

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

鹿野 裕 明^{†,††} 鈴木 裕 貴[†] 和田 康 孝[†]
白 子 準[†] 木 村 啓 二[†] 笠 原 博 徳[†]

汎用プロセッサや専用プロセッサ、アクセラレータ等、様々な種類のプロセッサエレメント(PE)を複数個同一チップ上に集積し、処理を並列で行うことで、動作周波数を過度に引き上げることなく、消費電力を抑えつつ高い演算性能を得ることが可能なソフトウェア協調ヘテロジニアスチップマルチプロセッサ(HCMP)の有効性を評価した。本プロセッサは各PEにローカルメモリ、データ転送機構および電力制御機構を持ち、コンパイラと協調することで性能向上と低電力の両立を狙う。今回、MP3エンコーダプログラムを利用し、その並列処理方式と電力制御方式を検討し、HCMP上での評価を実施した。その結果、汎用プロセッサコア2個、動的再構成プロセッサコア4個を搭載したHCMPは、汎用プロセッサ1個のシングルコアプロセッサと比較して18.4倍の性能向上となることが分かった。また、同様の構成に対し電力制御を実施することにより、汎用プロセッサ1個のシングルコアプロセッサに対し、電力消費エネルギーを最大80.0%削減できることが確認できた。

Performance Evaluation of MP3 Audio Encoder on OSCAR Heterogeneous Chip Multicore Processor

HIROAKI SHIKANO,^{†,††} YUKI SUZUKI,[†] YASUTAKA WADA,[†]
JUN SHIRAKO,[†] KEIJI KIMURA[†] and HIRONORI KASAHARA[†]

This paper evaluates a heterogeneous chip multi-processor (HCMP) and its scheduling scheme. The HCMP possesses different types of processing elements (PEs) such as CPUs as general-purpose processors, as well as digital signal processors or dynamic reconfigurable processors (DRPs) as specific-purpose processors. The HCMP realizes higher performance and lower power consumption than conventional single-core processors or even homogeneous multi-core processors in some specific applications such as media processing with low operating frequency supplied. In this paper, the performance of the HCMP is analyzed by studying parallelizing scheme and power control scheme of an MP3 audio encoding program and by scheduling the program onto the HCMP using these two schemes. As a result, it is observed an HCMP, consisting of two CPUs and four DRPs, outperforms a single-core processor with one CPU by a speed-up factor of 18.4. It is also obtained that the estimated energy on the HCMP with a power control results in as much as 80.0% reduction.

1. はじめに

半導体製造技術の進歩による素子の微細化により、1チップ上に膨大な数のトランジスタを集積することが可能となっている。従来、微細化とともにプロセッサの高周波数化が進んできたが、動作時電力の増加、またリーク電流に起因する待機時電力の増加により、動

作周波数の向上は望めなくなっており、さらに論理方式の改善による性能向上にも限界が見え始めている。一方で、自動車のナビゲーションシステム、携帯電話、デジタルテレビ等、画像、音声、データベース情報等、多様なデータを同時に扱うデジタル民生機器が登場し、特性の異なる膨大なデータを短時間にかつ低電力に処理することが強く求められている。

この要求に対し、性能改善と低電力化を両立させる手段として、現在複数個のプロセッサコアを1チップ上に集積した、チップマルチプロセッサが大きな注目を集めている。たとえば、富士通のFR-V¹⁾、ARM MPCore²⁾、IBM、ソニー、東芝のCell³⁾といったプロセッサがすでに開発されている。筆者らも、コン

[†] 早稲田大学基幹理工学部情報理工学科

Department of Computer Science, Waseda University

^{††} 株式会社日立製作所中央研究所

Central Research Laboratory, Hitachi, Ltd.

現在、株式会社NTTデータ

Presently with NTT DATA Corporation

パイラ主導によりプログラムを複数粒度のタスクに分割し並列性を抽出して実行するマルチグレイン並列処理と、マルチグレイン処理ならびにコンパイラによる最適電力制御に対応した、OSCAR チップマルチプロセッサを提案している⁴⁾。

また、特にデジタル民生機器応用等の組み込みプロセッサでは、高い電力性能が求められており、汎用プロセッサに加え専用プロセッサ、アクセラレータ等、様々な種類のプロセッサエレメント (PE) をオンチップで搭載した、ヘテロジニアスチップマルチプロセッサが提案⁵⁾⁻⁷⁾ されている。ヘテロジニアスチップマルチプロセッサは、デジタル民生機器向けアプリケーションを効率良く処理する専用プロセッサの並列活用により、動作周波数を過度に引き上げることなく消費電力を低く保ったまま高い演算性能を得ることを狙いとする。しかしながら、これを実現するためには、プログラム中からタスク間の並列性を抽出すること、ならびに各 PE の性質と個数に応じたタスクスケジューリングを行うことが必須となる。これまでの同プロセッサの活用手法としては、プログラム内の専用プロセッサ向け処理部の手動による固定的な割付け利用⁵⁾、アプリケーション単位での並列利用⁶⁾、プロセッサ個数が限られた条件でのプログラム自動分割による利用⁷⁾、がある。

また、並列性抽出以外にも、メモリウォール問題に対応するためのキャッシュやローカルメモリの最適利用⁸⁾、PE や各種メモリを接続するネットワーク上のデータ転送オーバーヘッドの最小化を目指したデータ転送スケジューリング⁹⁾ も必須となる。さらに、多数個の PE を集積することによる消費電力の増大という問題も生じるため、各 PE 単位で電源、クロック周波数変更等のこまめな電力管理が重要となる。筆者らは並列化コンパイラによるタスクスケジューリングならびに電力制御に対応したソフトウェア協調 OSCAR ヘテロジニアスチップマルチプロセッサ (HCMP)¹⁰⁾、並列化コンパイラによる HCMP 向けスケジューリング手法¹¹⁾、および並列化スケジューリング結果に基づき性能劣化なしに消費電力を低減させるホモジニアスチップマルチプロセッサ向け電力制御手法¹²⁾ の提案を行っている。

本論文では、提案するタスクスケジューリング手法ならびに電力制御手法を簡易的に実現したスケジューラを作成し、MP3 エンコーダに適用することで、様々な PE 構成における HCMP の性能および電力評価を実施した。以下、2 章では HCMP アーキテクチャ、3 章では粗粒度タスク並列処理方式と電力制御方式、4

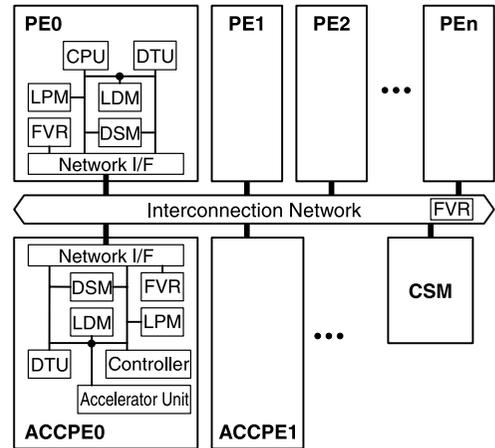


図 1 HCMP アーキテクチャ構成例

Fig. 1 Organization example of HCMP architecture.

章では MP3 エンコーダを例とした並列処理方式ならびに電力制御方式とその結果について述べる。

2. ヘテロジニアスチップマルチプロセッサ

ソフトウェア協調 OSCAR ヘテロジニアスチップマルチプロセッサ (HCMP) は、従来の OSCAR チップマルチプロセッサ⁴⁾ を拡張し、汎用プロセッサ (CPU) に加え、動的再構成可能プロセッサ (DRP)、信号処理プロセッサ (DSP)、等の専用プロセッサ (アクセラレータ) を 1 チップに集積した、異種 PE 混合アーキテクチャである。図 1 に HCMP の構成例を示す。並列化コンパイラとの協調による高い電力性能を達成すべく、構成要件として以下の機構を備える。

● 階層メモリアーキテクチャ

各 PE 近傍に高速な各種ローカルメモリ (ローカルプログラムメモリ LPM, ローカルデータメモリ LDM, 分散共有メモリ DSM) を配置する。コンパイラ管理下でデータをこれらの高速メモリへ分割配置することで、低速な外部メモリアクセスを低減させ、処理性能を最大限引き出す。なお、メモリコヒーレンスはソフトウェアで保証する。

● データ転送機構

データ転送ユニット (DTU) を各 PE の近傍に持ち、ソフトウェア制御によりタスク処理と並行してメモリ間データ転送を行うことで、データ転送を隠蔽する。コンパイラによるデータ転送スケジューリングを実現するために、ローカルメモリ等に DTU 転送命令列を配置し CPU と非同期で複数転送を連続実行する、転送リストに対応する。

- 低電力制御機構
プロセッサコア、メモリ等の機能ブロックに対する周波数・電源電圧を決定する、電力制御レジスタ (FVR) を持つ。コンパイラは電力制御レジスタを操作するコードを生成し、電源管理を行う。
- コントローラ内蔵専用プロセッサ
専用プロセッサはコントローラを内蔵し、コンパイラ生成コードを実行することで、専用プロセッサ単体でスケジューリング管理 (同期、データ転送、アクセラレータユニット制御) を行う。

3. コンパイラ並列処理方式と電力制御方式

本章ではまず、HCMP 上の異種 PE に対し、その PE の性質に合わせたプログラムのタスク分割とスケジューリングを行う粗粒度タスク並列処理方式およびヘテロ構成対応スケジューリングアルゴリズム¹¹⁾ について説明する。続いて、本方式による並列化結果に基づき、処理時間を劣化させずに電力を最適化するコンパイラ電力制御方式を説明する。

3.1 粗粒度タスク並列処理方式

粗粒度タスク並列処理⁴⁾ とは、通常の逐次型プログラムを、繰返しブロック (RB; Repetition Block)、サブルーチン (SB; Sub Routine)、擬似代入文ブロック (BPA; Block of Pseudo Assignment statements) の粗粒度マクロタスク (MT) に分割し、MT 間の並列性を利用する並列処理手法である。並列化コンパイラは、分割生成した MT 間の制御フローおよびデータ依存性を解析し MT の実行順序関係を定義する最早実行可能条件解析により、プログラムの並列性を表現したマクロタスクグラフ (MTG) を生成する¹³⁾。その後、コンパイラは MTG 上の MT を各 PE にスケジューリングする。このとき、MTG 内に条件分岐等のコンパイル時不確定要素がなければ、同期やデータ転送等のオーバーヘッドを最小化するために静的に MT を各 PE に割り当てる (スタティックスケジューリング)。また、不確定要素が存在する場合は、実行時に MT の最早実行条件を管理しつつ MT を割り当てるスケジューラコードを生成する (ダイナミックスケジューリング)。

3.2 ヘテロ構成対応スケジューリングアルゴリズム

これまでの並列化コンパイラは、スケジューリング対象がすべて同一の演算性能を持った汎用プロセッサマルチ構成であることを前提としていた。ヘテロジニアスマルチプロセッサ向けスケジューリング手法に関する研究^{14),15)} は従来からあるが、ランダムグラフや FFT 等のシンプルなアプリケーションに対するスケジューリング実行時間のみしか評価されておらず、プ

ロセッサの種類や個数の変化に対するアプリケーションの速度向上率が評価されていない。筆者らは新たに、通常の汎用プロセッサに対し特定の処理を高速に行う専用プロセッサをその種類ごとにグループ化し、汎用プロセッサに対するスケジューリングを 3.1 節で示した手法で行いつつ、全体の処理時間を短縮できるように専用プロセッサにタスク割当てを行う、スタティックスケジューリング手法を開発した。なお、専用プロセッサで実行可能なプログラム部分は、あらかじめ指示文等で専用プロセッサが適用できる旨、指定されているとする。以下、スケジューリングの手順を説明する。

Step 1. (タスクコスト解析) コンパイラが生成するプログラム内のタスク並列性と実行順序を表現したマクロタスクグラフにおいて、各タスクの処理コストと出口ノードからの最長パス長 (クリティカルパス長) を求める。

Step 2. (レディタスク検出) その時刻で次に実行可能なタスクをマクロタスクグラフ情報より検出する。次に実行可能タスク (レディタスク) とは、あるタスクが実行されると依存制約が解決し実行可能となるタスク群を指す。

Step 3. (終了時刻計算) クリティカルパス長が最大のタスクを割当てタスクとし、当該タスクを割当て可能な PE に対する割当てを仮定したときの処理終了時刻を推定する。なお本論文では、タスクのプロセッサタイプを、指示文として処理ブロックやサブルーチン単位でユーザがあらかじめ記述しておく。時刻推定は、下記の式に基づき行う。

$$\begin{aligned} \text{終了時刻} &= \text{先行タスクの終了時刻} \\ &+ \text{ローカルメモリへのデータ/プログラム} \\ &\quad \text{転送サイクル数} \\ &+ \text{プロセッサの起動サイクル数} \\ &+ \text{対象 PE 上でのタスク実行サイクル数} \\ &+ \text{ローカルメモリから他メモリへの} \\ &\quad \text{データ転送サイクル数} \end{aligned}$$

Step 4. (タスク割当て) 終了時刻が最も早いと推定された PE に対しタスクを割り当てる。なおこのとき、対象タスクが専用プロセッサ処理可能タスクであってもその処理状況により、汎用プロセッサでの処理を仮定した終了時刻の方が早い場合は、汎用プロセッサに割り当てられる。

Step 5. (終了判定) 未割当てタスクが存在する場合は、次に最も早くタスク時刻を終了する PE のタスク終了時刻をスケジューリング時刻に更新し、Step 2 からを繰り返す。未割当てタスクがない場合は、スケジューリングを終了する。

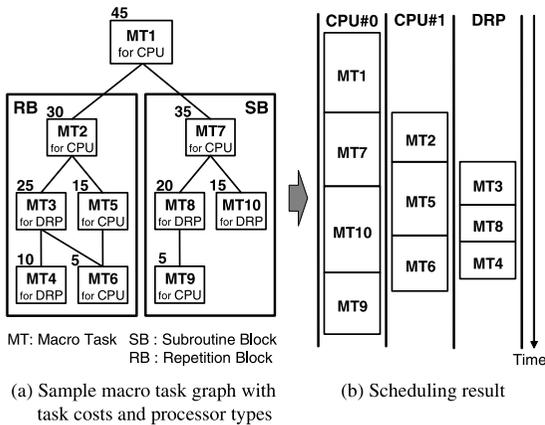


図2 マクロタスクグラフとヘテロスケジューリング例
Fig. 2 Examples of a macro task graph and its scheduling.

本スケジューリング手法を、CPU×2+DRP×1 構成の HCMP に対するスケジューリングを具体例として図2を用いて説明する。図2(a)は並列化コンパイラによるプログラム解析の結果生成されたマクロタスクグラフである。グラフの各ノードがタスクを表現し、タスク間の実線がデータ依存を表す。図中 MT ブロック左上に書かれた数字は、当該タスクからのクリティカルパス長を示す。

コンパイラは、まず処理開始時のレディタスク MT1 (CPU タスク) を CPU#0 に割り当てる。次の実行可能タスクは、MT1 処理終了後の MT2 と MT7 である。MT2, MT7 は CPU タスクであり、この時点で CPU#0, #1 はアイドル状態のため、クリティカルパス長が大きな順に MT7 を CPU#0 に、MT2 を CPU#1 に割り当てる。次にレディ状態となるタスクは、MT2 処理終了後の MT3 (DRP タスク) および MT5 である。この時点で CPU#1 および RPP はアイドル状態のため、MT3 を DRP に、MT5 を CPU#1 に割り当てる。

続いて、MT7 が終了時に MT8 および MT10 がレディとなる。MT8 および MT10 はともに DRP タスクである。コンパイラはクリティカルパス長が大きな順に、まず MT8 をスケジューリング対象とする。続いて、MT8 を実行可能な各 PE に対して、MT8 実行時の終了時刻を計算する。その結果、本例では終了時刻が最も早い DRP に処理を割り当てる。次に MT10 に対しても同様に終了時刻を計算する。MT10 は本来 DRP 上で高速に処理可能なタスクであるが、DRP 上には実行中のタスクが存在するため、CPU#0 で実行した場合が終了時刻が最も早くなる。よって、コンパイラは MT10 を CPU#0 に割り当てる。以上の手

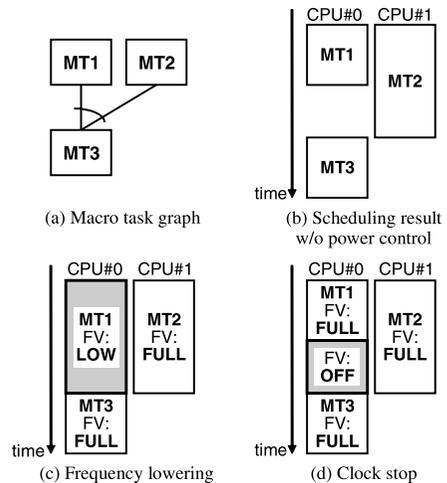


図3 コンパイラによる電力制御手法
Fig. 3 Compiler power control scheme.

順と同様に、MT4 を DRP に、MT6 は CPU#1 に、MT9 は CPU#0 に割り当て、スケジューリングを終了する。

3.3 タスク並列処理電力制御方式

並列化コンパイラは、前述のように HCMP 上の PE を最大限活用するタスク並列化スケジューリングを行い、プログラムの実行時間を最小化する。しかしながら、実際のプログラムにはデータ依存や制御依存の制約から逐次実行せざるをえない部分が存在するため、すべての PE をつねに同時に活用することはできない。並列化コンパイラによるタスクスケジューリングでは、その結果から各時刻における PE の状態を予測することができる。つまり、コンパイラが PE ごとに、その状態に応じた電力制御コードを生成することで、プログラム実行時の電力を低減させることができる¹²⁾。たとえば、PE が同期待ち状態の場合に周波数 f および電源電圧 V を低減させたり、アイドル状態の場合にはクロック停止や電源遮断を行ったりすることで、電力を削減する。

図3(a)にコンパイラが生成するマクロタスク(MT)例を、(b)に2個のCPUへのスケジューリング例を示す。本例では、MT1をCPU#0、MT2をCPU#1にて並列実行し、MT3をCPU#0で実行するとする。まず、図3(c)にタスク実行時の周波数低減を行う手法の適用例を示す。コンパイラは、CPU#0におけるMT1の処理サイクル数および、CPU#1におけるMT2の処理サイクル数を見積もることで、双方の処理時間が等しくなるようCPU#0の f を決定する。この結果、 f が低減されるため、電源電圧 V も低減されること(FV: LOW モード)が可能となり、 f, V^2 に比例す

動的電力 P_{at} を削減できる。

次に、図 3(d) に PE アイドル時にクロック停止/電源遮断を行う手法の適用例を示す。CPU#0 上の MT1 は MT2 に先行して終了するため、CPU#0 は MT2 終了時までアイドル状態となる。そこで、MT1 終了時点で CPU#0 を待機状態 (FV: OFF モード) とし、MT2 が終了した時点で再び CPU#0 に対し通常の電源電圧および動作クロックを供給することで通常状態に復帰させ、MT3 を開始する。この結果、CPU#0 は停止し電源遮断されるため、 P_{at} に加えリーク電流を含めた静的電力 P_{st} も削減できる。

4. MP3 エンコーダを用いた評価

HCMP アーキテクチャの性能評価を目的に、MP3 (MPEG 1 Audio Layer 3) エンコーダを評価対象アプリケーションとしてその並列処理方式と電力制御方式を検討し、性能評価を実施した。

4.1 MP3 エンコード処理

MP3 エンコードは、オーディオデータを人間の聴覚特性に応じて圧縮する。オーディオ信号を時間方向にサンプリングした PCM データに対し、フレーム単位で処理を行う。まず PCM データに対し、サブバンド解析 (S) および MDCT (MD) によって、時間-周波数変換を行う。次に、心理聴覚分析 (P) により人間の聴覚特性に基づいて周波数データのマスクングを行う。続いて非線形量子化 (Q) によって、周波数データを、心理聴覚分析結果を用いて音質悪化が最低限に抑えられるレベルで間引く。次に、ハフマン符号化 (HF) により符号情報量の削減を行い、ビットストリーム生成 (BS) で MP3 規格に対応したストリームデータを生成する。今回の評価では、UZURA MP3 エンコーダ¹⁶⁾を参照実装したプログラムを利用した。UZURA は FORTRAN で記述されており、プログラムの並列性解析に FORTRAN に対応した OSCAR 並列化コンパイラを利用できる。また MP3 エンコード規格を忠実に参照実装しており、特定のアーキテクチャに依存しないため、標準的なベンチマークと位置づけた。

MP3 エンコードでは、一部の処理を除きフレーム間並列性を活用することで、並列処理が可能である。図 4(a) に評価プログラムのプログラム構造を、図 4(b) に 4 フレーム並列時のタスクグラフを示す。オリジナルのプログラム構造は、1 フレームごとに各処理を実行する形であったが、評価プログラムではコンパイラによるフレーム間の並列性抽出に対応するために、各処理に対し指定されたフレーム数分連続して実行するようループ化した。これにより、フレーム間で依存性

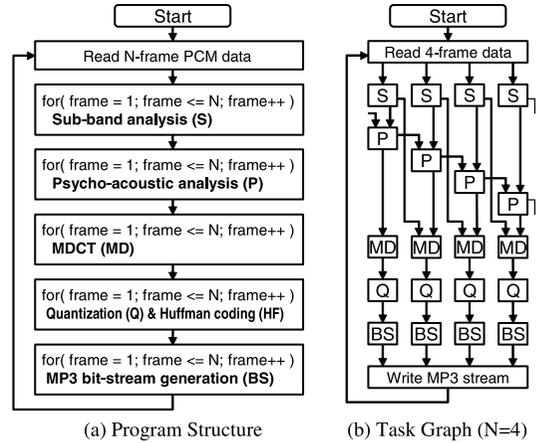


図 4 MP3 エンコーダプログラムの構造とタスクグラフ
Fig. 4 Program structure and task graph of MP3 Encoder.

がない処理は並列化可能ループと判定される。また、フレーム間でデータ依存がある処理 (心理聴覚分析) は逐次実行される。

4.2 評価モデルと評価条件

HCMP による性能向上ならびに電力削減効果を見積もるための評価モデルを、今回新たに定義した。並列化コンパイラと詳細シミュレータ開発前の高精度な性能評価を行うにあたり、プログラムの構造解析によるタスクグラフの構築と、3.2 節で説明したタスクスケジューリング手法、3.3 節で示した電力制御を実施するタスクスケジューラの構築を行った。

評価対象 HCMP は、CPU コアとして SH-4 相当¹⁷⁾ (以下 SH) ならびに DRP の一種である Flexible Engine/Generic ALU Array (FE)¹⁸⁾ をそれぞれ複数個混載する構成とした。評価パラメータの一覧を表 1 に示す。クロック比は SH:FE:バス = 2:1:1 とし¹⁹⁾、各コアは 1 サイクルアクセス可能な 128 KB のローカルメモリ (LDM, DSM 共用)、ならびにデータ転送ユニットを備える。またさらに、入出力データ等の共有データを格納する 512 KB のオンチップ CSM を持つ。各 PE を接続するバスはアトミックトランザクションタイプとし、転送を排他的に行う。また電力評価対象はプロセッサコアのみとし、動作時電力算出に用いる単位電力値を、SH は 0.3 mW/MHz²⁰⁾、FE-GA は 0.8 mW/MHz¹⁸⁾ とした。これらの値は、プロセッサコアおよび SH においてはキャッシュ、FE においてはローカルメモリを含めた、動作時における平均電力値である。なお、本論文での電力は、単位時間消費電力に処理時間を乗じた、電力エネルギーとして算出した。

表 1 評価モデルと条件

Table 1 Evaluation model and conditions.

PE 構成	SH と FE のヘテロマルチ構成
クロック比	SH, FE, バス, クロック比 2:1:1
ローカルメモリ	PE ごとに 128 KB, 1 サイクルアクセス
共有メモリ	512 KB オンチップ共有メモリ
データ転送	PE ごとにデータ転送ユニット 転送設定に 40 SH サイクル 32 B 転送ごとに 16 SH サイクル
オンチップバス	64 b アトミックトランザクションバス ×1
電力指標	SH: 0.3 mW/MHz ²⁰⁾ , FE: 0.8 mW/MHz ¹⁸⁾

表 2 シミュレータ評価条件

Table 2 Simulator model and conditions.

CPU モデル	SH-4 (SH7750) 相当 ¹⁷⁾
演算器	整数演算器, 浮動小数点演算器 (FPU) 搭載 (FPU を含む 2-way スーパースカラ)
キャッシュ	命令 (16 KB), データ (32 KB)

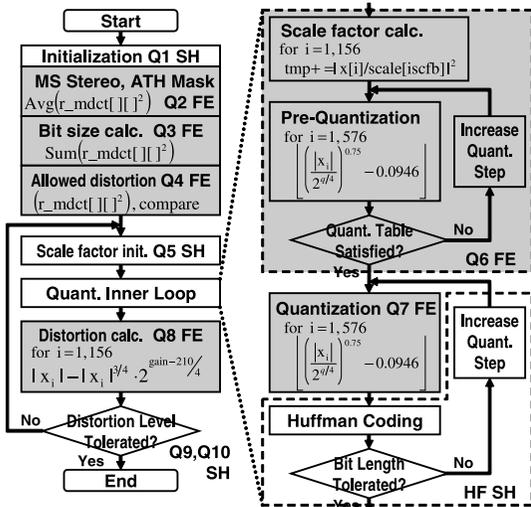


図 5 非線形量子化処理のタスク分割

Fig. 5 Extracted tasks of non-linear quantization.

4.3 評価手順

評価では、フレーム並列化単位を $N = 4$ とし、4 フレーム PCM データ (16 bit 44.1 KHz) 入力時のエンコード処理サイクル数を算出し、様々な PE 構成による処理時間、電力の差異を評価した。

Step 1. (タスク分割) プログラムを粗粒度タスクに分割し、データ依存解析と処理対象 PE の分類を行う。解析には、OSCAR 並列化コンパイラの並列性解析機能を利用した。評価対象 HCMP が持つ FE は、その性能を効果的に引き出すことが可能なプログラム構造が限定されるため、手動で調査分類を行った。具体的には、ループ処理で一定以上の回転数 (データ数) を持ち、なおかつ当該ループ内のステートメント間のデータフローが一定以上の並列度と複雑さを持つものを、FE 処理対象ブロックとして指示文により指定した。図 5 に非線形量子化処理のタスク分割結果を示す。

Step 2. (処理時間導出) タスク処理サイクルを、SH アーキテクチャシミュレータ²¹⁾ を用いて算出す

る。算出条件を表 2 に示す。処理サイクルは、4 フレーム分処理実行時の平均をとった。求めた各タスクの処理サイクルを表 3 に示す。なお、FE に分類したタスクの処理サイクル数は、MP3 と同様のオーディオ圧縮規格である AAC エンコーダの評価結果¹⁹⁾ より平均 10 倍性能が得られるとし、SH サイクルの 1/10 として算出する。FE-GA 実行可能タスクの処理全体に占める割合は、SH サイクル換算で処理全体の 79% となっている。

Step 3. (データ転送時間算出) 変数の宣言サイズより、タスクごとの入出力データ転送量を算出し、データ転送サイクル数を求める。各 PE が持つデータ転送ユニット (DTU) を用い、入出力データを共有メモリ (CSM)・ローカルメモリ (LM) 間でタスク実行前後に転送する。求めた転送サイクル数を表 3 に示す。

Step 4. (スケジューリング) タスクグラフを構築し、3.2 節で説明したタスクスケジューリング手法でタスク割付けを行い、エンコードに要するサイクル数を導出する。なお、非線形量子化処理では分岐や不定回転数ループが存在するため、本評価では分岐方向およびループ回転数を実行プロファイルにより求めた。

4.4 タスクスケジューリング結果

前節の手順で表 3 に示した各タスクに対し、様々な PE 構成の HCMP に対しスケジューリングを実施し、4 フレームデータ入力時のエンコード処理サイクル数を導出した。なお実行時間は、第 1 フレームの入力データを CSM よりロード開始してから、第 4 フレームの出力データを CSM にストアするまでの時間とした。以下では、HCMP における専用プロセッサ (以下、アクセラレータ) の導入効果、ならびにアクセラレータを含めた複数 PE 上でのデータ転送オーバーヘッドを考慮した並列処理性能の評価を目的に、シングルコア CPU での逐次実行である SH×1 に対する加速率を求めた。表 4 に、評価 PE 構成と、各構成における処理サイクル、SH×1 に対する加速率を示す。

まず、シングルコア SH における逐次処理では、処理時間が 99.6 M サイクルとなった。ホモジニアスマルチコア構成では、SH×4 で 25.7 M サイクルとなり、

表 3 タスク分割の詳細とタスク実行・データ転送サイクル数
Table 3 Extracted tasks and their execution and data transfer cycles.

処理		タスク	対象 PE	SH 処理サイクル	ロードサイクル	ストアサイクル
サブバンド 解析 (S)	ポリフェーズフィルタ	S1	SH, FE	5,028,610	19,240	9,256
	サブバッファ処理	S2	SH	132,885	9,256	13,864
心理聴覚分 (P)		P	SH, FE	239,837	9,256	56
MDCT (MD)		MD	SH, FE	810,564	13,876	9,256
量子化 (Q)	初期化	Q1	SH	209,767	16,282	14,516
	MS ステレオ, ATH マスク	Q2	SH, FE	929,841	9,260	9,260
	ビット量算出	Q3	SH	723	72	56
	歪み許容量算出	Q4	SH, FE	138,288	16,176	9,256
	圧縮タイプ決定	Q5	SH	53,337	19,234	6,962
	スケールファクタ計算	Q6	SH, FE	2,5864,09	7,726	2,348
	量子化処理	Q7	SH, FE	177,923	2,348	2,348
	歪み量算出	Q8	SH, FE	1,675,571	9,528	56
	スケールファクタ前処理	Q9	SH	119,184	22,018	6,952
	歪み量判定	Q10	SH	65,847	2,824	2,376
ハフマン符号化 (HF)		HF	SH	14,522	2,344	2,344
ビットストリーム生成 (BS)		BS	SH	322,806	2,348	2,344

表 4 各 PE 構成における実行サイクル数と加速率

Table 4 Operation cycles and speed-up ratio on various PE configurations.

Configuration	Operation cycles [Mcytc]	Speed-up to SH×1
SH×1	99.6	1.0
SH×4	25.7	3.9
SH×1 + FE×1 (Seq.)	16.3	6.1
SH×1 + FE×1	12.7	7.8
SH×2 + FE×1	10.9	9.1
SH×1 + FE×2	9.0	11.0
SH×2 + FE×2	6.8	14.6
SH×2 + FE×3	5.7	17.4
SH×3 + FE×2	6.5	15.3
SH×2 + FE×4	5.4	18.4
SH×3 + FE×3	5.5	18.2
SH×4 + FE×4	4.7	21.1

逐次処理に対し 3.9 倍の速度向上を得た。SH×4 に対するタスクスケジューリング結果および、データ転送 (DMA) を表すガントチャートを図 6 に示す。図中の記号はタスク名と、対象フレーム番号を表す。たとえば、“S1 F1” は第 1 フレーム (F1) のサブバンド解析処理 (S1) タスクを示す。フレーム間でデータ依存がある心理聴覚分析タスク (P) を除き、フレーム並列でスケジューリングされたことが確認できる。

次に、SH に対し 10 倍性能としたアクセラレータ (FE) を導入した HCMP 構成に対するスケジューリング結果を見る。まず、SH×1 + FE×1 の構成で SH と FE 交互に逐次的に処理を実行した場合、処理サイクル数は 16.3M サイクルとなり、SH×1 に対し 6.1 倍性能となった。これは、アクセラレータによる速度向上率を表している。次に 3.2 節で説明した並列処理手法を適用した場合、SH×1 + FE×1 の構成では処理

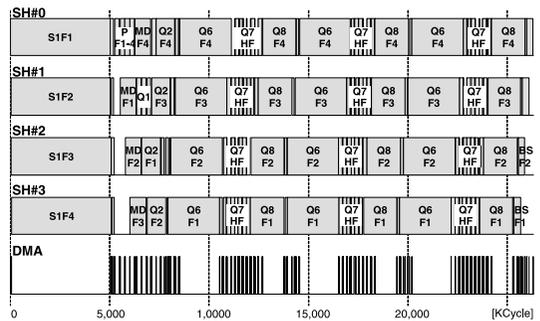


図 6 SH×4 スケジューリング結果
Fig. 6 Scheduling result on SH×4.

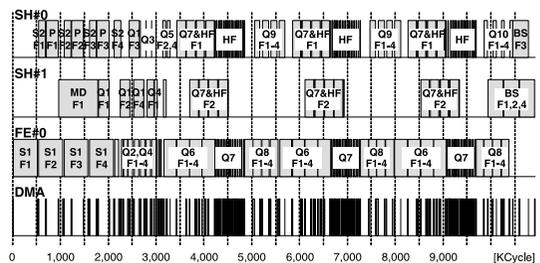


図 7 SH×2 + FE×1 スケジューリング結果
Fig. 7 Scheduling result on SH×2 + FE×1.

時間は 12.7M サイクル、SH×2 + FE×1 の構成では処理時間は 10.9M サイクルとなり、SH×1 の逐次処理に対して、それぞれ 7.8 倍、9.1 倍の加速率を得た。SH×2 + FE×1 構成におけるスケジューリング結果を示すガントチャートを図 7 に示す。これより、FE 処理に適しているタスクは FE に割り当てられ、アクセラレータの活用と並列処理がなされていることが確認できる。なお、FE の個数が 1 個と少ないため、FE の

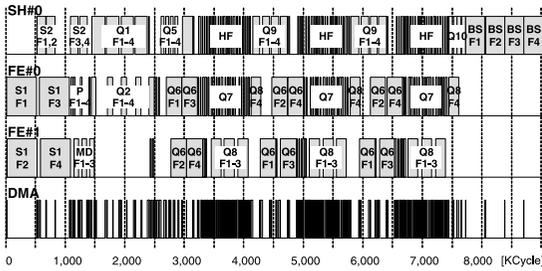


図 8 SH×1 + FE×2 スケジューリング結果
Fig. 8 Scheduling result on SH×1 + FE×2.

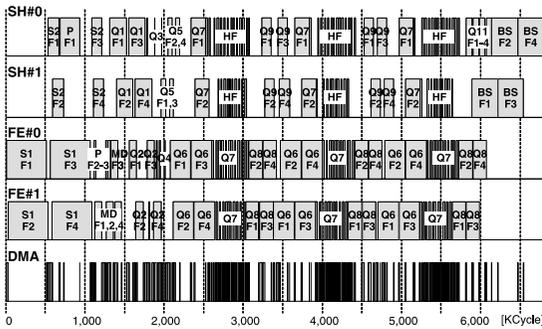


図 9 SH×2 + FE×2 スケジューリング結果
Fig. 9 Scheduling result on SH×2 + FE×2.

状況によっては FE 向けタスクが SH に割り当てられたことも確認できる。たとえば、心理聴覚分析 (P) は FE 向けタスクであるが、図 7 を見ると 1~3 フレーム目のタスクが SH#0 に割り当てられている。

次に、アクセラレータの個数を増加させると、SH×1 + FE×2 の構成では処理時間は 9.0M サイクルとなり、SH×1 の逐次処理に対して 11.0 倍の加速率を得た。本構成におけるガントチャートを図 8 に示す。続いて、SH の個数を増やし SH×2 + FE×2 の構成では処理時間は 6.8M サイクルとなり、SH×1 の逐次処理に対して 14.6 倍の加速率を得た。本構成におけるガントチャートを図 9 に示す。さらにアクセラレータ数を増加した SH×2 + FE×4 の構成では処理時間は 5.4M サイクルとなり、SH×1 の逐次処理に対して 18.4 倍の加速率を得た。本構成におけるガントチャートを図 10 に示す。また、SH×4 + FE×4 の構成では処理時間は 4.7M サイクルとなり、SH×1 の逐次処理に対して 21.1 倍の加速率を得た。

次にアクセラレータを含めたタスク並列化の効果を確認するために、SH×1 + FE×1 構成での逐次処理を基準とした速度向上率を見る。表 4 より SH×1 + FE×1 で並列処理を行った場合、SH×1 + FE×1 での逐次処理に対し 1.3 倍の向上、基準の 2 倍数となる SH×2 + FE×2 で 2.4 倍、4 倍数となる SH×4 +

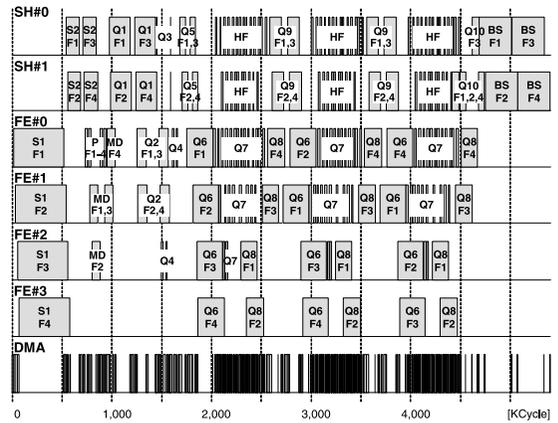


図 10 SH×2 + FE×4 スケジューリング結果
Fig. 10 Scheduling result on SH×2 + FE×4.

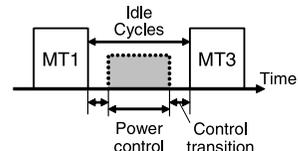


図 11 電力制御適用対象の決定方法
Fig. 11 Determination of power control.

FE×4 で 3.5 倍であった。SH、FE 個数の増加に対する速度向上率は飽和するが、これは今回の評価では並列化単位フレーム数を $N = 4$ と固定しているため、並列化単位を超えた PE 数の増加により PE の稼働率が減少することに起因する。また、ガントチャートより、SH×2 + FE×2 構成が各 PE の利用率が高く、複数のアクセラレータが効果的に並列利用されている様子が確認できる。MP3 エンコード処理では、プロセッサ数に合わせ最適な並列化単位フレーム数 N を設定することが、性能を引き出す鍵となる。

4.5 電力制御の適用

次に、タスクスケジューリング結果に対し電力制御を適用し、各構成における電力値を算出した。今回適用した制御方式は、タスク実行時は通常の電源電圧・クロックを供給し、アイドル状態においてのみ電力制御を行う図 11 に示す手法とした。適用した電力制御手法は、以下の 4 手法である。

- (1) 周波数低減 通常のクロック ϕ を $1/8\phi$ とした低周波クロックに切り替えることで、アイドル時の動的電力 P_{at} を削減させる。クロック切替え時の安定化時間が必要となる。
- (2) クロック停止 クロックゲーティング等によりクロック供給を止め、アイドル時 P_{at} を削減する。PE は完全に停止するが、メモリやレジスタ等の

表 5 適用電力制御手法
Table 5 Power control techniques applied.

電力制御手法	遷移時間	電力効果
(1) 周波数 1/8 低減	100	$P_{at} \rightarrow 1/8 P_{at}$
(2) クロック停止	2,000	$P_{at} \rightarrow 0$
(3) 電源遮断	20,000	$P_{at} \rightarrow 0, P_{st} \rightarrow 0$
(4) 上記併用	< 20,000	(3) > (2) > (1) 優先

[遷移時間の単位はサイクル]

記憶素子内のデータは保持される。

- (3) 電源遮断 PE に対する電源供給を遮断し、アイドル時 P_{at} および静的電力 P_{st} を削減させる。メモリデータが破壊されてしまうため、PE 外メモリへ退避する必要がある。また、復帰時にも退避データの書き戻し、また電源安定化時間も必要となる。
- (4) (1) ~ (3) 併用 (3) > (2) > (1) の優先順で、アイドル時間の大きさにより (3) を優先的に適用し、アイドル時間が短く (3) を適用できない場合は (2) を、同様に (2) を適用できない場合は (1) を適用する。なお、このとき各手法の効果を引き出すために、各手法の適用基準を、アイドル時間が制御時間の 5 倍以上とした。

電力制御時の制御時間は、Cooperative Voltage Scaling²²⁾ を参考に、制御オーバヘッドサイクルとして、表 5 のとおり定義した。動作周波数が 400 MHz 時で、それぞれ (1) 250 nsec, (2) 5 μ sec, (3) 50 μ sec に相当する。

4.6 電力値の算出方法

電力値は、4 フレーム分データに対するエンコード処理を実行した際の消費エネルギーとして求めた。まず、動的電力 P_{at} を「単位電力値 (mW/MHz) \times 処理時間」として算出した。またリーク電力等に起因する静的電力 P_{st} は、製造プロセスやトランジスタ数に依存するが、本論文では P_{at} を基準として、電力制御非適用時の P_{at} と P_{st} を合わせた総電力値の 2 割となるように P_{st} を設定した²³⁾。なお、今回の評価では、プロセッサ全体電力に占める割合が大きなブロックである²³⁾ プロセッサコアを電力算出の対象とし、メモリ、バス、I/O ユニット等の電力は算出対象外とした。

4.7 評価結果

表 4 で示した各 PE 構成において、表 5 に示した手法を適用し、MP3 エンコード処理 (4 フレームデータ入力時) の電力エネルギーを算出した。評価結果を図 12 に示す。グラフは、各 PE 構成における電力制御非適用、(1) ~ (4) の手適用時の電力エネルギーを示す。

まず、電力制御非適用時のエネルギーを見ると、

SH \times 1 の逐次処理では 37.4 mJ の消費エネルギーに対し、SH \times 4 では 39.5 mJ となり、並列化による処理時間の短縮により PE 数が増加してもエネルギーは微増の結果となっている。次にアクセラレータを導入した HCMP 構成を見ると、SH \times 2 + FE \times 2 の構成においては 11.9 mJ となり、電力エネルギーは SH \times 1 の逐次処理に対して 68.1%低減される。また、SH \times 2 + FE \times 4 の構成においては 14.9 mJ となり、電力エネルギーは SH \times 1 の逐次処理に対して 60.2%低減される。これは、アクセラレータである FE の単位電力値は SH に比べ約 2.7 倍大きい、処理効率が高い FE の活用により処理時間が大きく削減されたからである。電力効率の良いアクセラレータの活用により、電力エネルギーの大幅な削減が期待できることが分かる。

また、ヘテロジニアス構成の電力制御非適用時における電力エネルギーの傾向を見ると、PE 数の増加によってエネルギーは増加するが、全体の PE 数が同じ場合 PE 構成によって電力エネルギーに差が見られる。MP3 エンコード処理では、4.4 節で示したように SH 数と FE 数が同等の構成において、処理時間が最小となっており、電力効率も同様のことがいえることが確認できる。

続いて、各電力制御手法の電力エネルギー低減に対する効果について着目する。たとえば、SH \times 2 + FE \times 2 の構成においては、制御非適用の場合 11.9 mJ に対し、(1) クロック低減では 8.5 mJ, (2) クロック停止では 8.1 mJ, (3) 電源遮断では 8.6 mJ, (4) 併用では 7.9 mJ となった。これより、電源制御手法単独適用では、クロック停止が効果が最大となる結果となった。また、併用手法によりアイドル状態の期間によって各手法を併用することで、電力エネルギーは 34.9%削減された。これに対して、SH \times 2 + FE \times 4 の構成においては、制御非適用の場合 14.87 mJ に対し、(1) クロック低減では 9.1 mJ, (2) クロック停止では 8.4 mJ, (3) 電源遮断では 8.3 mJ, (4) 併用では 7.5 mJ となった。これより、電源制御手法単独適用では、電源遮断が効果が最大となる結果となった。また、併用手法によりアイドル状態の期間によって各手法を併用することで、電力エネルギーは 49.7%削減された。SH \times 1 の逐次処理に対する電力エネルギーと比較すると、併用手法を適用することにより、80.0%の削減となった。

電力制御手法単独適用時に、PE 構成によって電力削減効果が変わっている。たとえば、PE 数が 5 以下では電力削減に対し最も効果的な手法がクロック停止であるのに対し、PE 数が 6 以上では電源遮断が最も効果的な手法となっている。これは PE 数が増えると、

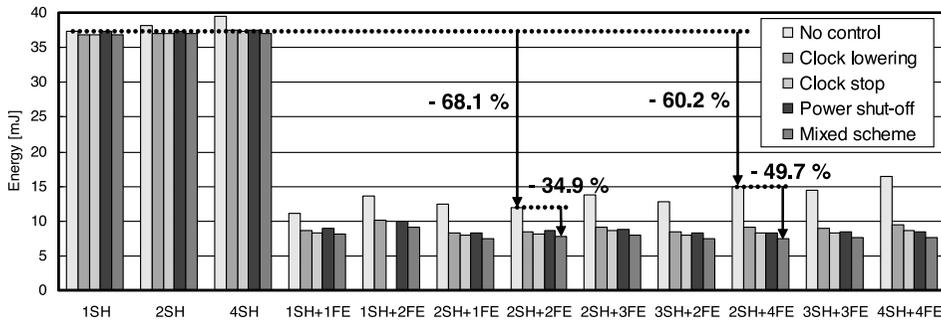


図 12 消費電力結果

Fig. 12 Estimated energy with power controlling applied on various configurations.

表 6 電力制御併用手法適用時の各電力手法適用回数とサイクル数

Table 6 Detail of power controls applied in mixed control mode.

Applied cycles	SH#0	SH#1	FE#0	FE#1	FE#2	FE#3
Task execution cycles	2,592	2,548	2,848	2,628	1,997	1,781
Idle cycles	2,813	2,857	2,557	2,778	3,409	3,624
(1) Frequency lowering	9 (2)	66 (11)	88 (10)	14 (3)	36 (5)	0 (0)
(2) Clock stop	1,328 (49)	800 (26)	1,099 (41)	1,059 (32)	156 (5)	75 (1)
(3) Power shut-off	1,238 (7)	1,738 (10)	1,207 (4)	1,541 (5)	3,066 (7)	3,407 (7)

[単位は K サイクル, () は適用回数]

タスクの負荷分散により各 PE のアイドル時間が大きくなるため、オーバヘッド（制御時間）が大きい電源遮断が多く適用され、静的電力 P_{st} が削減されたからである。 P_{st} がより大きな条件では、電源遮断がより効果的な手法となる。

また、(4) 併用手法で、どの構成でも最大の効果が得られているが、これはアイドル状態の期間によって適切な手法が選択された結果に起因する。表 6 に $SH \times 2 + FE \times 4$ の構成に対する電力制御併用手法適用時の、タスク実行サイクル、アイドルサイクル、(1)~(3) 各手法適用回数および適用サイクル数を示す。これより、タスクが密にスケジューリングされている $SH\#0$, $SH\#1$, $FE\#0$, $FE\#1$ では、(2) クロック停止が多用されているのに対し、アイドル期間が長い $FE\#2$, $FE\#3$ では (3) 電源遮断が多用されることとなる。

5. まとめ

本論文では、汎用プロセッサ、および各種専用プロセッサ（アクセラレータ）を 1 チップ上に複数個搭載するヘテロジニアスチップマルチプロセッサ（HCMP）の有効性を、HCMP 向けタスクスケジューリング手法、ならびに電力制御手法を適用し、評価した。その結果 HCMP において、並列化コンパイラ主導によるプログラム内での並列処理と電力制御の適用により、下記に示すような速度性能の大幅な向上を、低電力に実現できることを示した。

評価は MP3 エンコードを対象とし、その並列処理方式を検討を行い、評価モデルを構築した。続いて、ヘテロスケジューリング手法を、汎用プロセッサ（SH）と動的再構成プロセッサ（FE）で構成される HCMP に適用し、4 フレーム入力時のエンコード処理時間を見積もった。その結果、 $SH \times 2 + FE \times 2$ 構成の HCMP では $SH \times 1$ 上での逐次処理に対し 14.6 倍の加速率、 $SH \times 2 + FE \times 4$ の構成では 18.4 倍、 $SH \times 4 + FE \times 4$ の構成では 21.1 倍の加速率を得た。

次に、以上の HCMP 上での MP3 エンコード処理における電力エネルギーの見積りを実施した。その結果、 $SH \times 2 + FE \times 4$ の構成においては、 $SH \times 1$ の逐次処理に対する電力エネルギーと比較し、60.2% の削減となった。またさらに、HCMP 上の各プロセッサに対しタスク実行を行わないアイドル状態に、周波数低減、クロック停止、電源遮断を組み合わせた電力制御を適用した。その結果、 $SH \times 2 + FE \times 4$ の構成においては、 $SH \times 1$ の逐次処理に対する電力エネルギーと比較し、80.0% の削減となった。

今後の課題として、スケジューリング機能、電力制御機能を組み込んだ並列化コンパイラの開発ならびに、HCMP アーキテクチャを詳細にシミュレートできるシミュレータによる、実動作モデルでの評価があげられる。また、電力制御手法をプロセッサコアだけでなく、バス、各種メモリ、周辺回路へ拡張し、評価を行っていく予定である。

謝辞 本研究の一部は、NEDO「先進ヘテロジニア

「マルチプロセッサ研究開発」の支援により実施された。有用な討論をいただきました。(株)日立製作所小高俊彦氏, 内山邦男氏, 伊藤雅樹氏, 戸高貴司氏, 田中博志氏, 早稲田大学笠原・木村研究室の諸氏に心より感謝いたします。

参 考 文 献

- 1) Shiota, T., Kawasaki, K., Kawabe, Y., Shibamoto, W., Sato, A., Hashimoto, T., Hayakawa, F., Tago, S., Okano, H., Nakamura, Y., Miyake, H., Suga, A. and Takahashi, H.: A 51.2GOPS 1.0GB/s-DMA Single-Chip Multi-Processor Integrating Quadruple 8-Way VLIW Processors, *Proc. IEEE International Solid-State Circuits Conference (ISSCC2005)* (2005).
- 2) Cornish, J.: Balanced Energy Optimization, *International Symposium on Low Power Electronics and Design* (2004).
- 3) Pham, D., et al.: The Design and Implementation of a First-Generation CELL Processor, *Proc. IEEE International Solid-State Circuits Conference (ISSCC 2005)* (2005).
- 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. 9th Workshop on Interaction between Compilers and Computer Architectures (INTERACT-9)* (2005).
- 5) Torii, S., Suzuki, S., Tomonaga, H., Tokue, T., Sakai, J., Suzuki, N., Murakami, K., Hiragata, T., Shigemoto, K., Tanabe, Y., Ohbuchi, E., Kitayama, N., Eda, H., Kusano, T. and Nishii, N.: A 600MIPS 120mW 70 μ A Leakage Triple-CPU Mobile Application Processor Chip, *Proc. IEEE International Solid-State Circuits Conference (ISSCC 2005)* (2005).
- 6) Hattori, T., et al.: A Power Management Scheme Controlling 20 Power Domains for a Single-Chip Mobile Processor, *Proc. IEEE International Solid-State Circuits Conference (ISSCC 2006)* (2006).
- 7) Mishra, M., Callahan, T.J., Chelcea, T., Venkataramani, G., Budiu, M. and Goldstein, S.C.: Tartan: Evaluating Spatial Computation for Whole Program Execution, *Proc. 12th Int'l Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS XII)* (2006).
- 8) 中野, 仁藤, 丸山, 中川, 鈴木, 内藤, 宮本, 和田, 木村, 笠原: OSCAR マルチコア上でのローカルメモリ管理手法, 情報処理学会研究報告 2006-ARC-169 (28) (2006).
- 9) 宮本, 中川, 浅野, 内藤, 仁藤, 中野, 木村, 笠原: 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ラップ方式, 情報処理学会第 159 回計算機アーキテクチャ・第 105 回ハイパフォーマンスコンピューティング合同研究発表会 (HOKKE-2006) (2006).
- 10) Shikano, H., Suzuki, Y., Wada, Y., Shirako, J., Kimura, K. and Kasahara, H.: Performance Evaluation of Heterogeneous Chip Multi-Processor with MP3 Audio Encoder, *Proc. IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips IX)*, pp.349-363 (2006).
- 11) 和田康孝, 押山直人, 鈴木裕貴, 白子 準, 中野啓史, 鹿野裕明, 木村啓二, 笠原博徳: 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, 情報処理学会研究報告 2006-ARC-166 (3) (2006 年 1 月).
- 12) 白子 準, 吉田宗弘, 押山直人, 和田康孝, 中野啓史, 鹿野裕明, 木村啓二, 笠原博徳: マルチコアプロセッサにおけるコンパイラ制御低消費電力化手法, 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.SIG12 (ACS15), pp.147-157 (2006).
- 13) 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).
- 14) Banino, C., Beaumont, O., Carter, L., Ferrate, J., Legrand, A. and Rober, Y.: Scheduling Strategies for Master-Slave Tasking on Heterogeneous Processor Platforms, *IEEE Trans. Parallel and Distributed Systems*, Vol.15, No.4 (2004).
- 15) Topcuoglu, H., Hariri, S. and Wu, M.-Y.: Task scheduling algorithms for heterogeneous processors, *Proc. Heterogeneous Computing Workshop* (1999).
- 16) UZURA3: MPEG1/LayerIII encoder in FORTRAN90. http://members.at.infoseek.co.jp/kitaurawa/index_e.html
- 17) Arakawa, F., Nishii, O., Uchiyama, K. and Norio, N.: SH4 RISC Multimedia Microprocessor, *IEEE Micro*, Vol.18, No.2 (1998).
- 18) 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)*, pp.393-408 (2006).
- 19) 田中, 津野田, 秋田, 高田, 伊藤, 佐藤: 再構成型プロセッサ FE-GA のオーディオ処理への応用, 信学技報 RECONF2005-67 (2005).
- 20) Yamada, T., Abe, M., Nitta, Y., Ogura,

K., Kusaoke, M., Ishikawa, M., Ozawa, M., Takada, K., Arakawa, F., Nishii, O. and Hattori, T.: Low-Power Design of 90-nm SuperH Processor Core, *Proc. 2005, Int'l. Conf. on Computer Design (ICCD'05)* (2005).

- 21) Renesas: High Performance Embedded Workshop (HEW). <http://www.renesas.com/hew>.
- 22) Kawaguchi, H., Shin, Y. and Sakurai, T.: μ ITRON-LP: Power-Conscious Real-Time OS Based on Cooperative Voltage Scaling for Multimedia Applications, *IEEE Trans. Multimedia* (2005).
- 23) Naffziger, S., Stackhouse, B. and Grutkowski, T.: The Implementation of a 2-core Multi-Threaded Itanium-Family Processor, *Proc. IEEE International Solid-State Circuits Conference (ISSCC 2005)* (2005).

(平成 18 年 10 月 10 日受付)

(平成 19 年 1 月 16 日採録)



鹿野 裕明 (学生会員)

昭和 52 年生。平成 12 年中央大学理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理工学研究科情報・ネットワーク専攻博士課程入学。平成 19 年情処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。



鈴木 裕貴

平成 16 年早稲田大学理工学部電気電子情報工学科卒業。平成 18 年同大学大学院修士課程修了。同年 (株) NTT データ入社、現在に至る。



和田 康孝 (正会員)

昭和 54 年生。平成 14 年早稲田大学理工学部電気電子情報工学科卒業。平成 16 年同大学大学院修士課程修了。平成 16 年同大学院博士課程進学。平成 18 年早稲田大学理工学部助手、現在に至る。



白子 準 (正会員)

昭和 54 年生。平成 14 年早稲田大学理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 年早稲田大学理工学部助手。平成 19 年学振特別研究員 PD、現在に至る。



木村 啓二 (正会員)

昭和 47 年生。平成 8 年早稲田大学理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 年早稲田大学理工学部助手。平成 16 年同大学理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 年早稲田大学理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 年～60 年早稲田大学助手。昭和 60 年学振第 1 回特別研究員。昭和 61 年早稲田大学理工学部専任講師。昭和 63 年同助教授。平成 9 年同教授。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 年情処坂井記念特別賞。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。