

ヘテロジニアスマルチコアプロセッサ上での スタティックスケジューリングを用いた MP3 エンコーダの並列化

和田 康孝^{†1} 林 明宏^{†1} 益 浦 健^{†1}
白 子 準^{†1} 中 野 啓 史^{†1} 鹿 野 裕 明^{†1}
木 村 啓 二^{†1} 笠 原 博 徳^{†1}

情報家電の市場拡大にともない、低消費電力でありながら高い性能を実現するプロセッサが求められるようになってきている。この要求に対応するため、汎用プロセッサに加え、動的再構成可能プロセッサ (DRP) や信号処理用プロセッサ (DSP) 等のアクセラレータを 1 チップ上に複数集積したヘテロジニアスマルチコアアーキテクチャが注目を集めている。このようなヘテロジニアスマルチコアにおいては、処理の特性やコア間のデータ転送を考慮して適切に各コアに処理を割り当てる必要がある。本論文では、このようなヘテロジニアスマルチコア用の粗粒度タスクスタティックスケジューリング手法を提案する。本論文で提案するスタティックスケジューリング手法では、ループやサブルーチン、基本ブロック間の並列性を利用する粗粒度タスク並列処理において、各タスクがどのコアで実行可能か等の特性、各コア間でのデータ転送オーバーヘッドを考慮して処理時間を最小とするように汎用コアあるいはアクセラレータに割り当て、さらにコア間でのデータ転送を DMA を用いてタスク処理とオーバーラップして行う。これによりプログラムの階層的な並列性とチップ上のアクセラレータを有効に利用し、処理の高速化を図ることができる。本手法を用い、世界初のヘテロジニアス並列化コンパイラを開発し MP3 エンコーダに適用し評価した結果、SH4A 1 コアのみを用いた場合に対して、SH4A 4 コアで 3.99 倍、SH4A 2 コアと DRP 2 コアで 14.55 倍、SH4A 4 コアと DRP 4 コアを用いたときに 25.20 倍の性能向上を得られることが確認できた。

Parallelization of MP3 Encoder Using Static Scheduling on a Heterogeneous Multicore

YASUTAKA WADA,^{†1} AKIHIRO HAYASHI,^{†1}
TAKESHI MASUURA,^{†1} JUN SHIRAKO,^{†1}
HIROFUMI NAKANO,^{†1} HIROAKI SHIKANO,^{†1}
KEIJI KIMURA^{†1} and HIRONORI KASAHARA^{†1}

Heterogeneous multicore architectures integrating various kind of accelerators like dynamically reconfigurable processors (DRPs) or digital signal processors (DSPs) in addition to general purpose processor cores have attracted much attention to realize high performance with low power consumption. These heterogeneous multicores require scheduling schemes considering characteristics of tasks on each core and data transfers on chips. This paper proposes a static scheduling scheme for coarse grain task parallel processing on a heterogeneous multicore processor with overlapping data transfer and task execution. In the proposed scheme, the compiler extracts parallelism using coarse grain parallel processing and assigns tasks considering characteristics on each core to minimize the execution time of an application. Performance of the proposed scheme is evaluated on a heterogeneous multicore processor using an MP3 encoder. Heterogeneous configurations give us 14.55 times speedup with two SH4As and two DRPs and 25.20 times speedup with four SH4As and four DRPs against sequential execution with one SH4A core.

1. はじめに

携帯電話、ゲーム機、DVD レコーダ、デジタル TV 等に代表される情報家電の市場拡大にともない、その付加価値の源泉となる低消費電力高性能プロセッサに対する要求は年々高まっている。特に、価格性能比の向上や開発期間の短縮、低消費電力化はそのプロセッサの市場競争力を大きく左右する。これらの要求に対応するため、現在では組み込みシステムにおいても、1 つのチップ上に複数のプロセッサコアを集積したマルチコアアーキテクチャ^{1),2)} が積極的に用いられるようになってきている。

^{†1} 早稲田大学理工学術院情報理工学科

Department of Computer Science and Engineering, Faculty of Science and Engineering, Waseda University

さらに、情報家電システムにおいてよく用いられるメディアアプリケーション等の処理を高速化するために、信号処理用プロセッサ (DSP) や動的再構成可能プロセッサ (DRP) 等のアクセラレータをチップ上にあわせて集積するヘテロジニアスマルチコア^{3),4)} が注目を集めている。

ヘテロジニアスマルチコアプロセッサでは、アクセラレータを利用することで高速な演算処理と低消費電力化を両立させることが可能となるが、その反面、効率良くアクセラレータを利用するためのチューニングが大変困難なものとなる。そのため、実効性能の向上と開発期間の短縮を両立させるためには、チップ内の各コアの特性やデータ転送を考慮して適切に処理を割り当てる手法およびそれを実現する自動並列化コンパイラの開発が求められる。

ヘテロジニアスな並列処理環境に関しては、従来から主にグリッドやクラスタを対象としてスケジューリング手法が提案されており^{5),6)}、データを生成するタスクを複製して実行することによりデータ転送を削減する手法⁷⁾ や、プログラム中の逐次処理が必要な部分の割合を考慮した負荷分散の手法⁸⁾ 等が提案されている。しかしながら、従来の手法では割当て先によってタスクの実行時間が異なるという条件のみで、タスクの特性によって割当て先が制限されるといった条件を考慮していない。ヘテロジニアスマルチコアでは、特定の処理に特化した専用 HW 等をアクセラレータとして搭載することも考えられ、それに対応したタスクの割当て手法が求められる。

また、ヘテロジニアスマルチコアに関しては、同一の命令セットを持つ異なる性能のコアを複数搭載することでシステムのスループットを向上させようとするもの⁹⁾ や、特定アプリケーション向けの専用 IP を多数搭載した携帯端末向けのもの¹⁰⁾ 等が提案、発表されている。ただし、チップ上の計算資源への処理の割当てが固定的であったり、手動でのチューニングが必要であったりすることが多く、ソフトウェア生産性の向上のためには並列化コンパイラとの協調が重要である。

本論文では、プログラム全域から並列性を引き出して利用することが可能な階層的マルチグレイン並列処理¹¹⁾ の主要な構成要素である粗粒度タスク並列処理において、ヘテロジニアスマルチコア上の各コアの特性を考慮してタスクの割当てを行うスタティックスケジューリング手法および本手法を用いた MP3 エンコーダの並列化について述べる。

以下、2章で本手法が対象とするヘテロジニアスマルチコアアーキテクチャについて、3章でヘテロジニアスマルチコア向けのコンパイル方式について、4章でヘテロジニアスマルチコア上での粗粒度タスク並列処理について、5章でヘテロジニアスマルチコアアーキテクチャを対象とした粗粒度タスクのスタティックスケジューリングアルゴリズムについて、

6章で MP3 エンコーダを用いた並列性能評価についてそれぞれ述べる。

2. ヘテロジニアスマルチコアアーキテクチャ

本章では、汎用コアに加えて動的再構成可能プロセッサ (DRP) や DSP 等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プロセッサアーキテクチャについて述べる。

本手法で対象とするヘテロジニアスマルチコアプロセッサは、図 1 に示すような、各々がローカルデータメモリ (LDM) および分散共有メモリ (DSM) を持つ汎用 CPU コアあるいは DRP 等のアクセラレータコアを複数バスやクロスバ等の結合網で集中共有メモリ (CSM) へ接続した、OSCAR 型メモリアーキテクチャ^{12)–14)} を持つものである。

各プロセッサエレメント (PE) は、汎用プロセッサコアまたはアクセラレータコア、ローカルデータメモリ (LDM)、2 ポート構成の分散共有メモリ (DSM) およびデータ転送ユニット (DTU) を持つ。

LDM はアプリケーションプログラム中のプロセッサプライベートデータを格納する。DSM は 2 ポート構成となっており、自 PE と他 PE が同時にアクセス可能である。DSM は、タスク間のデータ・同期フラグの授受に利用される。DTU は高機能 DMA コントローラで、プロセッサの処理と非同期に PE 間、PE-CSM 間のデータ転送を行うことが可能であり、

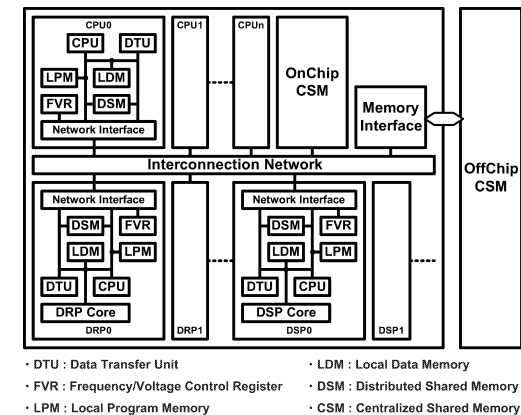


図 1 OSCAR 型メモリアーキテクチャを持つヘテロジニアスマルチコアプロセッサの例
Fig. 1 An example of a heterogeneous multicore with OSCAR-Type memory architecture.

データ転送とタスク処理のオーバーラップのために利用される。ローカルプログラムメモリは各 PE が実行するプログラムコードを格納するために用いられる。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、その他一定の計算処理が可能なコントローラを持つ。このコントローラとアクセラレータコアとの連携により、アクセラレータ PE においても分岐等を含む比較的複雑な処理の実行が可能となる。また、アクセラレータ PE にも汎用コア PE と同様の LDM と DSM を持たせることにより、データの分割配置やデータ転送を汎用コア PE と同様に行うことができる。

3. ヘテロジニアスマルチコア向け並列化コンパイル方式

本章では、ヘテロジニアスマルチコアアーキテクチャを対象とする並列化コンパイル方式について述べる。

ヘテロジニアスマルチコアにおいては、様々なベンダの様々な種類のアクセラレータコアが混載される可能性があり、またその組合せも用途によって様々となる。そのため、市場の様々なアクセラレータコアに対応した専用目的コンパイラ（実行可能タスクの抽出、コスト推定、アクセラレータコア用のコード生成等を行う）を開発するのは、限られた時間とコストでは現実的ではないと考えられる。そのため、本手法では各アクセラレータコア用に専用目的コンパイラが開発された場合に、それをあわせて利用できるようにしている。

図 2 にヘテロジニアスマルチコア向けのコンパイルフローを示す。図 2 では、対象となる逐次プログラムのソースファイルを、まず使用するアクセラレータコア用の専用コンパイラに入力する。アクセラレータコア用の専用コンパイラは入力されたソースファイルに対して解析を行い、実行あるいは高速化可能なタスクを抽出し、それに対して実行コストの推定および実行コードの生成を行う。また同時に、入力されたソースファイルに対してディレクティブを挿入することによりアクセラレータ PE で実行可能なタスクの指定を行う。あるいは、アクセラレータコア用のコンパイラの解析結果を基に手動でディレクティブを挿入する。

本論文で対象とする自動並列化コンパイラは、このディレクティブが挿入されたソースファイルを入力とし、それに対して並列化およびアクセラレータの利用を考慮した最適化を行うものである。この際、ターゲットとなるマルチコアの構成はオプションあるいは入力ファイルとして並列化コンパイラに入力される。並列化コンパイラにより並列化およびタスクスケジューリング等の最適化が施された後、アクセラレータ用コンパイラにより生成されたアクセラレータコア用の実行コードを利用して対象マルチコア向けの実行コードが生成

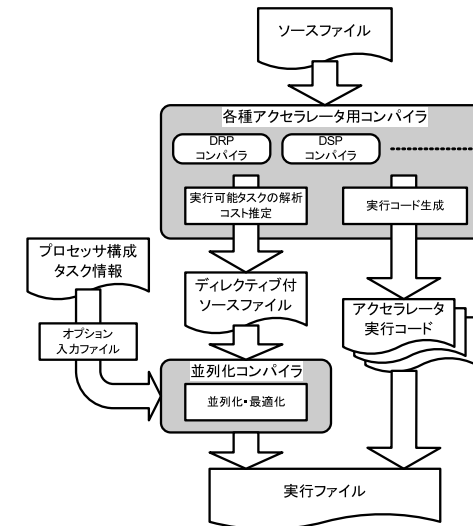


図 2 ヘテロジニアスマルチコア向けコンパイルフロー
Fig. 2 Compile flow for a heterogeneous multicore.

される。

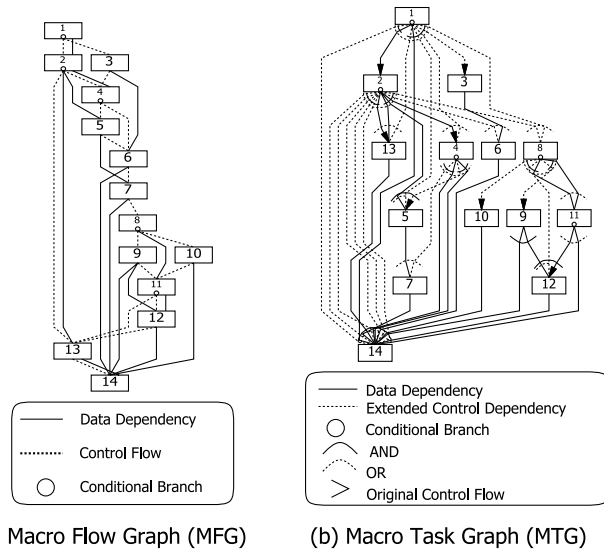
このように、各アクセラレータ用コンパイラと並列化コンパイラの間で情報をやりとりするためのインターフェースを用意することで、様々な構成のヘテロジニアスマルチコアへの対応が容易に可能となる自動並列化コンパイラを実現することができる。

4. ヘテロジニアスマルチコア上で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アーキテクチャ上での階層的粗粒度タスク並列処理について述べる。

4.1 粗粒度タスク並列処理

粗粒度タスク並列処理では、対象とするプログラムはまず擬似代入文ブロック (BPA)、繰返しブロック (RB)、サブルーチンブロック (SB) の 3 種類の粗粒度タスク (マクロタスク, MT) に分割される。MT 生成後、MT 間のコントロールフローおよびデータ依存を解析し、図 3 (a) に示すようなマクロフローグラフ (MFG) が生成される。さらに最早実行可能条件解析によって MFG から MT 間の並列性を抽出して図 3 (b) に示すようなマク



(a) Macro Flow Graph (MFG) (b) Macro Task Graph (MTG)

図3 マクロフローグラフ (MFG), マクロタスクグラフ (MTG) の例
Fig. 3 Examples of a Macro-Flow Graph (MFG) and a Macro-Task Graph (MTG).

ロタスクグラフ (MTG) を生成する^{15),16)}。この際, RB および SB はネスト構造を持つ場合があり, その場合には内部ボディ部に対して階層的に MT および MTG の生成を行う。

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) をグループとしたプロセッサグループ (PG) に割り当てられる。このとき, MTG 内に条件分岐等がなければ, 静的に MT を PG に割り当てるスタティックスケジューリング手法が適用される。本論文ではこのスタティックスケジューリング手法を対象としており, プロセッサ間のデータ転送や同期, 実行時のタスク割当て等のオーバーヘッドを最小化することができる。また, 本論文では取り扱わないが, MTG に条件分岐等による実行時不確実性が存在する場合には, MT のコードに加えて MT の最早実行可能条件を管理しつつ MT を PG に割り当てるためのスケジューラのコードもあわせて生成し, 実行時に MT を PG に割り当てるようにするダイナミックスケジューリング手法が適用される。

さらに, SB や RB 内部の MTG に粗粒度並列性が存在する場合, その内部で階層的に PE をグルーピングし, 階層的に粗粒度タスク並列処理を適用する。粗粒度並列性を利用する階層および並列性に応じたプロセッサグルーピングの決定には, 並列処理階層自動決定手

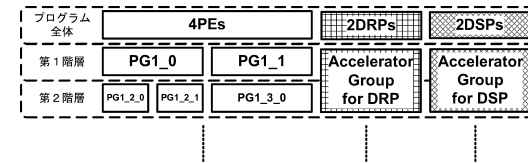


図4 ヘテロジニアスマルチコアを考慮した階層的なプロセッサグルーピングの例
Fig. 4 An example of hierarchical processor grouping for a heterogeneous multicore.

法^{17),18)} が用いられる。

4.2 ヘテロジニアスマルチコアを考慮したプロセッサグルーピング

4.1 節で述べたように, 粗粒度タスク並列処理では PE をグルーピングした PG が MT の割当ておよび実行単位として扱われ, プログラムの階層構造および並列性によって階層的なプロセッサグルーピングが行われる。しかし, ヘテロジニアスマルチコアを対象とした場合, 汎用コア PE ではすべての種類の MT を実行することが可能であるのに対し, アクセラレータ PE では特定の処理を非常に高速化することが可能だが, 実行可能な MT が限られてしまう。そのため本手法では, 汎用コア PE のみの場合と異なり, アクセラレータ PE の利用を考慮したプロセッサのグルーピングを行う。

本手法においては, 汎用コア PE は各 MTG の粗粒度並列性と汎用コア PE の数を考慮した階層的なグルーピングを行うが, 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われ, あらゆる階層の MTG から適宜必要に応じて MT が割り当てられるものとする。これにより, アクセラレータで実行可能な MT がどの階層の MTG にどのように分散しているにかかわらず, すべての階層の MTG からアクセラレータを有効に利用することができる。

プロセッサのグルーピングの例を図4に示す。図4は, 汎用コア4基, DRP および DSP をそれぞれ2基ずつ持つヘテロジニアスマルチコアの例である。図4では, 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 および PG1_1) にグルーピングされており, 各々のPGにこの階層のMTG中の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。ある時刻において, PG1_0に内部に粗粒度並列性を持つMTが割り当てられた場合, PG1_0はさらに第2階層で1プロセッサずつの2つのPG (PG1_2_0 および PG1_2_1) にグルーピングされる。このように, 汎用コアはプログラムの並列性に依って階層的にグルーピングされる。それに対し, 図中のDRP および DSPは汎用コアの階層グルーピングと異なり, 同一機能を持つアクセラレータごとにグループ

を生成しているため、あらゆる階層の MTG からのタスク処理依頼を受け付けることができる。

5. ヘテロジニアスマルチコア用タスクスケジューリングアルゴリズム

本章では、2 章で述べたヘテロジニアスチップマルチプロセッサを対象とした粗粒度タスクスタティックスケジューリング手法について述べる。

ヘテロジニアスマルチコアプロセッサでは、汎用コア PE ではすべての種類のマクロタスク (MT) が実行可能であるが、アクセラレータ PE ではその種別により実行が可能な MT の種類が限られてしまう。また、同じ MT であっても、どの種別の PE で実行されるかによって処理にかかる時間が異なる。一般に、アクセラレータ PE で実行可能な MT はそのままアクセラレータ PE に割り当てて実行した方がその MT 自体の処理効率を向上させることができるが、負荷集中時もそのままアクセラレータ PE に MT を割り当ててしまえば、逆にプログラム全体の実行時間が伸びてしまう場合が考えられる。本アルゴリズムは、このような場合には各種アクセラレータ PE と汎用コア PE を効率良く利用し、プログラム全体の処理効率の向上を図るものである。本スケジューリング手法は、リストスケジューリングにおける優先度およびタスク割当ての方針をヘテロジニアスマルチコアプロセッサ用に拡張したものである。

5.1 グローバル CP 長

4.2 節で述べたように、本手法では異なる階層のマクロタスクグラフ (MTG) 中の MT を必要に応じてアクセラレータ PE に割り当てる。そのため、実行可能タスクのアクセラレータ PE への割当て時に優先度を比較する際、すべての MTG を通じて共通の優先度を定義する必要がある。本手法では、メインプログラムの出口ノードから各 MTG 中のタスクまでの最長のパス長をグローバル CP 長として定義し、これを割当ての優先度として採用した。グローバル CP 長の算出例を図 5 に示す。

図 5 において、MT3 の内側の MTG に含まれる MT3-3 のグローバル CP 長を求めることを考える。まず、プログラム全体の出口ノードから、内部に MTG を持つ MT3 の末尾までの最長パス長を求める。ここでは、MT4 を経由するパスが最も長く、パス長は 70 となっている。MT3 は最も上位の MTG に属するので、これが MT3 の末尾までのグローバル CP 長となる。次に、MT3-3 の MTG 内 CP 長は 40 であることが分かる。よって、MT3-3 に対するグローバル CP 長は、MTG 内での CP 長 40 に、MTG を内包する MT3 の末尾までのグローバル CP 長 70 を加えて 110 となる。また、仮に MT3 が RB であった場合には、

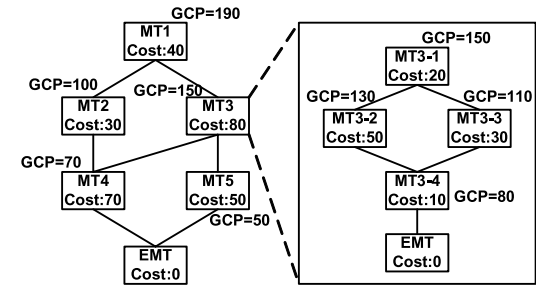


図 5 グローバル CP 長
Fig. 5 Global CP length.

MT3 内部の MTG の出口ノードから入り口ノードまでの CP 長に (RB の回転数 - 1) を乗じたものをさらに加える。そうすることで、ループの回転数を考慮した CP 長を算出することができる。なお、「最長」パス長という定義から、上記の計算においては各 MT のコストは汎用コア PE 上で実行したものが用いられている。

5.2 処理およびデータ転送コストを考慮したタスクスケジューリングアルゴリズム

本手法では、MT を割り当てる対象を、汎用コア PE の集合であるプロセッサグループ (PG) またはアクセラレータ PE とする。以下に、ヘテロジニアスマルチコアにおけるスタティックスケジューリングアルゴリズムの基本フローを示す。

- step 1: 各 MTG 上の MT に対し、実行可能なコアそれぞれに対するタスク処理コストの計算あるいは取得を行う。また割当て優先度であるグローバル CP 長を計算する。
- step 2: スケジューリング時刻を 0 にセットする。また、最上位 MTG をスケジューリング処理中の MTG 一覧に追加する。ここで、スケジューリング時刻とは、リストスケジューリングにおいて次の割当てを決定する時刻のことをいう。
- step 3: スケジューリング処理中の MTG に含まれる MT の中から最早実行可能条件を満たした実行可能 MT (レディタスク) を検出する。
- step 4: 現スケジューリング時刻において IDLE な汎用コア PG あるいは IDLE かつその MT を実行可能なアクセラレータ PE を持つレディタスクが存在すれば、そのうち最も優先度の高いものを割当て対象として選択する。存在しなければ、step 8 へ。
- step 5: 対象 MT を割当て可能な PG あるいはアクセラレータ PE に対して、対象 MT の終了時刻を以下の式により推定する。ここで、「割当て可能」とは、汎用コア PG が当該時刻において IDLE であるか、あるいはアクセラレータ PE がその MT を実行可

能であることをいう。

当該 MT の先行タスク集合を P 、先行タスク MT_p からの DTU を用いたデータ転送終了時刻を $DT_{send}(p)$ 、CSM からの DTU を用いたデータ転送終了時刻を DT_{load} とすると、対象 MT の実行に必要なデータ転送の終了時刻 DT_{end} は

$$DT_{end} = \max[DT_{load}, \max_{MT_p \in P} \{DT_{send}(p)\}]$$

となる。なお、これらのデータ転送は、後述の 5.4 節のとおり、データ転送とタスク処理のオーバーラップを考慮してタイミングが決定される。

さらに、当該コアが IDLE となる時刻を T_{idle} 、当該コア上での対象 MT の実行コストを $COST_{MT}$ 、CSM への DTU を用いたデータ転送コストを DT_{store} とすると、終了予測時刻 MT_{fin} は

$$MT_{fin} = \max(T_{idle}, DT_{end}) + COST_{MT} + DT_{store}$$

と計算される。ただし、割当て対象 MT が内部にスケジューリング対象の MTG を内包する場合は、内部 MTG のスケジューリングを考慮した終了予測時刻を推定する。

step 6: 割当て可能なコアのうち、最も終了予測時刻の早い汎用コア PG あるいはアクセラレータ PE に対象 MT を割り当て、その際のデータ転送タイミングを採用する。

step 7: MT を割り当てた汎用コア PG またはアクセラレータ PE が次に IDLE になる時刻等の情報を更新し、step 4 へ。なお、割り当てた MT が内部に MTG を内包していれば、その MTG をスケジューリング処理中の MTG 一覧に追加する。

step 8: すべての MT が割当ておよび実行終了済みであれば終了、そうでなければ、次に最も早く MT 実行を終了する汎用コア PG あるいはアクセラレータ PE の MT 実行終了時刻を次のスケジューリング時刻とする。

step 9: 当該スケジューリング時刻において実行を終了する MT があれば、それを終了した MT の一覧に加え、step 3 へ。また、内部にスケジューリング対象の MTG を含む MT に関して、内部 MTG に含まれる MT の実行がすべて終了していれば、これに関しても終了時刻を確定し、終了した MT の一覧に追加する。

以上のようにして、汎用コア PG とアクセラレータ PE に対してそれぞれの負荷を考慮して適切に処理を割り当てていく。

5.3 階層的な MTG のスケジューリング

5.2 節のスケジューリングアルゴリズムにおいて、内部にスケジューリング対象の MTG を含む MT、つまり階層的な並列性を持つ MT が割当て対象として選択された場合、その

内部のスケジューリングを考慮したうえで割当て先を決定する必要がある。このような場合、本手法ではその MTG を最上位の MTG として再帰的に上記のスケジューリングアルゴリズムを適用する。これにより内部を事前に 1 度仮スケジューリングして終了予測時刻を算出し、MTG を内包する MT の割当て先を決定する。その後、その内部の MTG はスケジューリング処理中の MTG に登録され、改めて内部の MT が他の MTG と平行してスケジューリングされる。

ただし、当該 MT が繰返しブロック (RB) であった場合、その繰返し構造のため他の MTG と同期してスケジューリングしていくことは難しい。よって本手法では、RB 内の MTG に関しては他の MTG からは独立して扱う。具体的には、内部にスケジューリング対象の MTG を内包する RB を割り当てる際、当該スケジューリング時刻において IDLE 状態となっているアクセラレータ PE の中から必要なアクセラレータ PE を確保し、RB の実行が終了するまでの間占有して利用する方式をとる。このとき、RB が多重にネストされている場合には、外側の RB によって占有されているアクセラレータ PE の中から割り当てるアクセラレータ PE を決定する。ただし、最外側にある MTG 等、すべての汎用コア PE が参加するような RB の場合は、例外としてすべてのアクセラレータを占有して利用する。

RB によるアクセラレータ PE 占有の例を図 6 に示す。図 6 は汎用コア PE (CPU) 2 基およびアクセラレータ PE (DRP) を 1 基搭載するヘテロジニアスマルチコアに対するスケジューリング例である。なお、図中の MT をグローバル CP 長の大きいものから列挙すると、 $\{MT1.1, MT1.2, MT1.3, MT1.5, MT1.5.1, MT1.5.4, MT1.5.2, MT1.5.3, MT1.4, MT1.5.5, MT1.5.6\}$ となるものとする。

図中に定義される階層的な MTG のうち、最上位の MTG である MTG1 は粗粒度タスク並列性が 2 程度あると判断されるので、汎用コア PE は 2 つの PG にグルーピングされる。この階層においては、まず MT1.1 が CPU0 に割り当てられる。次にレディタスクが発生するのは MT1.1 の終了時であり、レディタスクは MT1.2 と MT1.3 である。ここで、MT1.2 は汎用コア PE のみで実行可能なタスクであるので、CPU0 に割り当てられ、MT1.3 は DRP 上で実行可能なので DRP0 に割り当てられる。次にレディタスクが発生するのは MT1.3 の終了時であり、レディタスクは MT1.5 である。MT1.5 はこのとき IDLE となっている CPU1 に割り当てられるが、MT1.5 は内部にスケジューリング対象の MTG (MTG1.5) を含み、MTG1.5 はアクセラレータ PE 向け MT を持つ。

ここで、MT1.5 は RB であるため、RB の開始から終了までの間、アクセラレータを占有利用することを考える。図では、この時点で IDLE である DRP0 が MT1.5 によって占

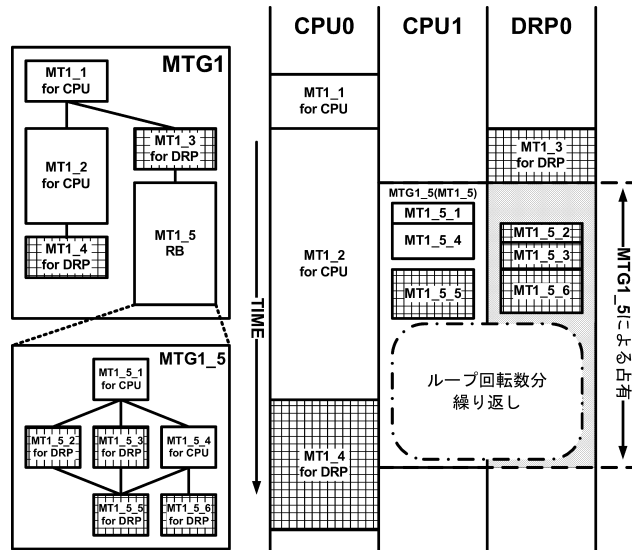


図 6 RB によるアクセラレータ PE の占有
Fig. 6 Use of accelerators by RB.

有される。またこのとき、MTG1 に対するスケジューリング処理は保留され、MTG1.5 内部のみを考慮したスケジューリング処理が先に実行される。MTG1.5 内部の MT を CPU1 と DRP0 に対してスケジューリングし、1 イタレーション分の割当てが決定した後、RB の回転数を考慮して MTG1.5 の終了時刻が算出される。このようにして MTG1.5 内部のスケジューリングが行われ、MT1.5 の終了時刻が決定したうえで、改めて MTG1 を含めたスケジューリングが再開される。

以上のようにして、RB 内部に MTG を持つような階層的 MTG に対するスケジューリングが行われる。

5.4 データ転送コストおよび転送タイミングの決定

本手法では、配列データは DTU¹⁹⁾ を用いて転送を行う。DTU の駆動は MT の開始あるいは終了のタイミングで行い、1 回の駆動で複数の DTU 命令からなる複数の DTU 転送を駆動できる。また、転送タイミングの決定は PE 間ネットワークの状態を考慮して決定される。今回の実装では、MTG をまたいだデータ転送は行わないものとし、転送対象となるデータは DSM 上に配置する。

ある MT を割り当て実行する際に必要となるデータの DTU を用いたデータ転送タイミングの決定²⁰⁾ においては、

1. 先行タスクからの転送に含まれず、MTG の開始時点で CSM から読み出すことが可能な配列データ
2. 異なるコアに割り当てられた先行タスクとの間のフロー依存に含まれる配列データ
3. 転送範囲の解析結果が不定となった際、整合性確保に必要なデータに対する CSM からの読み出しおよび CSM への書き戻し
4. MTG の終了時まで CSM に書き戻せばよいデータの 4 種類のデータ転送パターン各々に対し、以下のように転送を行う。

1. に関しては、MTG の開始時刻から現在時刻までの間で転送可能な最早のタイミングを探索し、見つければ転送を挿入し、転送の終了時刻を計算する。もし現在時刻までに転送のタイミングが見つからなければ、現在時刻以降に転送を開始するものとして転送の挿入と終了時刻の計算を行う。2. に関しては、先行タスクの終了時刻から現在時刻までの間で同様に転送可能なタイミングを探索する。これに関しても、現在時刻までに見つからなければ現在時刻以降を開始時刻とするデータ転送がスケジューリングされる。3. に関しては、MT の実行直前および直後に転送するものとした。4. に関しては、当該 MTG 内部の MT のスケジューリングおよびそれらに対する 1. から 3. までの転送タイミングが決定した後、当該 MT の終了時刻以降で転送可能なタイミングを探索する。

図 7 にデータ転送の挿入例を示す。図 7 は汎用コア PE (CPU) およびアクセラレータ PE (DRP) をそれぞれ 1 基、PE 間相互接続網としてバス 1 本を持つマルチコアを想定したものである。簡単のため、図 7 は転送パターン 3. に属するデータ転送が含まれていない例となっている。また、説明を容易にするため、以下の記述は MT の割当て先の決定後に転送を挿入する形になっているが、実際の割当て処理においては、5.2 節のアルゴリズムのとおり、転送を考慮したうえで MT の割当て先を決定する。つまり、MT の割当て先を決定する際に、当該 MT を割当て可能なコアごとに転送の挿入の状態および MT の終了時刻を考慮し、最も MT の終了時刻が早いものを MT の割当て先および転送の挿入タイミングとして採用する。

図 7 において、スケジューリング対象の MTG が定義されている。この MTG 内の MT を、グローバル CP 長の大きい順に列挙すると、{MT1, MT2, MT3, MT4, MT5, MT7, MT6, MT8} となっているものとする。

まず、MTG 開始時点でレディタスクとなっている MT1 が CPU0 に割り当てられる。

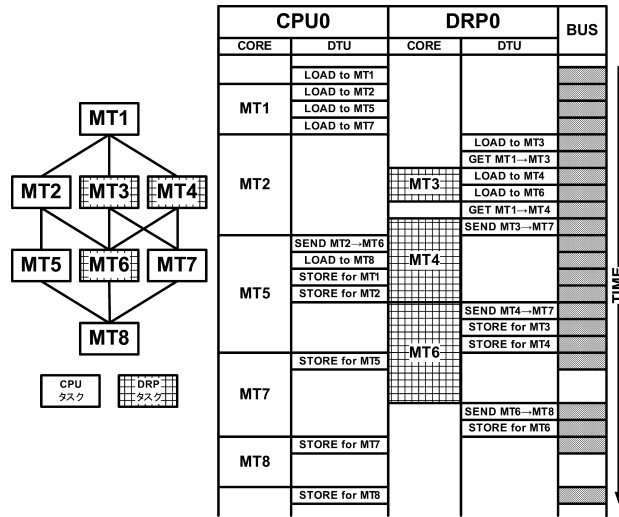


図 7 データ転送挿入例

Fig. 7 Overlapping of data transfer and task execution.

MT1 に対しては、開始時に CSM からのデータ読み出しが挿入される。次にレディタスクが発生するのは MT1 の終了時であり、レディタスクは MT2, MT3 および MT4 である。このうち割当て優先度の最も高い MT2 が割当て対象として選択される。ここで、CSM からの読み出しは MT1 開始時の転送後に続けて挿入することが可能なので、MT1 の開始時の転送と連続して行われるようにスケジューリングされる。このとき、MT1 は MT1 の実行に必要なデータの転送が終わった時点で実行が開始され、MT1 の実行とオーバーラップして MT2 に対する転送が行われる。また、MT2 は MT1 と同じ CPU0 に割り当てられるので MT1 からの転送は必要ない。次に、優先度を比較した結果 MT3 が割当て対象の MT として選択される。MT3 に関しては、これより先に DRP0 で実行される MT が存在しないため、直前に CSM からの読み出しの転送が行われる。また、異なるコアに割り当てられている MT1 からの転送は、MT1 の終了時刻以降でタイミングを探すことになるが、バスの使用状況を考慮した結果、CPU0 側から DRP0 に対する転送を行うタイミングがないため、DRP0 が MT3 の開始時に CPU0 の DSM からデータを読み出す形で転送が挿入される。次にレディタスクと割当て可能コアの組合せが発生するのは MT3 の終了時であり、MT4 が DRP0 に割り当てられる。このとき、CSM からの読み出しに関しては、MT3 開始時の転

送後に連続して行うことが可能であるため、ここに転送が挿入される。また、MT1 からのフロー依存範囲のデータは、MT1 の終了後のタイミングではバスが他の転送によって利用されているため、MT4 の開始直前に、DRP0 が CPU0 の DSM から読み出す形で転送が割り当てられる。以下同様にして、PE 間ネットワークの状況を考慮しながら、可能な限りデータ転送が MT の実行に隠蔽されるようにして MT およびデータ転送のスケジューリングを行う。

このようにして CSM からの読み出しおよびコア間の転送を考慮した MT のスケジューリングが行われた後、最後に CSM への書き戻しの転送のタイミングを決定する。まず MT1 および MT2 に関しては、ネットワークの状態を考慮した結果、MT2 の終了時の転送と連続する形で挿入される。MT5, MT7, MT8 に関しては、それぞれの MT の終了時に転送が挿入される。MT3 と MT4 に関しては、同様にネットワークの状態を考慮して MT4 終了時の転送と連続する形で CSM への書き戻しが行われる。MT6 に関しては、MT6 から MT8 への転送と連続して CSM への転送が割り当てられる。

6. 性能評価

本章では、提案スケジューリング手法の性能評価について述べる。

6.1 評価アーキテクチャ

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 8 つまで搭載するシステムを想定し、アクセラレータ PE 内のアクセラレータコアとして動的再構成可能プロセッサ (DRP) を用いるものとした。なお、本評価では、汎用プロセッサコアおよびアクセラレータ PE 内のコントローラを株式会社ルネサステクノロジーの SH4A プロセッサ²¹⁾ とし、動的再構成可能プロセッサコアを株式会社日立製作所の FE-GA²²⁾ とした。

また、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等については、SH4A を 300 MHz で動作させることを想定し、表 1 に示すように、自 PE 内部の分散共有メモリ (DSM) およびローカルデータメモリ (LDM) へのアクセスに 1 クロック、他 PE の持つ分散共有メモリへのアクセスに 4 クロック、集中共有メモリ (CSM) へのアクセスにチップ外を想定したときに 16 クロック、チップ内を想定したときに 4 クロックかかるものとした。なお、ヘテロジニアスマルチコアへのスケジューリングアルゴリズムの有効性を検証するため、ローカルメモリ (LDM および DSM) に関しては、評価のベースとなるプロセッサ 1 台の際に全データがローカルメモリに格納可能となるサイズを想定している。

表 1 各メモリのアクセスコスト
Table 1 Memory access costs.

分散共有メモリ (自 PE)	1 クロック
分散共有メモリ (他 PE)	4 クロック
ローカルデータメモリ	1 クロック
集中共有メモリ	16 クロック (オフチップ時) 4 クロック (オンチップ時)

評価には上記の構成を持つマルチコアプロセッサをクロックレベルで精密に再現するシミュレータを用いた。

6.2 評価アプリケーション

本評価に用いるアプリケーションとしては、オーディオ圧縮方式として広く利用されている MP3 エンコーダを選択した。精密なシミュレーションを行うためには、アプリケーションのどの部分をアクセラレータで高速実行できるか、またその動作クロック数がどうか等の情報が必要である。しかしながら、本論文のターゲットが図 2 における並列化コンパイラ部分であることと、本論文執筆時点でアクセラレータ用の専用目的コンパイラが入手できないことから、本評価においては、アクセラレータコアで高速化および実行が可能な処理の抽出、FE-GA のセルアレイへのマッピング、コスト推定およびアクセラレータ PE へ割当て可能なタスクの選定を手動で行った。アクセラレータ PE へ割当て可能なタスクおよびその実行コストは、並列化コンパイラへの入力となるソースプログラム中で指示文を用いて指定した。この作業は多大なコストと期間を要するため、今後様々な種類のアプリケーションやアクセラレータコアを用いた評価を行っていくには、アクセラレータ用の専用コンパイラを利用できることが必要となる。

なお、本評価に用いるプログラムは、UZURA MPEG1/LayerIII encoder in FORTRAN90²³⁾ を参照実装し、Fortran77 で実装されたプログラムである。参照実装したシーケンシャルプログラムを、5 章で述べたスケジューリングアルゴリズムを実装した OSCAR 自動並列化コンパイラ²⁴⁾ を用いてコンパイルすることで並列性の抽出および粗粒度タスクのスケジューリングを行った。ただし、通常オプションとしてあたえられるパラメータ等を定数として表記したり、フレーム間の並列性²⁵⁾、粗粒度並列性の抽出が容易になるようあらかじめループのアンローリング等を適用する等の変更を行った。

今回の評価では、エンコーディングの入力データはステレオの PCM データ 16 フレーム、エンコーディングのパラメータはサンプルレート 44.1 [KHz]、ビットレート 128 [kbps] で

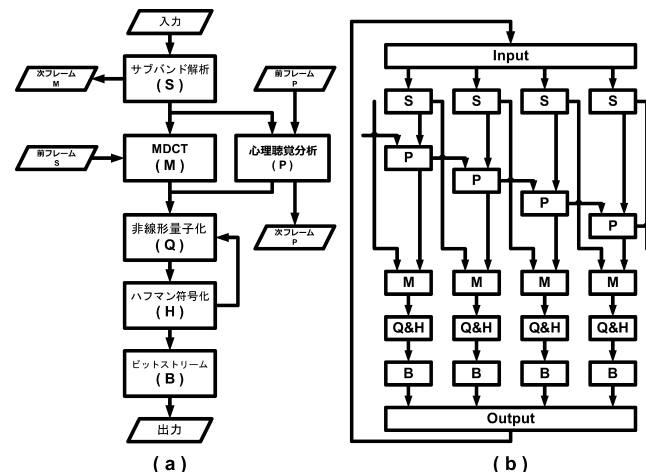


図 8 MP3 エンコーダの処理フローとフレーム間並列性利用を考慮したプログラム構造の概要
Fig. 8 Program structure and parallelism of MP3 Encoder.

ある。入力の PCM データがすべて CSM 上に格納されている状態から符号化後のデータが CSM に書き込まれるまでを評価対象とし、初期値計算やエンコード結果確認のための出力部分は評価対象から除外した。

また、スケジューリングの際に利用する各 MT のコストとして、入力となる 16 フレームに対してプロファイルを取得し、その平均値を用いた。これは、スケジューリング手法の有効性を確認するためと、アクセラレータコアによる実行コストが実際にセルアレイに処理をマッピングした結果から算出されているためである。スケジューリング時に用いるデータ転送コストに関しては、メモリアクセスコストおよび転送対象データ量からコンパイラ内部で計算される値を用いた。

6.2.1 MP3 エンコーダの構造と並列性

MP3 の処理の流れおよびフレーム間並列性が利用可能となるようにリストラクチャリングを施した場合のプログラム構造の概要を図 8 に示す。図 8(a) に示すとおり、MP3 エンコード処理は、サブバンド解析、MDCT、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からなり、MDCT は前フレームのサブバンド解析の結果を利用し、心理聴覚分析では、前フレームの心理聴覚分析の結果を用いる。MDCT および心理聴覚分析においてフレーム間でデータの受け渡しを行う箇所以外では、複数のフレームに対する処

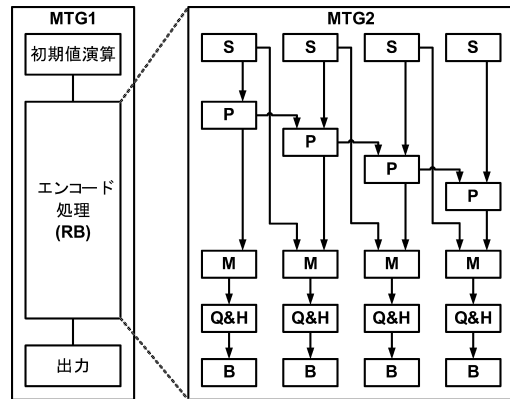


図 9 MP3 エンコーダの階層構造の概要 (4 フレーム並列実行の場合)
Fig. 9 Hierarchical structure of MP3 Encoder.

理を同時に行うことができ、フレーム間の並列性を利用して並列処理を行う手法が効果的である。よって本評価で用いたプログラムでは、図 8 (b) に示すような形でプログラム中のエンコード部分を記述した。図 8 (b) では、複数のフレームのデータを一度に読み込み、各フレームに対して同時に図 8 (a) の各処理を適用する形になっている。なお、本評価では、エンコードを行うメインループにおいて、1 イタレーションあたり 16 フレームに対する処理を並列に行う構造のプログラムを用いた。

また、図 9 に並列化コンパイラにより解析された MP3 エンコーダの階層構造の概要を示す。なお、図 9 はメインループ 1 イタレーションあたり 4 フレームに対して処理を行う場合の例となっている。図 9 では、まずメインプログラム階層 (MTG1) において「初期値演算」、「エンコード処理」、「出力」に対応した形で処理が記述されている。このうち、本評価で対象とするエンコード処理部分は入力フレーム数に応じて繰り返される繰り返しブロック (RB) であり、各フレームに対する処理間での並列性が大きいため、この RB のボディ部が MTG2 として定義され、粗粒度タスク並列処理の対象階層として選択されている。また、その並列性を生かすため、コンパイラ粗粒度タスクの割当て単位である汎用コア PG の数が最大となるように汎用コア PE のグルーピングが行われ、1 つの汎用コア PG は 1 つの汎用コア PE により構成される形となる。これにより、この階層では DTU を用いた MT 間のデータ転送を行うことができる。さらに、これらの階層では条件分岐等が存在しないため、MTG1, MTG2 とともにスタティックスケジューリング手法が適用される。

```
integer i, j, k
integer istart
real*8 tmp(512)
real*8 prototype(512)
real*8 pcm(1632, 2)
...
do i = 1, 36
  istart = 32 * (i-1) + 512
  do j = 1, 2
    do k = 1, 512
      tmp(k) = prototype(k)
      & * pcm(istart+1-k, j)
    end do
  end do
end do
...
```

図 10 FE-GA 演算セルアレイへのマッピング対象コード例
Fig. 10 Sample code accelerated by FE-GA.

6.2.2 アクセラレータ PE によるスピードアップ

本評価において、FE-GA をアクセラレータコアとして持つアクセラレータ PE で実行可能な処理²⁶⁾ は、サブバンド解析の一部 (フィルタ処理)、心理聴覚分析、MDCT、非線形量子化 + 符号化である。これらの処理のうち、FE-GA で高速化可能な演算処理に関しては、FE-GA 用の専用コンパイラが本論文執筆時点で入手できないため、実際に FE-GA の演算セルアレイに処理をマッピングした結果と入力データ量から処理クロック数を算出し、スケジューリングおよびシミュレーションのための実行コストとして利用した。

FE-GA は 32 個の汎用演算セル (算術論理演算セル 24 個、乗算セル 8 個) からなる二次元演算セルアレイを持ち、各セルの機能をソフトウェアで 1 クロックサイクルで変更可能である。また、このセルアレイはメモリアクセス制御専用セルとクロスバネットワークを介して接続されている。このような構造を持つ FE-GA の演算セルアレイに対して実際に手動で行った演算処理のマッピングの例を図 10 および図 11 に示す。図 10 はサブバンド解析 (フィルタ処理) のコードの一部を取り出したものであり、そのうち網掛けになっているループを実際に FE-GA の演算セルアレイにマッピングしたものが図 11 となっている。

図 11 では、メモリアクセス専用セル (LS) からクロスバ (XBAR) を介して入力データが供給され、中央のセルアレイ上で演算が行われた後、再度 XBAR を介して LS に出力される形になっている。図 11 は、図 10 の対象ループで扱われるデータを 32 bit の固定小数

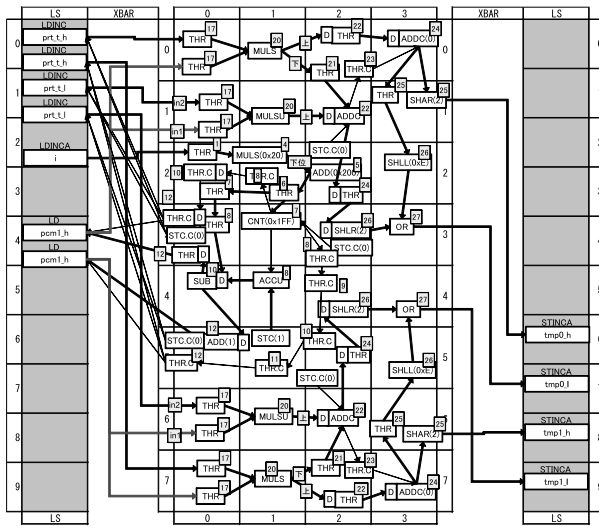


図 11 FE-GA 演算セルアレイへのマッピング例
 Fig. 11 An example of code mapping on FE-GA's ALU array.

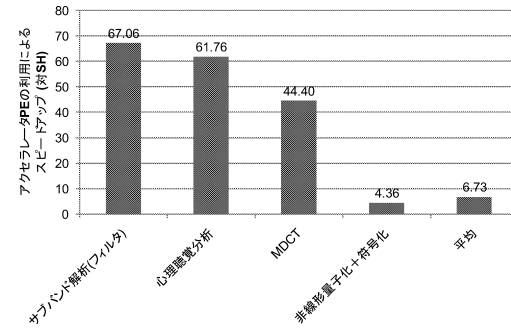


図 12 アクセラレータ PE によるスピードアップ
 Fig. 12 Speedup by using an accelerator-PE.

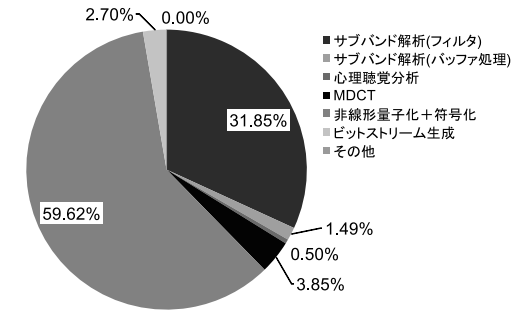


図 13 MP3 エンコーダ中の各処理の実行時間の割合
 Fig. 13 Profiling results of MP3 Encoder.

点に変換し、さらにセルアレイの有効利用のためにループを 2 分割したうえでマッピングを行ったものである。

なお、その他の分岐処理やメモリコピー等、アクセラレータコアによって高速化が難しい部分に関しては、アクセラレータ PE 内のコントローラによって処理されるものとした。図 12 に、アクセラレータ PE を用いた際の CPU に対する各処理のスピードアップを示す。

図 12 では、比較的単純な処理であるサブバンド解析(フィルタ)や MDCT、心理聴覚分析においてそれぞれ 67.06 倍、44.40 倍、61.76 倍と大きく高速化されている。非線形量子化および符号化は内部の制御フローが複雑であることと、アクセラレータ内のコントローラが担当する処理の割合が比較的大きいため他のタスクに比べてスピードアップは小さいが、それでも 4.36 倍となっている。アクセラレータ PE で実行可能なタスクの平均で、6.73 倍のスピードアップとなった。また、MP3 エンコーダにおける、各処理の汎用コア PE 上での実行時間の割合を図 13 に示す。MP3 エンコーダにおいて、アクセラレータ PE に割当て可能なタスクの処理時間割合は 95.81[%] となった。

6.3 評価結果

MP3 エンコーダを用いた評価結果を図 14 に示す。図 14 において、横軸はプロセッサおよび想定する CSM の構成を表し、縦軸は OnChip CSM 構成において汎用コア PE のみを用いて逐次処理を行った結果に対するスピードアップを表している。なお、図中の nCPU+mDRP とは、n 個の汎用コア PE と m 個のアクセラレータ PE を使用していることを示しており、OnChip CSM は集中共有メモリ(CSM)がチップ内にある(レイテンシ 4 クロック)場合を、OffChip CSM は CSM がチップ外にある(レイテンシ 16 クロック)場合を示している。なお、本評価で用いた MP3 エンコーダプログラムでは、16 フレームを同時にエンコードするために約 2 MB のデータを必要とする。よって、6.2 節で述べたように、

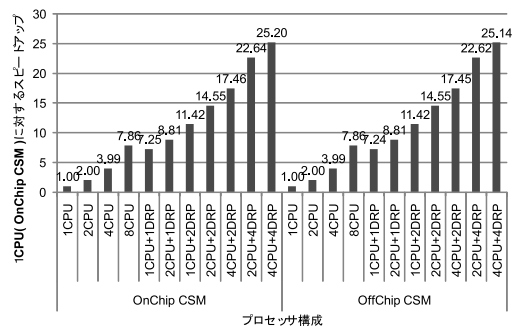


図 14 MP3 エンコーダを用いた評価結果

Fig. 14 Evaluation results using MP3 Encoder.

評価の基準となる 1CPU 時に全データをローカルメモリ (LDM あるいは DSM) へ転送し格納することが可能となるよう、1PE あたり 2MB 程度のローカルメモリを使用するという想定での評価となっている。

図 14 より、OnChip CSM の場合、1CPU の場合と比較して、1CPU+1DRP の構成で 7.25 倍、2CPU+1DRP で 8.81 倍、1CPU+2DRP で 11.42 倍、2CPU+2DRP で 14.55 倍、4CPU+2DRP で 17.46 倍、2CPU+4DRP で 22.64 倍、4CPU+4DRP で 25.20 倍のスピードアップが得られており、アクセラレータ PE を有効に利用することで、大幅に性能を向上させることができた。また、ホモジニアス構成の場合においても、2CPU の構成で 2.00 倍、4CPU で 3.99 倍、8CPU で 7.86 倍とスケラブルにスピードアップしている。これは、ホモジニアスな構成がヘテロジニアス構成の特殊な場合として指定でき、本手法が有効に働くことを示している。

また、OffChip CSM の場合でも、ほとんどのメモリアクセスが LDM (ローカルデータメモリ) あるいは DSM (分散共有メモリ) へのものとなり CSM (集中共有メモリ) へのアクセスを最小化できているため、OnChip CSM の 1CPU と比較して、2CPU の構成で 2.00 倍、4CPU で 3.99 倍、8CPU で 7.86 倍、1CPU+1DRP で 7.24 倍、2CPU+1DRP で 8.81 倍、1CPU+2DRP で 11.42 倍、2CPU+2DRP で 14.55 倍、4CPU+2DRP で 17.45 倍、2CPU+4DRP で 22.62 倍、4CPU+4DRP で 25.14 倍のスピードアップを得ることができ、ほとんど性能が低下することがないことが確かめられた。

図 15 に 2CPU+2DRP の構成で OffChip CSM を想定した場合の実行トレースを示す。図 15 において MT157 および MT158 は非線形量子化および符号化 (Q&H) の処理に対

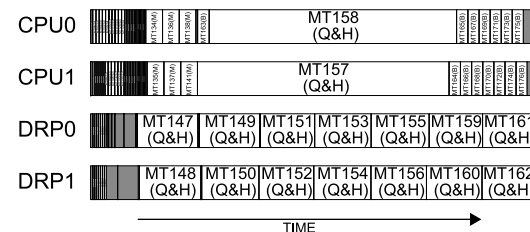


図 15 2CPU+2DRP (OffChip CSM) 時の実行トレース

Fig. 15 Execution trace using 2CPU+2DRP.

応しており、これらはアクセラレータ PE に割り当てることにより効率良く実行可能な MT である。しかし、この場合他のフレームの処理がすでにアクセラレータ PE に割り当てられており、さらにこれらをアクセラレータ PE に割り当てると逆に全体として性能が落ちてしまう。本手法を適用することでこのようなタスクは自動的に CPU に割り当てられ、CPU とアクセラレータ両方を効率良く利用することができる。

さらに、従来の研究で行われた手動による並列化およびスケジューリングを適用した場合の性能 (1CPU に対し 2CPU+2DRP で 14.6 倍、4CPU+4DRP で 21.1 倍²⁵⁾) と比較しても、同等あるいはそれ以上の性能を得ることができている。手動によるスケジューリングではデータ転送とタスク実行のオーバーラップを考慮しておらず、またアクセラレータ PE によるスピードアップを一律 10 倍としている等前提条件や精度は若干異なっているが、データ転送とタスク実行のオーバーラップを含め、従来実現されていないコンパイラによる自動並列化の実現により、様々な構成のマルチコアに対応した並列プログラムを自動生成することができるようになり、短い開発期間で同等以上の性能が得られることが確かめられた。

7. まとめ

本論文では、汎用 CPU に加え、動的再構成可能プロセッサ等のアクセラレータを複数 1 チップ上に集積したヘテロジニアスマルチコアプロセッサ上での粗粒度タスクスタティックスケジューリング手法を提案した。また、提案手法を自動並列化コンパイラに実装し、世界初のヘテロジニアスマルチコア向けの自動並列化を実現した。MP3 エンコーダを用いて行った性能評価では、汎用 CPU として SH4A コアを、アクセラレータコアとして DRP (FE-GA) を用いた場合、SH4A 1 コアを用いた逐次処理に対して、オンチップ集中共有メモリの場合に SH4A 4 コアの構成で 3.99 倍、SH4A 2 コアと DRP 2 コアの構成で 14.55

倍, SH4A 4 コアと DRP 4 コアの構成で 25.20 倍のスピードアップを得られることが確かめられた。また, オフチップ集中共有メモリの場合においても, SH4A 4 コアの構成で 3.99 倍, SH4A 2 コアと DRP 2 コアの構成で 14.55 倍, SH4A 4 コアと DRP 4 コアの構成で 25.14 倍のスピードアップを得られることが確かめられた。これは, 手動にてタスクのスケジューリングおよび評価を行った先行研究と比較しても同等以上の性能となっており, 並列化コンパイラによる自動化によりプログラムの並列化にかかるコストを大幅に削減できている。

今後の課題としては, NEDO “半導体アプリケーションチッププロジェクト「情報家電用ヘテロジニアス・マルチコア技術開発」” において開発される実チップへの適用, 実用化を考えている。

謝辞 本研究の一部は NEDO “先進ヘテロジニアスマルチプロセッサ研究開発”, 同 “半導体アプリケーションチッププロジェクト「情報家電用ヘテロジニアス・マルチコア技術開発」” の支援により行われた。本研究の遂行にあたり有益なご議論をいただいた, 株式会社日立製作所の小高俊彦フェロー, 内山邦男氏, 伊藤雅樹氏, 佐藤真琴氏に感謝いたします。

参 考 文 献

- 1) ARM: *ARM11 MPCore Processor Technical Reference Manual* (2005).
- 2) Suga, A. and Matsunami, K.: Introducing the FR 500 embedded microprocessor, *IEEE MICRO*, Vol.20, pp.21-27 (2000).
- 3) Pham, D., Asano, S., et al.: The Design and Implementation of a First-Generation CELL Processor, *Proc. IEEE International Solid-State Circuits Conference (ISSCC2005)* (2005).
- 4) Torii, S., Suzuki, S., Tomonaga, H., Tokue, T., Sakai, J., Suzuki, N., Murakami, K., Hiraga, T., Shigemoto, K., Tatebe, Y., Ohbuchi, E., Kayama, N., Edahiro, M., Kusano, T. and Nishi, N.: A 600 MIPS 120 mW 70 μ A Leakage Triple-CPU Mobile Application Processor Chip, *Proc. IEEE International Solid-State Circuits Conference (ISSCC2005)*, pp.136-589 (2005).
- 5) 須田礼仁: ヘテロ並列計算環境のためのタスクスケジューリング手法のサーベイ, 情報処理学会論文誌: コンピューティングシステム, Vol.47, No.SIG18, pp.92-114 (2006).
- 6) 小出 洋, 笠原博徳: メタスケジューリング—自動並列分散処理の試み. *bit*, Vol.33, No.4, pp.36-41 (2001).
- 7) Hagraas, T. and Janecek, J.: A high performance, low complexity algorithm for compile-time task scheduling in heterogeneous systems, *Proc. 18th International Parallel and Distributed Processing Symposium (IPDPS'04)* (2004).
- 8) 渡邊誠也: ヘテロジニアスな並列計算環境における最適な負荷割当, 電子情報通信学会論文誌 D-I, Vol.J88-D-I, No.11, pp.1688-1695 (2005).
- 9) Kumar, R., Tullsen, D.M., Ranganathan, P., Jouppi, N.P. and Farkas, K.I.: Single-ISA heterogeneous multi-core architectures for multithreaded workload performance, *Proc. 31st Annual International Symposium on Computer Architecture (ISCA'04)*, pp.64-75 (2004).
- 10) Hattori, T., Irita, T., Ito, M., Yamamoto, E., Kato, H., Sado, G., Yamada, T., Nishiyama, K., Yagi, H., Koike, T., Tsuchihashi, Y., Higashida, M., Asano, H., Hayashibara, I., Tatezawa, K., Shimazaki, Y., Morino, N., Hirose, K., Tamaki, S., Yoshioka, S., Tsuchihashi, R., Arai, N., Akiyama, T. and Ohno, K.: A Power Management Scheme Controlling 20 Power Domains for a Single-Chip Mobile Processor, *Proc. IEEE International Solid-State Circuits Conference (ISSCC2006)*, pp.2210-2219 (2006).
- 11) Kimura, K., Kodaka, T., Obata, M. and Kasahara, H.: Multigrain Parallel Processing on OSCAR CMP, *Proc. International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems (IWIA'03)* (2003).
- 12) 笠原博徳, 成田誠之助, 橋本 親: OSCAR (Optimally Scheduled Advanced Multiprocessor) のアーキテクチャ, 電子情報通信学会論文誌 D, Vol.J71-D, No.8 (1988).
- 13) 木村啓二, 尾形 航, 岡本雅巳, 笠原博徳: 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).
- 14) Kimura, K., Wada, Y., Nakano, H., Kodaka, T., Shirako, J., Ishizaka, K. and Kasahara, H.: Multigrain Parallel Processing on Compiler Cooperative Chip Multiprocessor, *Proc. 9th Workshop on Interaction between Compilers and Computer Architectures (INTERACT-9)* (2005).
- 15) 本多弘樹, 岩田雅彦, 笠原博徳: Fortran プログラム粗粒度タスク間の並列性検出法, 電子情報通信学会論文誌 D-I, Vol.J73-D-I, No.12, pp.951-960 (1990).
- 16) 笠原博徳, 合田憲人, 吉田明正, 岡本雅巳, 本多弘樹: Fortran マクロデータフロー処理のマクロタスク生成手法, 電子情報通信学会論文誌 D-I, Vol.J75-D-I, No.8, pp.511-525 (1992).
- 17) Obata, M., Shirako, J., Kaminaga, H., Ishizaka, K. and Kasahara, H.: Hierarchical Parallelism Control for Multigrain Parallel Processing, *Proc. 15th International Workshop on Languages and Compilers for Parallel Computing (LCPC2002)* (2002).
- 18) 小幡元樹, 白子 準, 神長浩気, 石坂一久, 笠原博徳: マルチグレイン並列処理のための階層的並列性制御手法, 情報処理学会論文誌, Vol.44, No.4 (2003).
- 19) Ito, M., Todaka, T., Tsunoda, T., Tanaka, H., Kodama, T., Shikano, H., Onouchi, M., Uchiyama, K., Odaka, T., Kamei, T., Nagahama, E., Kusaoke, M., Nitta, Y., Wada, Y., Kimura, K. and Kasahara, H.: Heterogeneous Multiprocessor on a Chip

Which Enables 54x AAC-LC Stereo Encoding, *Proc. 2007 IEEE Symposia on VLSI Technology and Circuits* (2007).

- 20) 宮本孝道, 中川正洋, 浅野尚一郎, 内藤陽介, 仁藤拓実, 中野啓史, 木村啓二, 笠原博徳: マルチコアプロセッサ上での粗粒度タスク並列処理におけるデータ転送オーバーラップ方式, *情報処理学会研究報告 ARC*, Vol.167, No.10 (2006).
- 21) Yoshida, Y., Kamei, T., Hayase, K., Shibahara, S., Nishii, O., Hattori, T., Hasegawa, A., Takada, M., Irie, N., Uchiyama, K., Odaka, T., Takada, K., Kimura, K. and Kasahara, H.: A 4320 MIPS Four-Processor Core SMP/AMP with Individually Managed Clock Frequency for Low Power Consumption, *Proc. IEEE International Solid-State Circuits Conference (ISSCC2007)* (2007).
- 22) Kodama, T., Tsunoda, T., Takada, M., Tanaka, H., Akita, Y., Sato, M. and Ito, M.: Flexible Engine: A Dynamic Reconfigurable Accelerator with High Performance and Low Power Consumption, *Proc. IEEE Symposium on Low-Power and High Speed Chips (COOL Chips IX)* (2006).
- 23) UZURA3: MPEG1/LayerIII encoder in FORTRAN90.
http://members.at.infoseek.co.jp/kitaurawa/index_e.html
- 24) 岡本雅巳, 合田憲人, 宮沢 稔, 本多弘樹, 笠原博徳: OSCAR マルチグレイコンパイラにおける階層型マクロデータフロー処理, *情報処理学会論文誌*, Vol.35, No.4, pp.513-521 (1994).
- 25) 鹿野裕明, 鈴木裕貴, 和田康孝, 白子 準, 木村啓二, 笠原博徳: MP3 エンコーダを用いた OSCAR ヘテロジニアスチップマルチプロセッサの性能評価, *情報処理学会論文誌: コンピューティングシステム*, Vol.48, No.SIG8 (2007).
- 26) 田中博志, 津野田賢伸, 秋田庸平, 高田雅士, 伊藤雅樹, 佐藤真琴: 再構成プロセッサ FE-GA のオーディオ処理への応用, *電子情報通信学会技術研究報告 RECONF2005-67*, Vol.105, No.451, pp.49-53 (2005).

(平成 19 年 10 月 9 日受付)

(平成 20 年 2 月 8 日採録)



和田 康孝 (正会員)

昭和 54 年生。平成 14 年早稲田大学工学部電気電子情報工学科卒業。平成 16 年同大学大学院理工学研究科電気工学専攻修士課程修了。平成 16 年同大学院理工学研究科情報・ネットワーク専攻博士課程入学。平成 18 年同大学工学部助手。平成 19 年同大学基幹理工学部助手, 現在に至る。



林 明宏

昭和 59 年生。平成 19 年早稲田大学工学部コンピュータネットワーク工学科卒業。平成 20 年同大学大学院修士課程修了。平成 20 年同大学院博士課程進学。現在に至る。



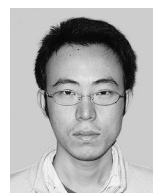
益浦 健

昭和 58 年生。平成 18 年早稲田大学工学部電気電子情報工学科卒業。平成 20 年同大学大学院理工学研究科情報ネットワーク専攻修了, ソニー(株)入社。



白子 準 (正会員)

昭和 54 年生。平成 14 年早稲田大学工学部電気電子情報工学科卒業。平成 19 年同大学大学院博士課程修了。博士(工学)。平成 17 年同大学同学部助手。平成 19 年学振特別研究員 PD。平成 20 年米国 Rice 大学ポスドクフェロー・早稲田大学アドバンスドチップマルチプロセッサ研究所客員講師。情報処理学会 SACSIS2006/IEEE Computer Society Japan Chapter 優秀若手研究賞受賞。IEEE Computer Society 正会員。



中野 啓史 (学生会員)

昭和 52 年生。平成 13 年早稲田大学工学部電気電子情報工学科卒業。平成 15 年同大学大学院修士課程修了。平成 15 年同大学院博士課程進学。現在, 同大学院博士課程在学中。



鹿野 裕明 (学生会員)

昭和 52 年生。平成 12 年中央大学工学部情報工学科卒業。平成 14 年同大学大学院修士課程修了。同年 (株) 日立製作所入社。平成 16 年より早稲田大学アドバンスチップマルチプロセッサ研究所客員研究員。平成 18 年早稲田大学大学院博士課程入学。平成 19 年情報処理学会山下記念研究賞受賞。現在、組み込みプロセッサに関する研究開発に従事。



木村 啓二 (正会員)

昭和 47 年生。平成 8 年早稲田大学工学部電機工学科卒業。平成 13 年同大学大学院理工学研究科電気工学専攻博士課程修了。平成 11 年同大学工学部助手。平成 16 年同大学工学部コンピュータ・ネットワーク工学科専任講師。平成 17 年同助教授。平成 19 年同大学情報理工学科准教授。現在に至る。マルチコアプロセッサのアーキテクチャとソフトウェアに関する研究に従事。



笠原 博徳 (正会員)

昭和 32 年生。昭和 55 年早稲田大学工学部電気工学科卒業。昭和 60 年同大学大学院博士課程修了，工学博士。昭和 58～60 年助手。昭和 60 年学振第 1 回特別研究員。昭和 61 年早稲田大学工学部専任講師。昭和 63 年助教授。平成 9 年教授。現在 CS 学科教授，アドバンスチップマルチプロセッサ研究所所長。昭和 60 年カリフォルニア大学バークレー，平成元年～2 年イリノイ大学 Center for Supercomputing R & D 客員研究員。昭和 62 年 IFAC World Congress 第 1 回 Young Author Prize，平成 9 年情報処理学会坂井記念特別賞，平成 16 年 STARC 共同研究賞受賞。主な著書：『並列処理技術』（コロナ社）。本会 ARC 主査，論文誌 HG 主査，会誌 HWG 主査，ACM ICS Program Vice Chair，IEEE CS Japan Chair，文科省地球シミュレータ中間評価委員，経産省/NEDO コンピュータ戦略 WG 委員長，“アドバンス並列化コンパイラ”，“リアルタイム情報家電用マルチコア”等プロジェクトリーダー等歴任。