

マルチコアプロセッサ上でのマルチメディア処理の並列化

宮本孝道[†]

田村圭[†]

田野裕秋[†]

見神広紀[†]

浅香沙織[†]

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．本稿ではこのようなマルチメディアアプリケーションのマルチコアプロセッサ上での並列化について述べる．特に動画処理における 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 エンコードを例として OSCAR マルチグレイン自動並列化コンパイラを用いた富士通株式会社製 FR1000 と早稲田大学，ルネサステクノロジ，株式会社日立製作所共同開発 RP1 マルチコアでの並列処理結果について述べる．

Parallelization for Multimedia Processing on Multicore Processors

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

Multicore processors have attracted much attention to handle the increase of power consumption, the slowdown 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 the consumer electronics devices like mobile phones, digital TV and games. This paper describes parallelization methods of multimedia applications on the multicore processors. Especially in this paper, MPEG2 encoding and MPEG2 decoding are selected as examples of video sequence processing, MP3 encoding is selected as an example of audio processing, JPEG 2000 encoding is selected as an example of picture processing. OSCAR multigrain parallelizing compiler parallelizes these media applications using newly developed multicore API. This paper evaluates parallel processing performances of these multimedia applications on the FR1000 multicore processor developed by Fujitsu Ltd, and the RP1 multicore processor jointly-developed by Waseda University, Renesas Technology Corp. and Hitachi Ltd.

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と共に並列性抽出の限界，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．SCE/IBM/東芝の Cell¹⁾，NEC/ARM の MPCore，MP211²⁾，富士通 FR1000³⁾，パナソニック Uniphier，ルネサステクノロジ SH-X3⁴⁾ といった情報家電向け組込み用マルチコアや，PC，サーバ向けのインテル Dual コア Xeon⁵⁾ や Core 2 Duo，AMD の Dual/Quad コア Opteron⁶⁾，ワークステーション，ハイエンドサーバ用に開発された Sun SPARC T1，T2，そして IBM Power4,5,6⁷⁾ などが例として挙げられる．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，タスクレベルのような粒度の粗い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，OS とも協調し低周波・数低電圧で動作させ，適切に並列処理することで，高性能化，低消費電力化が実現可能なアーキテクチャとしても期待されている．

筆者等は従来より自動マルチグレイン並列化コンパイラとの協調動作により実効性能が高く価格性能比の良いコンピュータシステムの実現を目指す OSCAR マルチコアアーキテクチャを提案している^{8),9)}．この OSCAR マルチコアアーキテクチャは，全てのプロセッサコアがアクセスできるオンチップあるいはオフチップ集中共有メモリ (CSM) の他に，プロセッサコアのプライベートデータを格納するローカルデータメモリ (LDM) とプロセッサコア間の同期やデータ転送に使用する分散共有メモリ (DSM)，そして，プロセッサコアと非同期に動作可能なデータ転送ユニット (DTU) を持つ．

このようなマルチコア上でのマルチメディア処理のマルチグレイン並列化を実現するためには適切なデータ分割，プロセッサ近傍メモリへのデータ配置，メモリ間のデータ転送最適化の適用が重要となる．本稿ではこのようなマルチメディア処理の例として，動画処理の MPEG2 エンコード，MPEG2 デコード，音声処理の MP3 エンコード，静止画処理の JPEG 2000 エンコードを対象とした並列化を行う．

[†]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 Tel: 03-5286-3371

[†]Department of Computer Science and Engineering,
Fundamental Science and Engineering, Faculty of Science
and Engineering, Waseda University 3-4-1 Ohkubo,
Shinjuku-ku, Tokyo, Japan 169-8555 Tel: +81-3-5286-3371

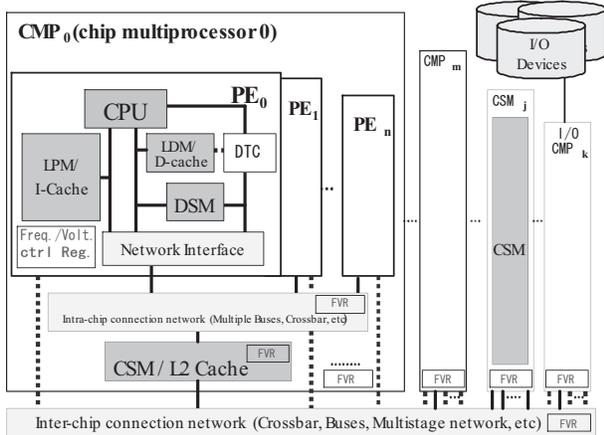


図 1: OSCAR マルチコアアーキテクチャ

本稿の構成を以下に示す。第 2 章では対象とするマルチコアアーキテクチャについて述べる。第 3 章ではマルチコアプロセッサ用並列化 API について述べる。第 4 章では本稿で扱う粗粒度タスク並列処理手法について述べる。第 5 章では対象マルチメディアアプリケーションと並列化手法について述べる。第 6 章ではマルチメディアアプリケーションの富士通 FR1000 上、早稲田大学 ルネサス 日立製作所 RP1 上での性能評価結果について述べる。第 7 章で本稿のまとめを述べる。

2 対象マルチコアアーキテクチャ

本章では、NEDO“リアルタイム情報家電用マルチコア技術の研究開発”プロジェクトにおいて標準的マルチコアメモリアーキテクチャとして採択された OSCAR マルチコアアーキテクチャと評価対象とする OSCAR マルチコアアーキテクチャのサブセットとなるメモリ構成を持つマルチコアアーキテクチャである富士通 FR1000 と 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早稲田大学 ルネサス 日立製作所 RP1 について説明する。

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自動マルチグレイン並列化コンパイラとの協調動作により、実効性能が高く価格性能比の良いコンピュータシステムの実現を目指したアーキテクチャである。

図 1 に示すように OSCAR マルチコアメモリアーキテクチャは 1 つのチップ上に複数のプロセッサエレメント (PE) を持つ。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)、そして CPU と非同期にバースト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 はバスやクロスバといった 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。

2.2 富士通 FR1000 ボード

FR1000 は図 2 に示すように、8 並列 VLIW の FR550 プロセッサコアを 4 コア搭載したマルチコアプロセッサであり、2 チャンネルのメインメモリコントローラ、WorkRAM 間の転送用に Direct Memory Access Controller (DMAC) を持つ。FR550 プロセッサコアは整数型算術命令、浮動小数点型算術命令、16bit メディア命令、固定小数点型算術命令の命令を持つ。FR1000 の仕様概要を表 1 に示す。

OSCAR マルチコアと FR1000 のメモリ対応関係は、OSCAR マルチコアにおける分散共有メモリ (DSM) が

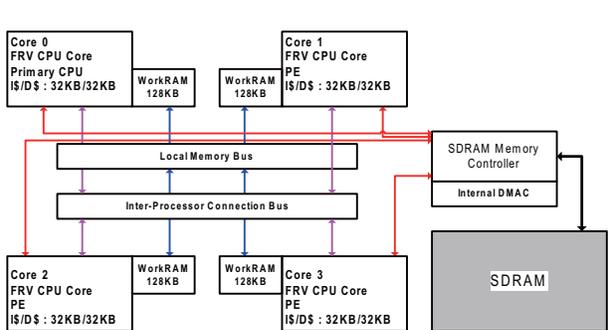


図 2: FR1000 ブロック図概要

表 1: FR1000 ボード仕様概要

| プロセッサコア | FR550 (8 並列 VLIW) 500MHz |
|----------|---------------------------------|
| システムバス | 64bit 166MHz |
| WorkRAM | 128 KB |
| 命令キャッシュ | 32 KB (4way), 64 Byte line, LRU |
| データキャッシュ | 32 KB (4way), 64 Byte line, LRU |

FR1000 の WorkRAM, off-chip 集中共有メモリ (off-chip CSM) が FR1000 の SDRAM にあたる。

2.3 早大 ルネサス 日立 RP1 ボード

本マルチコアプロセッサは図 3 に示すように、SH-4A (SH-X3) コアを 4 コア搭載したホモジニアスマルチコアとなっており、4 コアがそれぞれ独立して周波数制御が可能である。また、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 Symmetric Multi-Processing (SMP) モード、リアルタイム制約が保証しやすい Asymmetric Multi-Processing (AMP) モード、およ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。RP1 の仕様概要を表 2 に示す。

OSCAR マルチコアと RP1 のメモリ対応関係は、OSCAR マルチコアにおけるローカルデータメモリ (LDM) が RP1 の OLRAM, ローカルプログラムメモリ (LPM) が RP1 の ILRAM, 分散共有メモリ (DSM) が RP1 の URAM, on-chip 集中共有メモリ (on-chip CSM) が図 3 の CSM にあたる。

3 マルチコア並列化 API

本章では FR1000 と RP1 のように異なるプロセッサアーキテクチャのマルチコア上での OSCAR コンパイラによる並列化を可能とするために策定された情報家電用マルチコア並列化 API と本 API を用いた性能評価フローについて説明を行う。情報家電用マルチコア並列化 API は 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 (委員に参加, (株) 日立製作所, (株) ルネサステクノロジ, (株) 富士通

表 2: RP1 ボード仕様概要

| プロセッサコア | SH-4A 600 MHz |
|----------|---------------------------------|
| システムバス | 300 MHz |
| 命令キャッシュ | 32 KB (4way), 32 Byte line, LRU |
| データキャッシュ | 32 KB (4way), 32 Byte line, LRU |
| ILRAM | 8 KB |
| OLRAM | 16 KB |
| URAM | 128 KB |
| CSM | 128 KB / chip |

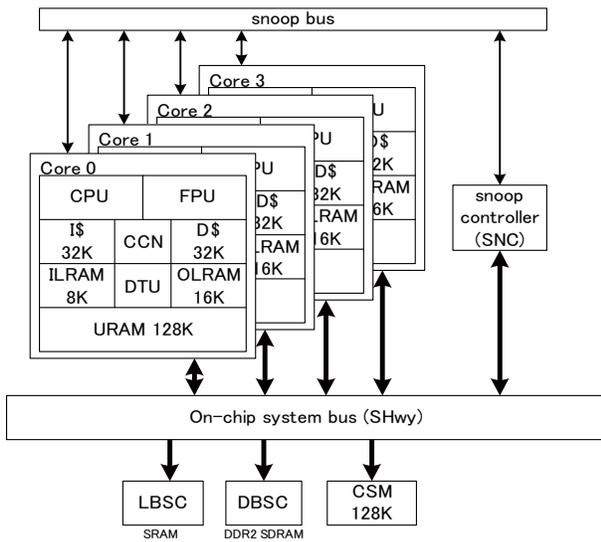


図 3: RP1 ブロック図概要

研究所, (株) 東芝, (株) 松下電器産業, (株) 日本電気) で策定された OpenMP API¹⁰⁾ 互換形式のディレクティブ (C と FORTRAN に対応) で OpenMP API のサブセットに主に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, オンチップ共有メモリ, データ転送ユニット, 消費電力制御用に独自に定義したディレクティブを加えた構成となっている。OSCAR 自動並列化コンパイラを用いて C あるいは FORTRAN + API の並列プログラムを出力することにより, 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, 各社のマルチコア上での並列処理が可能となる。具体的には, スレッド生成, 同期, 排他制御は OpenMP 互換となっており, スレッド生成に parallel sections, メモリアクセス順を保証する flush, 排他制御を行うための critical を用いる。これら 3 つのサポートにより, SMP 環境でのマルチグレイン並列処理が実現できる。独自定義のディレクティブとして, データのメモリ配置, DMAC によるデータ転送, 周波数・電源制御用の API の策定を行っている。

情報家電用マルチコア API の評価フローは図 4 に示すように, 逐次型 C プログラムを入力として OSCAR 並列化コンパイラにより出力される情報家電用マルチコア API を含んだ並列化プログラムを API 解釈を含む各社コンパイラによりマシンコードを生成し, マルチコア上で評価を行うフローとなる。

4 粗粒度タスク並列処理

今回のマルチメディア処理の並列化に利用する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) に分割を行う。そして, 3 種類に分割されたマクロタスクを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) に割り当てて実行することにより, マクロタスク間の並列性を利用する並列処理手法である。

4.1 マクロタスクの生成

粗粒度タスク並列処理では, まずソースプログラムを BPA, RB, SB の 3 種類のマクロタスクに分割する。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。

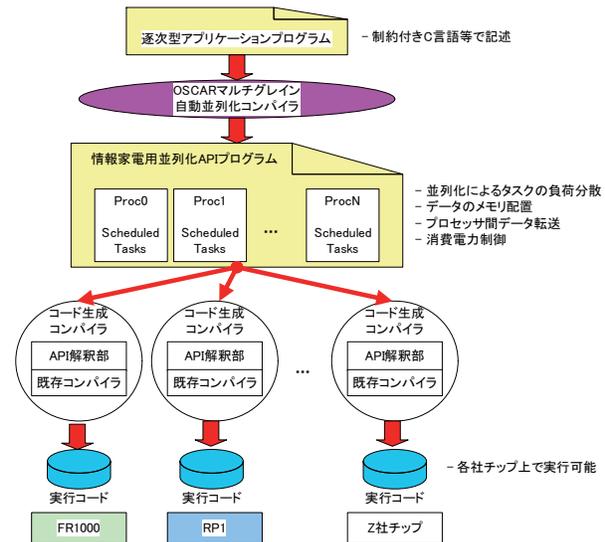


図 4: マルチコア並列化 API を用いた各社マルチコア評価フロー

4.2 マクロフローグラフ (MFG) の生成

マクロタスクの生成後, マクロタスク間のコントロールフローとデータ依存を解析し, その結果を表す図 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) の各ノードはマクロタスクを表し, 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 ではエッジの矢印は省略されているが, エッジの方向は下向を仮定している。

4.3 マクロタスクグラフ (MTG) の生成

MFG はマクロタスク間のコントロールフローとデータ依存を表すが, 並列性は表していない。並列性を抽出するためには, 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マクロフローグラフに対して行う。マクロタスク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マクロタスクが最も早い時点で実行可能になる条件である。

マクロタスク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 と同様に, MTG におけるノードはマクロタスクを表し, ノード内の小円はマクロタスク内の条件分岐を表している。実線のエッジはデータ依存を表し, 点線のエッジは拡張されたコントロール依存を表す。拡張されたコントロール依存とは, 通常のコントロール依存だけでなく, データ依存とコントロール依存を複合的に満足させるため先行ノードが実行されないことを確定する条件分岐を含んでいる。また, エッジを束ねるアークには 2 つの種類がある。実線アークはアークによって束ねられたエッジが AND 関係にあることを, 点線アークは束ねられたエッジが OR 関係にあることを示している。MTG においては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。また, 矢印を持つエッジはオリジナルのコントロールフローを表す。

4.4 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, 生成されたマクロタスクはプロセッサグループ (PG) に割り当てられて実行される。PG にマクロタスクを割り当てるスケジューリング手法として, コンパイル時に割り当てを決めるスタティックスケジューリングと実行時に割り当てを決めるダイナミックスケジューリングがあり, マクロタスクグラフの形状, 実行時不確定性などを元に選択される。スタティッ

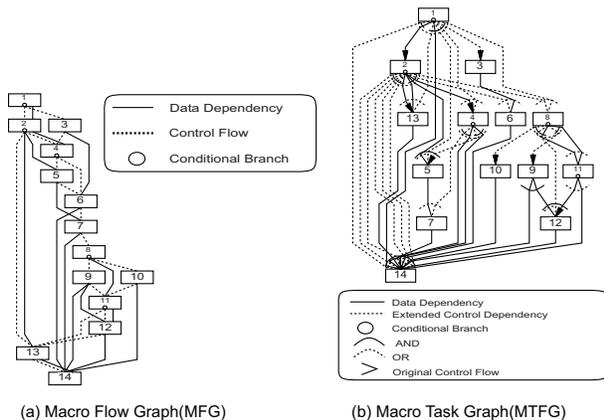


図 5: マクロフローグラフとマクロタスクグラフ

タスクスケジューリングは、マクロタスクグラフがデータ依存エッジのみを持つ場合に適用され、コンパイラがコンパイル時にマクロタスクの PG への割り当てを決定する方式である。スタティックスケジューリングでは、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。本稿では粗粒度タスクスケジューリング手法として、ETF/CP (Earliest Task First/ Critical Path) 法に対して図 6(a) で網掛けされた MT のようなデータ依存を持つ複数ループを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、図 6(b) のように複数のループを整合して分割を行うデータローカライゼーション手法¹¹⁾を適用する。ループ整合分割可能なループ集合において同一のデータにアクセスする分割後の MT 集合内では、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データ転送の最小化を目的とし、データローカライゼーショングループ (DLG) 内のタスクを同一のプロセッサへの連続タスク割り当てを行うアルゴリズム ETF/CP considering DLG 法を用いてスケジューリングを行った。

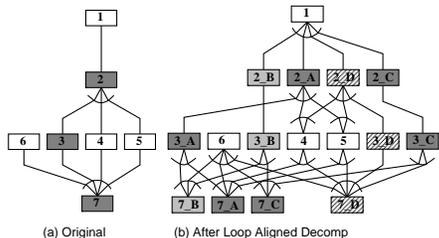


図 6: ループ整合分割の例

5 マルチメディアアプリケーションの並列化手法

情報家電上でのコーデック等のマルチメディア処理が重要になり、アルゴリズムの複雑化から高い処理性能が要求される。従来、本要求に応えるために、DSP¹²⁾、マルチメディアプロセッサ¹³⁾の利用、Intel SSE¹⁴⁾、Sun VIS¹⁵⁾、MIPS MDMX¹⁶⁾等のマルチメディア拡張命令セットの追加や並列化と SIMD 最適化を行う ALP¹⁷⁾が行われている。

本稿ではマルチメディア処理におい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 エンコードを選択した。以下に各マルチメディアアプリケーションの並列化手法について述べる。

5.1 MPEG2 エンコード

MPEG2 エンコード処理ブロック図とデータ構造を図 7 に示す。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 エンコードではマクロブロックレベルの並列性を利用する¹⁸⁾。ただし、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で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。MPEG2 エンコード処理における並列性抽出後の MTG を図 8(a) に示す。

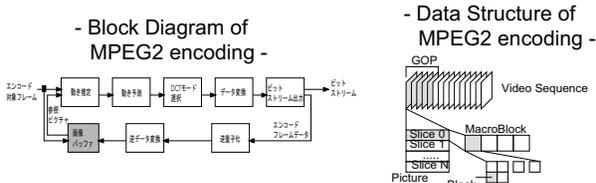
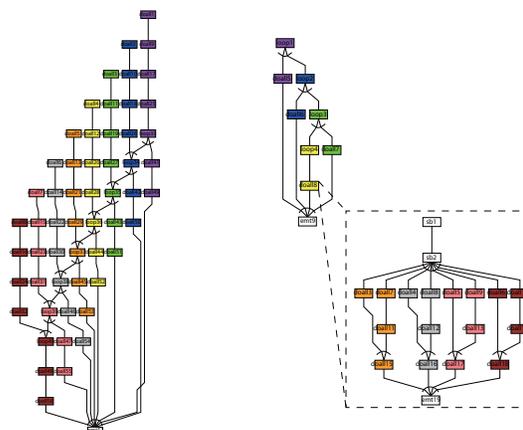


図 7: MPEG2 エンコードの処理ブロック図とデータ構造



(a) MTG of MPEG2 encoding (b) MTG of MPEG2 decoding

図 8: 並列性抽出後の MPEG2 エンコード、デコードの MTG 例

5.2 MPEG2 デコード

MPEG2 デコードのブロック図を図 9 に示す。MPEG2 デコード処理は、可変長復合化、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利用可能であり、本稿ではスライスレベルの並列性を利用する。スライスに対する可変長復合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ンニング手法を適用した¹⁹⁾。プレスキャンニング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。MPEG2 デコードにおける並列性抽出後のマクロタスクグラフを図 8(b) に示す。

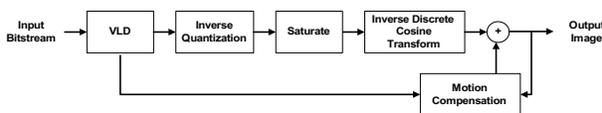


図 9: MPEG2 デコードの処理ブロック図

5.3 MP3 エンコード

MP3 のエンコード処理ブロック図とデータ構造を図 10 に示す。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 エンコード処理ではフレームレベルの並列性を利用する。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。MP3 エンコードにおける並列性抽出後のマクロタスクグラフを図 11(a) に示す。

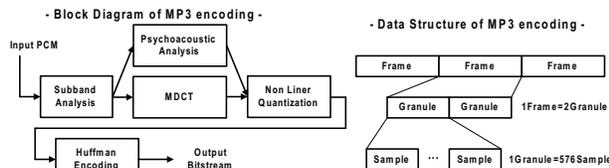


図 10: MP3 エンコードの処理ブロック図とデータ構造

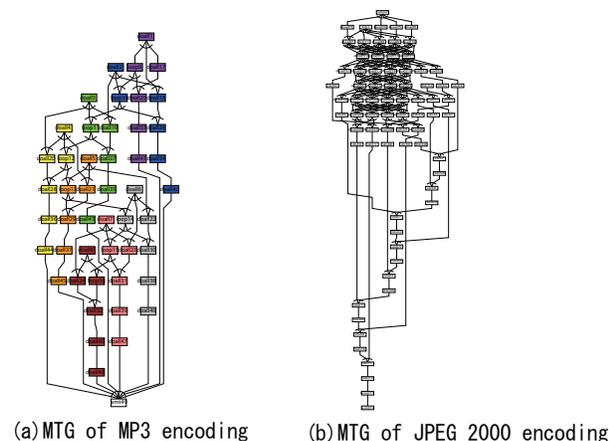


図 11: 並列性抽出後の MP3 エンコード, JPEG 2000 エンコードの MTG 例

5.4 JPEG 2000 エンコード

JPEG 2000 エンコードブロック図とデータ構造を図 12 に示す。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、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 エンコードでは DC レベル変換、離散ウェーブレット変換では画像の縦方向、横方向の並列性を利用し、スカラ量子化、EBCOT では 64x64 ピクセルのコードブロック単位の並列性を利用する。ただし、離散ウェーブレット変換においては縦方向、横方向の処理間や各レベルの処理間では転置転送により、データ転送が発生する。ここで、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。JPEG 2000 エンコードにおける並列性抽出後のマクロタスクグラフを図 11(b) に示す。

6 性能評価

本章では第 5 章で述べたマルチメディアアプリケーションの並列処理性能について述べる。

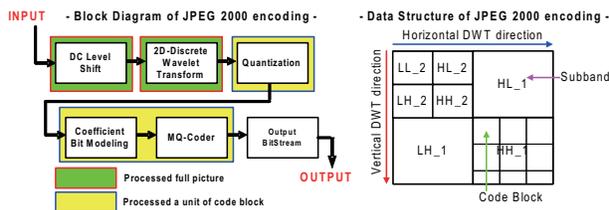


図 12: JPEG2000 エンコードの処理ブロック図とデータ構造

6.1 評価条件

評価アプリケーションとしては第 5 章で示した 4 つのメディアアプリケーションを用いた。表 3 に評価を行った各メディアアプリケーションの参照元プログラム、特徴的な処理パラメータを示す。記述していない処理パラメータは各参照プログラムにおけるデフォルトパラメータを使用した。表 4 に性能評価に使用した入力データを示す。本評価では入力データは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。

各マルチメディアアプリケーションのプログラムコード中には OSCAR コンパイラで解釈を行うための並列化階層の指定とループ分割における分割数指定の指示文を挿入したコードを用い、JPEG 2000 エンコードにおいてはプロファイル情報として実行時のコストを OSCAR コンパイラのスケジューリング時に用いた。

表 3: 評価メディアアプリケーション仕様

| MPEG2 エンコード | |
|----------------|---|
| 参照元プログラム | MediaBench ²⁰⁾ mpeg2encode |
| 評価パラメータ | Frames in GOP : 15 I/P frame distance : 3 Picture Type : Frame Picture Bitrate : 5Mbps |
| MPEG2 デコード | |
| 参照元プログラム | MediaBench mpeg2decode |
| MP3 エンコード | |
| 参照元プログラム | UZURA MP3 encoder ²¹⁾ |
| 評価パラメータ | Bitrate : 128kbps CBR Compression : lossy |
| JPEG2000 エンコード | |
| 参照元プログラム | JJ2000 ²²⁾ |
| 評価パラメータ | Bitrate : 100bpp WaveletLevel : 5 Compression : lossy |

6.2 富士通 FR1000 上での性能評価

図 13 に富士通 FR1000 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 を指定して評価を行った。ただし、MPEG2 デコード、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 コンパイラによるマルチコア並列化 API を用いた 4 プロセッサでの並列処理において、MPEG2 エンコードで 1 プロセッサ処理時の 4.60 倍、MPEG2 デコードで 1 プロセッサ処理時の 2.17 倍、MP3 エンコードで 1 プロセッサ処理時の 2.82 倍、JPEG 2000 エンコードで 1 プロセッサ処理時の 3.09 倍の速度向上が得られた。

6.3 早大 ルネサス 日立 RP1 上での性能評価

図 14 に早大 ルネサス 日立 RP1 上での評価結果を示す。本評価では API 解釈系と SH C コンパイラを用

表 4: 評価入力データ

| アプリケーション | 入力データ |
|----------------|--|
| MPEG2encode | SIF(352x240) 30 フレーム 4:2:0 format |
| MPEG2decode | SIF(352x240) 30 フレーム encoded with 5Mbps Bitrate |
| MP3encode | 44.1kHz Stereo PCM 32 フレーム |
| JPEG2000encode | 400x300 pixel (color) |

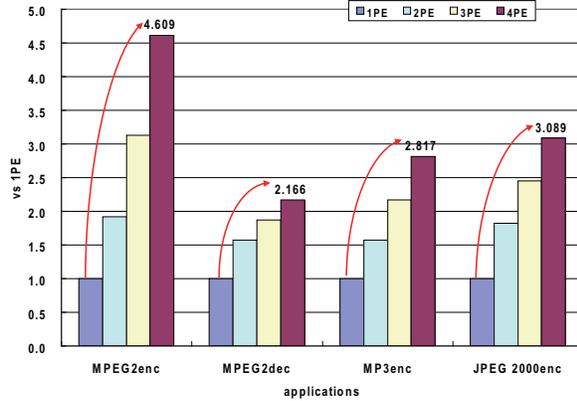


図 13: 富士通 FR1000 上での並列処理性能

い、最適化オプションとして speed, goptimize を指定して評価を行った。OSCAR コンパイラによるマルチコア並列化 API を用いた 4 プロセッサでの並列処理において、MPEG2 エンコードで 1 プロセッサ処理時の 3.35 倍、MPEG2 デコードで 1 プロセッサ処理時の 3.34 倍、MP3 エンコードで 1 プロセッサ処理時の 3.50 倍、JPEG 2000 エンコードで 1 プロセッサ処理時の 3.17 倍の速度向上が得られた。

7 まとめ

本稿では、マルチメディアアプリケーションの例として動画処理 MPEG2 エンコード、MPEG2 デコード、音声処理 MP3 エンコード、静止画処理 JPEG 2000 エンコードに対する粗粒度タスク並列処理を富士通 FR1000、早大ルネサス日立 RP1 上で評価を行った。FR1000 上では、逐次処理に対する 4 プロセッサでの並列処理で MPEG2 エンコードで 4.60 倍、MPEG2 デコードで 2.17 倍、MP3 エンコードで 2.82 倍、JPEG 2000 エンコードで 3.09 倍の速度向上が得られた。RP1 上では、逐次処理に対する 4 プロセッサでの並列処理で MPEG2 エンコードで 3.35 倍、MPEG2 デコードで 3.34 倍、MP3 エンコードで 3.50 倍、JPEG 2000 エンコードで 3.17 倍の速度向上が得られた。

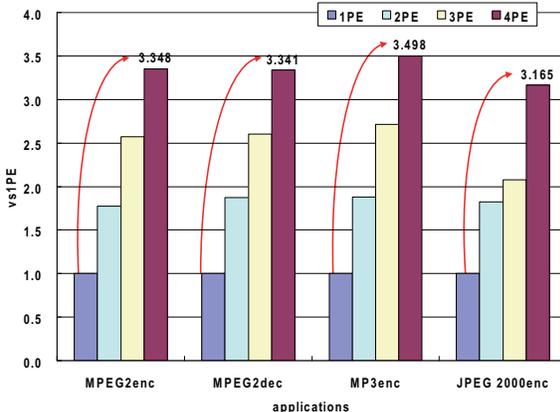


図 14: 早大ルネサス日立 RP1 上での並列処理性能

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. and Matsunami, K.: Introducing the FR 500 embedded microprocessor, Vol. 20, pp. 21–27 (2000).
- [4] Kamei, T.: SH-X3 : An Enhanced SuperH Core for Low-power MP Systems, *Fall Microprocessor Forum 2006* (2006).
- [5] Intel: <http://www.intel.com/multi-core/>.
- [6] Keltcher, C. N., McGrath, K., Ahmed, A. and Conway, P.: The AMD Opteron processor for multiprocessor servers, *IEEE Micro*, Vol. 23, pp. 66–76 (2003).
- [7] Friedrich, J. and et.al.: Design of the Power6 Microprocessor, *IEEE ISSCC*, pp. 96–97 (2007).
- [8] 木村, 尾形, 岡本, 笠原: 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).
- [9] 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).
- [10] *OpenMP Application Program Interface Version 2.5* (2005).
- [11] 石坂, 中野, 八木, 小幡, 笠原: 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).
- [12] Eiji Iwata and et.al.: A 2.2GOPS Video DSP with 2-RISC MIMD, 6-PE SIMD Architecture for Real-Time MPEG2 Video Coding/Decoding, *IEEE International Solid State Circuits Conference Digest of Technical Papers*, pp. 258–259 (1997).
- [13] Yong Yao: Chromatic’s Mpaact 2 Boosts 3D, *Microprocessor Report*, Vol. 10 (1996).
- [14] S. K. Raman, V. Pentkovki and J.Keshava: Implementing Streaming SIMD Extensions on the Pentium III Processor, Vol. 20, No. 4 (2000).
- [15] March Tremblay and et.al.: VIS Speeds New Media Processing, *IEEE Micro*, pp. 10–20.
- [16] Earl Killian: Extending the MIPS Instruction Set for Digital Media and Emerging Applications, *Microprocessor Forum 96* (1996).
- [17] Ruchiana Sasanka, Man-Lap Li, Sarita V. Adve, Yen-Kuang Chen and Eric Debes: ALP:Efficient support for all levels of parallelism for complex media applications, *ACM Transactions on Architecture and Code Optimization (TACO)*, Vol. 4.
- [18] 小高剛, 中野啓史, 木村啓二, 笠原博徳: チップマルチプロセッサ上での MPEG2 エンコードの並列処理, *情報処理学会論文誌*, Vol. 46, No. 9 (2005).
- [19] Iwata, E. and Olukotun, K.: Exploiting Coarse-Grain Parallelism in the MPEG-2 Algorithm, *Technical Report CSL-TR-98-771* (1998).
- [20] C. Lee, M. Potkonjak and W. H. Mangione-Smith: MediaBench: A Tool for Evaluating and Synthesizing Multimedia and Communications Systems, *30th International Symposium on Microarchitecture (MICRO-30)* (1997).
- [21] *UZURA3:MPEG1/LayerIII Encoder in FORTRAN90*. http://members.at.infoseek.co.jp/kitaurawa/index_e.html.
- [22] R Grosbois, D Santa Cruz, J. A. B. B. D. B. F. H. G. M. and Onno, P.: <http://jj2000.epfl.ch/>.