽出、データローライゼーション手法、粗粒度タスクスケジューリング手法について述べる。

4.1 粗粒度タスク並列性抽出

粗粒度タスク並列処理では、まずソースプログラムを基本ブロックあるいは基本ブロックを融合・分割した形である疑似代入文ブロック (BPA)、ループの一

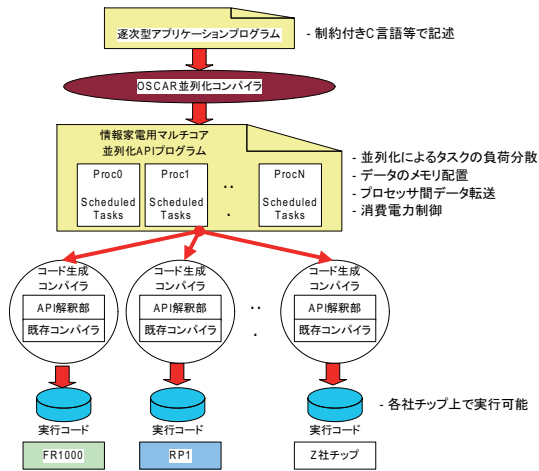


図4 情報家電用マルチコア並列化 API を用いた各社マルチコア評価手順

Fig. 4 Evaluation Flow on Multicores using Developed API

般形である繰り返しブロック (RB), 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。ループ並列処理不可能な実行時間の大きい RB や、インライン展開を効果的に適用できない SB に対しては、その内部を階層的に粗粒度タスクに分割して並列処理を行う。

MT の生成後、MT 間のコントロールフローとデータ依存を解析し、図 5 (a) に示すようなマクロフローグラフ (MFG) を生成する。図 5 (a) の各ノードは MT を表し、実線エッジはデータ依存を、点線エッジはコントロールフローを表す。また、ノード内の小円は条件分岐を表す。

MFG 生成後、コンパイラは並列性を抽出するためにコントロールフローとデータ依存の両方を考慮した最早実行可能条件解析を MFG に対して行う。MT の最早実行可能条件とは、コントロール依存とデータ依存を考慮したその MT が最も早い時点で実行可能になる条件である。MT の最早実行可能条件は図 5 (b) に示すようなマクロタスクグラフ (MTG) で表される。MFG, MTG 共にエッジの矢印は省略されているが、下向きが想定されている。MTG では横に並んでいる MT 間の並列性が表現されている。

4.2 データローカライゼーション手法

データローカライゼーション手法とは、アクセス速度の異なる複数のメモリ機構を持つアーキテクチャにおいて、プロセッサ近傍の高速なメモリを有効に利用して、処理速度の向上を図るものである。データローカライゼーション手法は従来の第一ループネスト内のブロッキングを用いたキャッシュ最適化と比較して複数

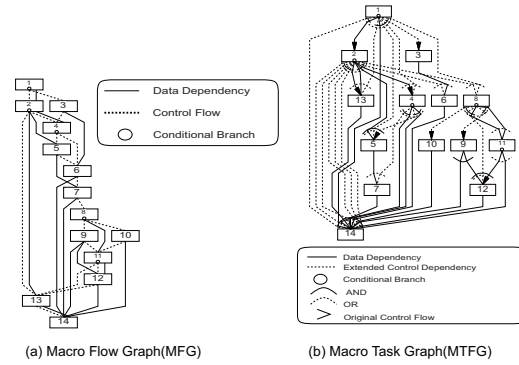


図5 マクロフローグラフとマクロタスクグラフ

Fig. 5 Macro flow graph and Macro-task graph

のループネスト集合に対するキャッシュあるいはローカルメモリ最適化技術である。データ依存を持つ複数のループをそれらの使用データサイズがキャッシュあるいはローカルメモリサイズにおさまるように、複数ループを整合して分割をする¹⁰⁾。分割後のデータローカライズブルグループ (DLG) を同一プロセッサで連続実行することでキャッシュあるいはローカルメモリを介したデータの授受により複数ループネスト間でのデータ転送の最小化が可能となる。

4.3 粗粒度タスクスケジューリング手法

粗粒度タスク並列処理では、複数のプロセッサエレメント (PE) から構成されるプロセッサグループ (PG) に割り当てられて実行される。本論文では MTG がデータ依存エッジのみを持つ場合にコンパイラがスタティックに MT の PG への割り当てを決定することで実行時スケジューリングオーバーヘッドを無くし、データ転送と同期のオーバーヘッドを最小化することが可能であるスタティックスケジューリングを用いた。本論文では ETF/CP (Earliest Task First/Critical Path) 法に対して DLG に属する MT を同一の PG への連続割り当てを優先したマルチプロセッサスケジューリングアルゴリズム ETF/CP considering DLG 法を適用した。本論文では中粒度、近細粒度の並列性を利用しないので以後 PG を PE として表す。

5. 情報家電用マルチコア並列化 API を用いた FR1000, RP1 上での並列性能評価

本章では OSCAR 並列化コンパイラで並列性の抽出が容易になるためのプログラム記述、本記述を用いたマルチメディアアプリケーションの OSCAR 並列化コンパイラによる並列化、富士通 FR1000 マルチコア及びルネサス 日立 早大 RP1 マルチコア上での並列性能評価について述べる。

```

void mpeg2decode()
{
  unsigned char arr[16][22][16][16];
  :
  for (i=0; i<16; i++) {
    process_slice(arr[i], ...);
  }
  :
}

void process_slice(unsigned char rarr[22][16][16], ...);

```

図 6 OSCAR 並列化コンパイラ向け並列化ループ記述
Fig. 6 Loop Description for Parallelization by OSCAR Compiler

5.1 OSCAR コンパイラにおける並列化のための C 言語記述

本節では C 言語で記述されたプログラムに対して OSCAR コンパイラによる並列性抽出を容易にするための C 言語記述について述べる。制限事項として、解析が困難であるポインタ・構造体利用の制限、再帰構造の禁止等の制限を設け、また、関数呼び出しを含むループにおいては図 6 に示す形式での記述を施している。図 6 の記述形式では、配列を多次元で宣言し、ループ内部における関数呼び出しの配列引数にある次元をループインデックス変数で指定する。この記述により、データが関数に対して配列次元の単位で渡されるため、ループイタレーション毎の配列アクセス範囲の独立性が解析でき、ループ並列性抽出が容易となる。

5.2 評価対象マルチメディアアプリケーション

本論文ではマルチメディア処理として動画処理における MPEG2 エンコード、MPEG2 デコード、音声処理における MP3 エンコード、静止画処理における JPEG2000 エンコードを用いた。各マルチメディアアプリケーションにおける OSCAR 並列化コンパイラによる並列化について述べる。

5.2.1 MPEG2 エンコード

MPEG2 エンコード処理は、動き推定、動き予測、DCT モード選択、データ変換、ビットストリーム出力、逆量子化、逆データ変換の 7 つのステージからなる。MPEG2 で扱うデータ構造は階層的であり、上位階層からビデオシーケンス、Group of Picture (GOP)、ピクチャ、スライス、マクロブロック、ブロックとなっている。MPEG2 エンコードでは GOP レベル、スライスレベル、マクロブロックレベルの並列性が存在する。各レベルにおいて、ビットストリーム出力ステージにおいてはマクロブロック間のデータ依存が存在するため、マクロブロック毎に逐次で処理を行う必要がある。

図 7 の左に示すピクチャ単位の逐次プログラムの

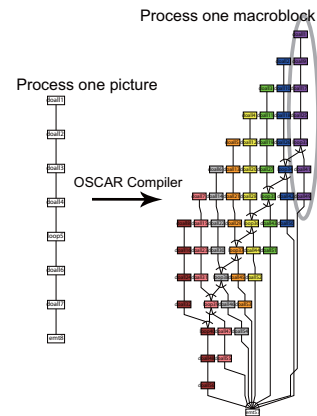


図 7 MPEG2 エンコードの並列化
Fig. 7 Parallelization MPEG2 encode

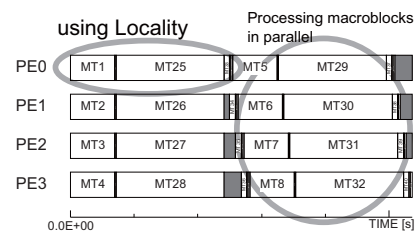


図 8 MPEG2 エンコードの 4PE スケジューリング結果
Fig. 8 Scheduling Result of MPEG2 encode on 4PE

MTG が OSCAR 並列化コンパイラにより右に示すようにマクロブロック単位に分割され、並列性が抽出された MTG となる様子を示す。図 8 に 4PE でのスケジューリング結果を示す。図 8 では分割されたマクロブロック処理が同一 PE で連続実行されることによるデータローカリティ最適化が行われていることを示す。

5.2.2 MPEG2 デコード

MPEG2 デコード処理は、可変長複合化、逆量子化、逆量子化後の各係数値の制限処理、逆離散コサイン変換、動き補償予測、足し合わせ処理の 6 つのステージからなる。MPEG2 デコード処理ではスライスレベルの並列性とスライス処理内部でのマクロブロックレベルの並列性が存在する。本論文では並列性向上のためにスライスに対する可変長復号化処理中のスライスヘッダの検出処理を分割するプレスキャッシング手法を適用した¹¹⁾。プレスキャッシング手法ではビットストリームを先頭から走査するためにスライス毎に逐次で処理を行う必要がある。

図 9 の左に示すピクチャ単位とスライス単位での逐次プログラムの MTG が OSCAR 並列化コンパイラにより右に示すようにピクチャ単位の処理がスライス単位の処理に分割され、内部階層においてはスライス単位の処理がマクロブロック単位の処理に分割され、

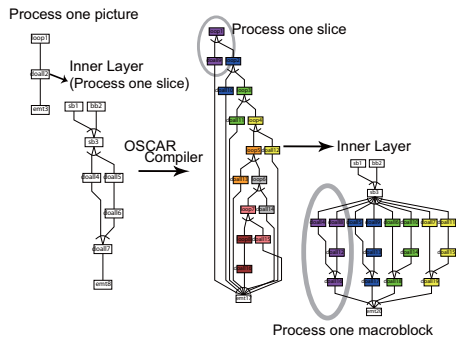


図 9 MPEG2 デコードの並列化

Fig. 9 Parallelization MPEG2 decode

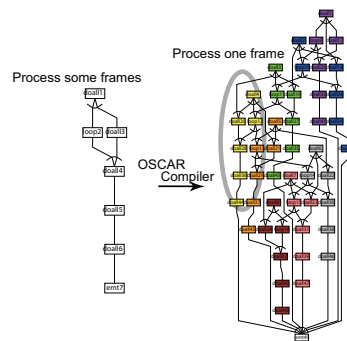


図 11 MP3 エンコードの並列化

Fig. 11 Parallelization MP3 encode

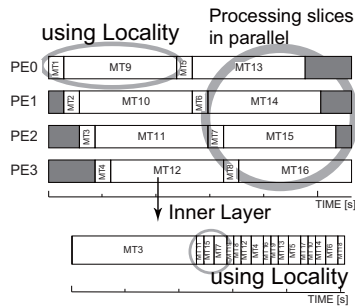


図 10 MPEG2 デコードの 4PE スケジューリング結果

Fig. 10 Scheduling Result of MPEG2 decode on 4PE

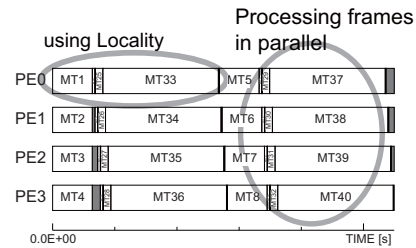


図 12 MP3 エンコードの 4PE スケジューリング結果

Fig. 12 Scheduling Result of MP3 encode on 4PE

並列性が抽出された MTG となる様子を示す．図 10 に 4PE でのスケジューリング結果を示す．図 10 では分割されたスライス処理が同一 PE で連続実行されることによるデータローカリティ最適化が行われ、スライス処理の内部階層では分割されたマクロブロック処理が連続実行されることによるデータローカリティ最適化が行われていることを示す．

5.2.3 MP3 エンコード

MP3 エンコード処理は、サブバンド分析、心理聴覚分析、変形離散コサイン変換、非線形量子化、ハフマン符号化の 5 つのステージからなる．MP3 で扱うデータ構造は階層的であり、上位階層からフレーム、グラニュール、サンプルとなっている．MP3 エンコード処理ではフレームレベルの並列性が存在する．ただし、心理聴覚分析、変形離散コサイン変換においてフレーム間でのデータ依存が存在するため、フレーム毎に逐次で処理を行う必要がある．

図 11 の左に示す複数フレーム単位での逐次プログラムの MTG が OSCAR 並列化コンパイラにより右に示すようなフレーム単位の処理に分割され並列性が抽出された MTG となる様子を示す．図 12 に 4PE でのスケジューリング結果を示す．図 12 では分割さ

れたフレーム処理が同一 PE で連続実行されることによるデータローカリティ最適化が行われていることを示す．

5.2.4 JPEG 2000 エンコード

JPEG 2000 エンコード処理は、DC レベル変換、離散ウェーブレット変換 (DWT)、スカラ量子化、EBCOT (Embedded Block Coding with Optimized Truncation) 符号化の 4 つのステージからなる．JPEG 2000 エンコードで扱うデータ構造として、DWT 後に生成される 2 次元画像中の矩形領域であるサブバンド、サブバンド内を最大 64x64 ピクセルで分割したコードブロックが存在する．JPEG 2000 エンコードでは DC レベル変換、DWT では画像の縦方向、横方向の並列性が存在し、スカラ量子化、EBCOT 符号化ではコードブロック単位の並列性が存在する．ただし、DWT においては縦方向、横方向の処理間では転置転送の形でデータ転送が発生する．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越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となる．

図 13 の左に示すピクチャ全体の逐次プログラムの MTG が OSCAR 並列化コンパイラにより右に示すようにピクチャ全体の処理が DC レベル変換、DWT における縦方向あるいは横方向の処理に分割され、ス

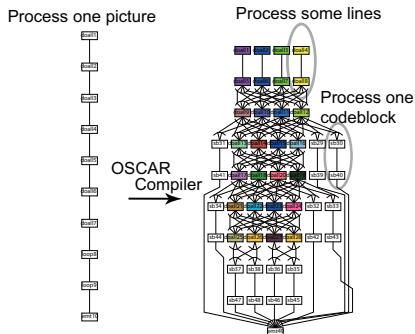


図 13 JPEG 2000 エンコードの並列化
Fig. 13 Parallelization JPEG 2000 encode

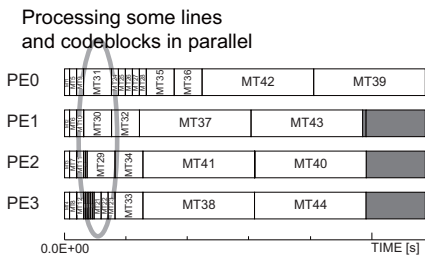


図 14 JPEG 2000 エンコードの 4PE スケジューリング結果
Fig. 14 Scheduling Result of JPEG 2000 encode on 4PE

カラ量子化, EBCOT 符号化においてはコードブロック単位の処理に分割され, 並列性が抽出された MTG となる様子を示す. 図 14 に 4PE でのスケジューリング結果を示す. 図 14 では画像の縦方向あるいは横方向単位の処理とコードブロック単位の処理間での並列処理が行われていることを示す.

5.3 並列性能評価条件

評価には, 第 5.2 節で説明した 4 つのマルチメディアアプリケーションを, MPEG2 エンコード・デコードは MediaBench¹²⁾, MP3 エンコードは UZURA MP3 encoder¹³⁾, JPEG 2000 エンコードは JJ2000¹⁴⁾ を参照実装してそれぞれ使用した. これらのプログラムを第 5.1 節で示した C 言語記述により修正した. ただし, FR1000 マルチコアでは倍精度浮動小数点演算をソフトウェアにより実装しているため, MPEG2 エンコードプログラムでは DCT を Intel アプリケーションノート 922¹⁵⁾ を参照実装したプログラムを用いた. 処理パラメータは参照元プログラムにおけるデフォルトパラメータを使用した. 本評価では入力データは MPEG2 エンコードでは 30 フレームの SIF (352x240) サイズの動画を, MPEG2 デコードでは 30 フレームの SIF (352x240) サイズの圧縮されたストリームデータを, MP3 エンコードでは 32 フレームのステレオ PCM を, JPEG 2000 エンコードでは 400x300 ピクセルの

カラー画像をそれぞれ用い, プログラム開始時に集中共有メモリ上に存在しているという想定で行った.

ただし, JPEG 2000 エンコード評価においては実行時の処理コストを OSCAR 並列化コンパイラにおける粗粒度タスクスケジューリング時に用いた.

5.4 富士通 FR1000 マルチコア上での性能評価

図 15 に FR1000 マルチコア上での評価結果を示す. 本評価では API 解釈系と gcc コンパイラを用い, 最適化オプションとして O3 を指定した. API を用いて同期変数を WorkRAM に配置し, CSM のバンク構成を考慮せず生成したコード (同一バンク), API を用いて同期変数を WorkRAM に配置し, CSM のバンク構成を考慮して生成したコード (バンク分け) の評価を行った. ただし, MP3 エンコードに対しては用いた gcc コンパイラの動作不良により最適化レベルを O0 として評価を行った.

OSCAR 並列化コンパイラによる情報家電用マルチコア並列化 API を用いた 4PE での並列処理では, 1PE に対して MPEG2 エンコードでは 3.76 倍, MPEG2 デコードでは 2.08 倍, MP3 エンコードでは 3.75 倍, JPEG 2000 エンコードでは 3.47 倍の並列処理性能が得られた. 本評価では CSM のバンク構成を考慮することにより並列性能の向上を得ることができた. ただし, MP3 エンコード, JPEG 2000 エンコードにおいてバンク構成を考慮することによる性能向上が多く見られないのは, メモリアクセスと比較して倍精度浮動小数点演算による演算処理コストが多いために相対的にメモリアクセスオーバーヘッドが低いためと考えられる.

5.5 ルネサス 日立 早大 RP1 マルチコア上での性能評価

図 16 に RP1 マルチコア上での評価結果を示す.

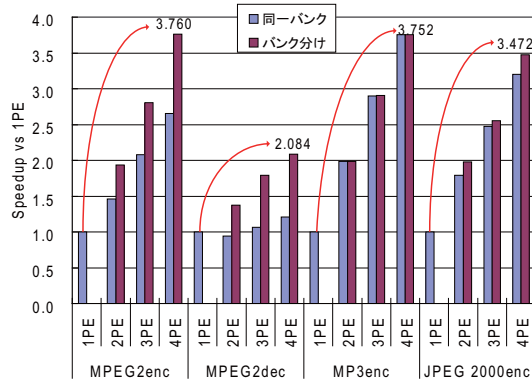


図 15 富士通 FR1000 マルチコア上での並列処理性能
Fig. 15 Evaluation Result on FR1000 Multicore

RP1 マルチコアでは SMP モードにおいて API 解釈系と SH 用最適化コンパイラである SH C コンパイラを用い、SMP 用コードの評価を行った。

OSCAR 並列化コンパイラによる情報家電用マルチコア並列化 API を用いた 4PE での並列処理では、1PE に対して MPEG2 エンコードでは 3.35 倍、MPEG2 デコードでは 3.24 倍、MP3 エンコードでは 3.50 倍、JPEG 2000 エンコードでは 3.17 倍の並列処理性能が得られた。SMP モードを持つ RP1 マルチコアでは SMP 用コードにおいて並列性能が得られているが、JPEG 2000 エンコードの 3PE 処理時の並列性能の鈍化はループの分割数による負荷の不均衡によるものと考えられる。

6. ま と め

本論文では、情報家電用マルチコア並列化 API を用いて、OSCAR 並列化コンパイラによる富士通 FR1000 マルチコア及びルネサステクノロジ、日立製作所、早稲田大学共同開発 RP1 マルチコア上での並列性能評価をマルチメディア処理を対象として行った。FR1000 マルチコアでは 4PE 処理時に 1PE に対して MPEG2 エンコードでは 3.76 倍、MPEG2 デコードでは 2.08 倍、MP3 エンコードでは 3.75 倍、JPEG 2000 エンコードでは 3.47 倍の並列処理性能が得られ、RP1 マルチコアでは 4PE 処理時に 1PE に対して MPEG2 エンコードでは 3.35 倍、MPEG2 デコードでは 3.24 倍、MP3 エンコードでは 3.50 倍、JPEG 2000 エンコードでは 3.17 倍の並列処理性能が得られた。この結果から本 API により OSCAR 並列化コンパイラを用いて異なるプロセッサコアを持つマルチコア上での並列化を行えることが確かめられた。

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”，NEDO“先進ヘテロジニアスマルチプロセッサ技術”及び NEDO“情報家電用ヘテロ

ジニアスマルチコア技術”の支援により行われた。

参 考 文 献

- 1) Pham, D. et al.: The Design and Implementation of a First-Generation CELL Processor, *In Proceeding of the IEEE International Solid-State Circuits Conference* (2005).
- 2) Cornish, J.: Balanced Energy Optimization, *International Symposium on Low Power Electronics and Design* (2004).
- 3) Suga, A. et al.: FR-V Single-Chip Multicore Processor:FR1000, *Fujitsu Sci Tech J*, Vol. 42, No. 2, pp. 190–199 (2006).
- 4) 木村浩三ほか: デジタル家電統合プラットフォーム UniPhier におけるメディアプロセッサ, DA シンポジウム (2005).
- 5) Kamei, T.: SH-X3 : An Enhanced SuperH Core for Low-power MP Systems, *Fall Microprocessor Forum 2006* (2006).
- 6) The Multicore ASSOCIATION: <http://www.multicore-association.org/>.
- 7) 木村啓二ほか: シングルチップマルチプロセッサ上での近細粒度並列処理, 情報処理学会論文誌, Vol. 40, No. 5 (1999).
- 8) Kimura, K. et al.: Multigrain Parallel Processing on Compiler Cooperative Chip Multiprocessor, *Proc. of 9th Workshop on Interaction between Compilers and Computer Architectures (INTERACT-9)* (2005).
- 9) *OpenMP Application Program Interface Version 2.5* (2005).
- 10) 石坂一久ほか: 共有メモリマルチプロセッサ上でのキャッシュ最適化を考慮した粗粒度タスク並列処理, 情報処理学会論文誌, Vol. 43, No. 4 (2002).
- 11) Iwata, E. et al.: Exploiting Coarse-Grain Parallelism in the MPEG-2 Algorithm, *Technical Report CSL-TR-98-771* (1998).
- 12) C. Lee et al.: MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems, *30th International Symposium on Microarchitecture (MICRO-30)* (1997).
- 13) *UZURA3:MPEG1/LayerIII Encoder in FORTRAN90*. http://members.at.infoseek.co.jp/kitaurawa/index_e.html.
- 14) R Grosbois et al.: <http://jj2000.epfl.ch/>.
- 15) Intel: A Fast Precise Implementation of 8x8 Discrete Cosine Transform Using the Streaming SIMD Extensions and MMX Instructions (1999). AP-922, Order Number 742474-001.

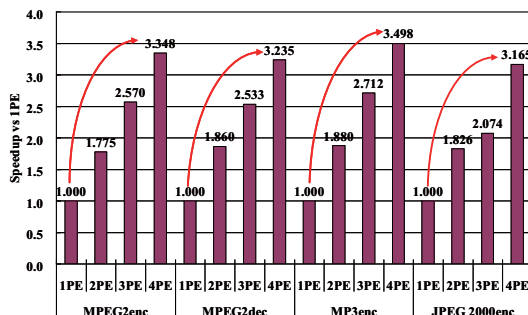


図 16 ルネサス 日立 早大 RP1 マルチコア上での並列処理性能
Fig.16 Evaluation Result on RP1 Multicore