Vol.2012-ARC-199 No.14
2012/3/28

メモリの冗長化とパワーゲーティングによる高信頼・低消費電力・高性能実装の提案

大友俊也[†]　栗原康志[†]　寺西佑太[††]
山内宏真[†]　鈴木貴久[†]　山下浩一郎[†]

階層メモリ型 CMP のシステム設計において，メモリの冗長化とパワーゲーティングを組み合わせることで高信頼性・低消費電力・高性能を実現するアーキテクチャを提案し，その効果検証を行うシミュレータを開発した．

High-reliability, Low-power and High-performance Implementation by Memory Redundancy and Power Gating

Toshiya Otomo[†]　Koji Kurihara[†]　Yuta Teranishi[††]
Hiromasa Yamauchi[†]　Takahisa Suzuki[†]
Koichiro Yamashita[†]

This paper proposes a high-reliability, low-power and high-performance architecture which combines power gating with memory redundancy. We also developed a simulator to evaluate the proposed method.

1. はじめに

組み込み SoC 分野で用いられる階層メモリ型 CMP (Chip Multi-Processor) のシステム設計において，メモリの冗長化とパワーゲーティングを組み合わせることで高信頼・低消費電力・高性能を実現するアーキテクチャを提案し，その効果検証を行うシミュレータを開発した. アプリケーションの特性と提案手法の親和性を整理した結果，効果の顕著なものについては信頼性を確保しつつ実行性能を約 25%，消費エネルギーを約 32%改善できることが確認された．

2. 課題

近年の商用組み込みシステムは，ハードウェア・ソフトウェアともに機能部品の組合せで出来ている．例えば，SoC は各社の IP コアとこれらの間を繋ぐインターフェースで構成され，ソフトウェアは汎用アプリケーション（実験向けベンチマーク等ではなく, 製品に搭載されているような意味のあるアプリケーション）や OSS (Open Source Software)の組合せで構成されている．

OSS と汎用アプリケーションの組合せ利用によりソフトウェアの肥大化・複雑化が進んでおり，本来独立であるはずのソフトウェア・プロセス同士が意図せずして相互に依存してしまうことがある．そこで，あるプロセスで発生した障害が他のプロセスやシステム全体に影響を及ぼすといった危険からシステムを守る仕組みが必要となる．

また，近年では携帯機器をはじめ様々な分野でプロセッサの低消費電力化が強く求められており, 低消費電力化技術として利用していない CPU やメモリの電源を遮断するパワーゲーティングや，CPU の動作周波数と駆動電圧を動的に制御する DVFS 技術などが実用化されている．しかし，前述のとおり現実の商用システムではソフトウェア同士が非明示的な依存関係にあるため，これらのハードウェア低消費電力化技術を効果的に利用できていない．

商用組み込みシステム設計においては，信頼性・省電力性，処理性能をバランスよく実現することが重要となる．

3. 関連研究

従来，メモリの信頼性を高める技術として，符号化された冗長情報を物理的に分れているメモリに分散配置する手法[1]がある．これにより，１つの物理メモリの完全な故障に対して，その他の物理メモリに分散している冗長情報から失われたデータを復

† 株式会社富士通研究所
FUJITSU LABORATORIES LTD.
†† 富士通九州ネットワークテクノロジーズ株式会社
Fujitsu Kyushu Network Technologies Limited

ⓒ 2012 Information Processing Society of Japan

元することが出来る．

また，メモリの消費電力を低減する技術として，メモリアロケーション時にメモリの物理的な隙間を埋めるように工夫することで，物理的に空いているメモリ領域をゲーティングする手法[2]がある．この手法により使用していないキャッシュやローカルメモリはデータの退避を行うことなく電源遮断することが可能となる．

また，メモリ利用を最適化する研究として，大量にデータアクセスを行う部分のコードを最適化することで SDRAM のような下位のメモリアクセスを削減する手法[3]がある．この手法ではキャッシュやローカルメモリの利用効率が向上するためプログラム実行時間を改善出来る．

4. 提案手法

本稿は，以下の３つの提案から構成されている．

- 信頼性を高める冗長化コントローラ
  プロセッサの内部バスに接続され，メモリの冗長化とデータ消失時のデータ復元を行うハードウェア．

- 冗長化コントローラと連動する電源遮断機構
  CPU と，CPU と対になるキャッシュ／ローカルメモリを１つのパワードメインとしてゲーティングを行う電源遮断機構．通常，ゲーティングを行う際はメモリの内容を退避する必要があるが，PMU（Power Management Unit）だけでなく前述の冗長化コントローラと連動することで，ゲーティングの際の中間遷移状態（図 2 を参照）をスキップする機能を持つ．

- 最適なバランスを求めるシミュレータ
  タスクグラフを入力とし，任意のパラメータで提案手法の動作をシミュレートする．出力は CPU ごとの状態遷移ログと消費エネルギーである．状態遷移ログを可視化するツールもあわせて開発した．

提案アーキテクチャを図 1 に示す．

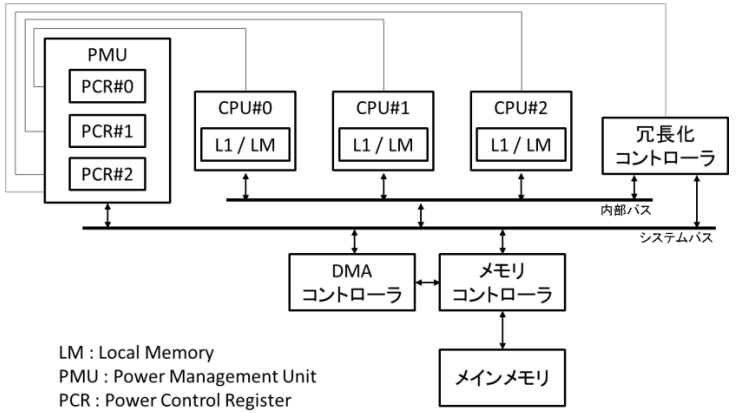

図 1 提案アーキテクチャ

4.1 提案手法の動作状態遷移

提案手法では，冗長化コントローラの働きにより CPU とメモリのゲーティングを容易に行えるようになる．従来と比較した提案手法の CPU パワーステート遷移の様子を図 2 に示す．

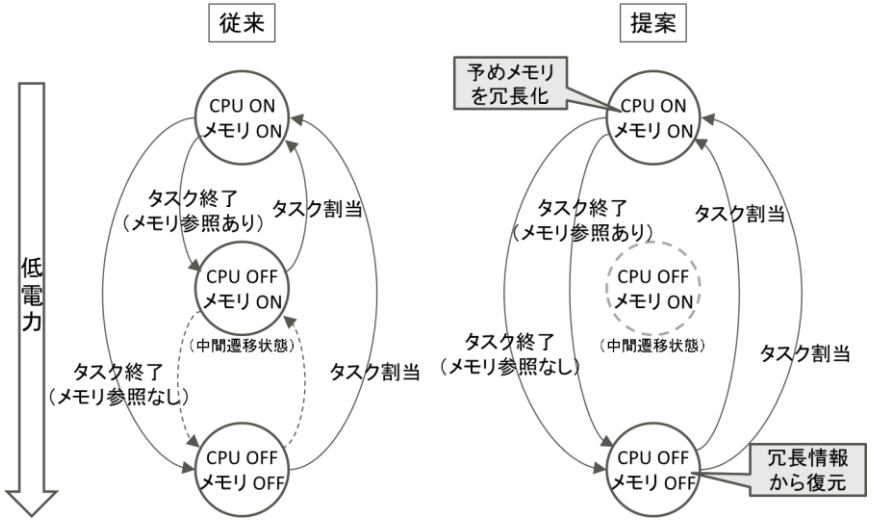

図 2 CPU パワーステート遷移図

ⓒ 2012 Information Processing Society of Japan

従来では，最も低電力なステートに遷移するためには，他 CPU から自ローカルメモリへの参照を無くす，すなわち依存関係を無くす必要があった．例えば，依存関係を引き継ぐため，ローカルメモリのデータをメインメモリへ退避するなどの措置が必要であった．

一方，提案手法では他 CPU からの参照があっても，予めメモリを冗長化しているため最も低電力なステートに直接遷移することができる．冗長化コントローラの動作によって消費するエネルギーは増えるが，消費電力の大きい CPU とローカルメモリのゲーティングを行える機会が増える．

タスク終了・タスク割当によってパワーステートの遷移が起きた時の時系列動作を図 3 に示す．

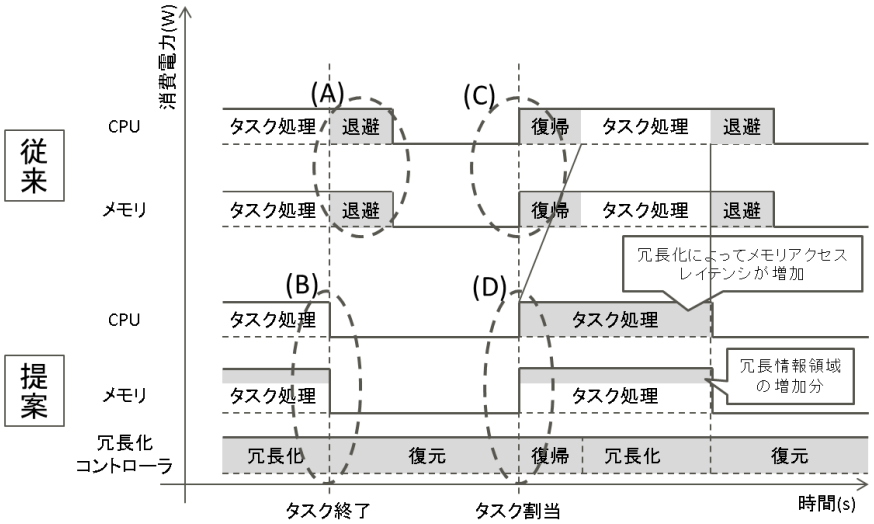

図 3 パワーステート遷移時の動作

タスク終了時，従来ではタスク終了後メモリ内容を退避した後にゲーティングを行う（図 3 の A）．一方提案手法では，予め冗長化コントローラによりメモリ内容の冗長化を行い，タスク終了後すぐにゲーティングを行う（図 3 の B）．ゲーティング後は，冗長化コントローラが冗長情報よりメモリ内容を復元する．メモリ内容の退避なしにゲーティングを行えることが提案手法の正の効果要因であるが，予めメモリの冗長化を行う必要があり，これが提案手法の負の効果要因である．

タスク割当時，従来では退避したメモリ内容を復帰させた後にタスク処理を開始する（図 3 の C）．一方提案手法では，冗長化コントローラがメモリ内容の復帰を行うため，CPU はメモリ内容の復帰を行わずにタスク処理を開始する（図 3 の D）．ただし，提案手法では冗長化コントローラの動作による影響でメモリアクセス・レイテンシが増加する．タスク割当後すぐにタスク処理を開始できることが提案手法の正の効果要因であり，メモリアクセス・レイテンシが増加することが提案手法の負の効果要因である．

4.2 冗長化コントローラ

提案手法では，冗長化コントローラがローカルメモリのデータから冗長情報を生成し，この冗長情報をローカルメモリに分散して配置する．ローカルメモリの構成と冗長化コントローラの動作を図 4 に示す．

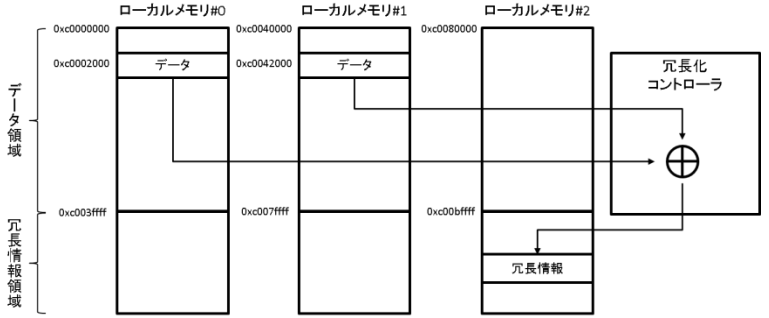

図 4 ローカルメモリの構成と冗長化コントローラの動作例

ローカルメモリはデータ領域と冗長情報領域とに分かれており，データ領域はシステムで一意のアドレスがマップされている．図 4 では，冗長化コントローラはローカルメモリ#0 の 0xc0002000 番地のデータとローカルメモリ#1 の 0xc0042000 番地のデータの排他論理和を取って冗長情報を生成し，ローカルメモリ#2 の冗長情報領域に書き込んでいる．

以下，動作状態に応じた冗長化コントローラの動作について説明する．

4.2.1 通常状態

通常状態では，全ての CPU がアクティブでそれぞれタスクを実行している．冗長化コントローラはローカルメモリへの書込みを監視しており，書込みがあった場合，変更されたデータとこのデータと対になるデータを読込み，冗長情報を生成する．図 5 にその動作を示す．

ⓒ 2012 Information Processing Society of Japan

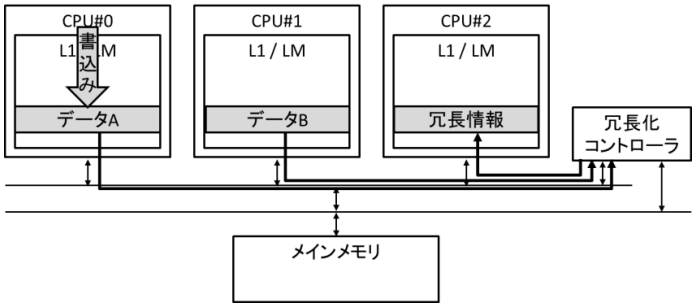

図 5 通常状態の動作

4.2.2 データ退避中状態

データ退避中状態では，アイドル状態となった CPU がゲーティングされている，あるいは障害が発生しローカルメモリが利用できない状態である．冗長化コントローラは，失われたデータを他 CPU のデータと冗長情報から復元し，メインメモリに退避している．図 6 にその動作を示す．

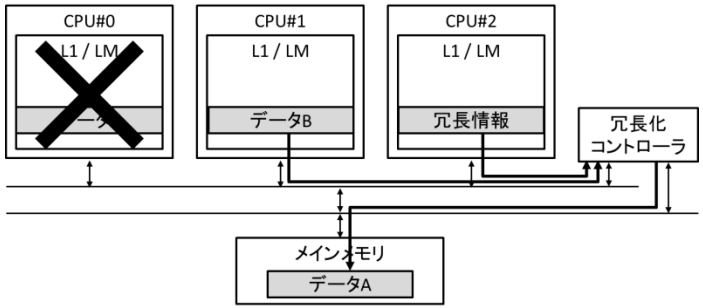

図 6 データ退避中状態の動作

ただし，提案手法を用いるには最低 3 つの CPU がアクティブでなければならない．アクティブな CPU が 2 つの時に一方の CPU をゲーティングする場合は，ゲーティングを行う前に CPU が自身のローカルメモリのデータをメインメモリへ退避する必要がある．

また，提案手法を用いない場合も同様で，アクティブな CPU をゲーティングする際は，事前に CPU が自身のローカルメモリのデータをメインメモリへ退避する．

4.2.3 データ退避完了状態

データ退避完了状態では，冗長化コントローラは利用不可能なローカルメモリへのアクセスを監視している．このメモリへの読込みがあった場合，冗長化コントローラは退避先のメインメモリからデータを読込み，アクセスを行った CPU に対して応答する．書込みがあった場合は，冗長化コントローラが CPU に代わって退避先のメインメモリに書込みを行う．図 7 に読込みの場合の動作を示す．

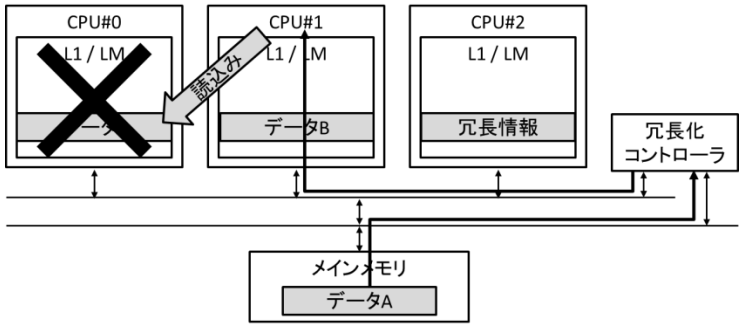

図 7 データ退避完了状態の動作

4.2.4 データ転送中状態

データ転送中状態では，ゲーティング中であった CPU がアクティブ状態に遷移している．冗長化コントローラは，退避したデータをメインメモリからアクティブになった CPU のローカルメモリへ転送を行っている．図 8 にその動作を示す．

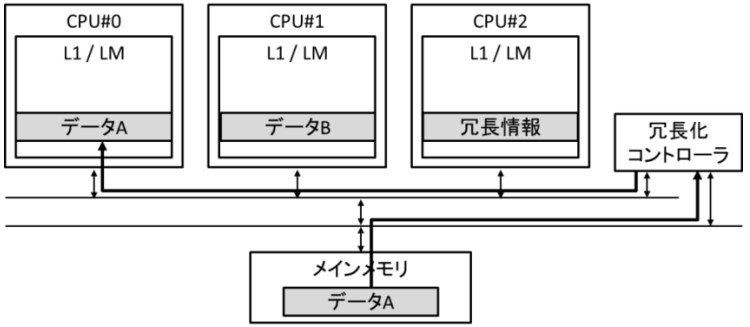

図 8 データ転送中状態の動作

ⓒ 2012 Information Processing Society of Japan

また，これらの動作を行う冗長化コントローラの構成例を図 9 に示す．

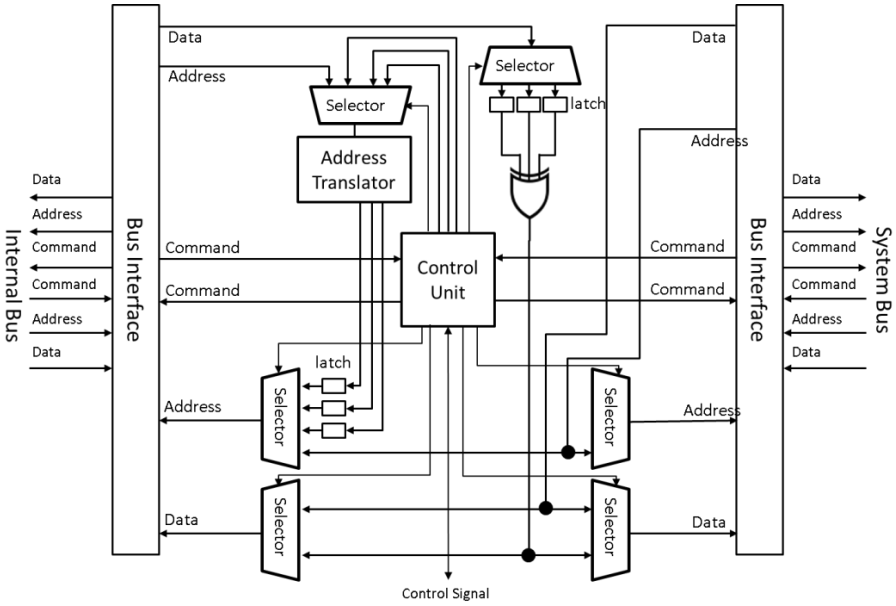

図 9 冗長化コントローラの構成例

4.3 シミュレータ

提案手法では，4.1 節で示したように正と負の効果要因がある．しかし，評価対象のハードウェアや動作させるソフトウェアのスペックによって，総合的な効果が改善するのか劣化するのかを容易に考察できない．このため，開発したシミュレータはタスクグラフを入力とし，評価対象のハードウェアパラメータを与えることで，CPU ごとの状態遷移ログと消費エネルギーを出力する．シミュレータの構成を図 10 に示す．

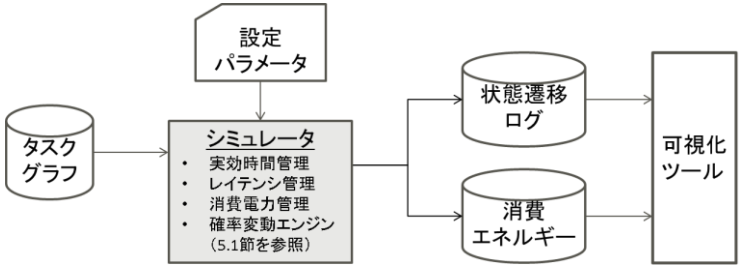

図 10 シミュレータの構成

4.3.1 タスクグラフ

タスクグラフは，タスクを表現するノードと，タスク間の依存関係を表現するエッジとで構成されるグラフである．このグラフは，アプリケーションのコンパイル段階の構造解析で得ることが出来る．タスクグラフの例を図 11 に示す．

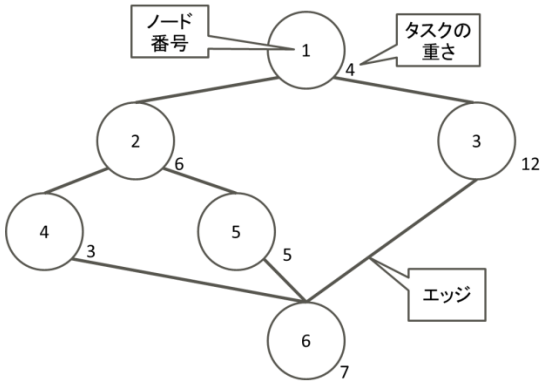

図 11 タスクグラフの例

4.3.2 パラメータ

本シミュレータで設定可能なパラメータの一覧を表 1 に示す．

表 1 設定パラメータ一覧

CPU 数
CPU 動作周波数
内部バス動作周波数
システムバス幅
ローカルメモリ容量
ローカルメモリのアクセスレイテンシ
メインメモリのアクセスレイテンシ
タスク中のメモリアクセス命令の割合
ローカルメモリアクセスの割合
CPU の消費電力
ローカルメモリの消費電力
メモリアクセスの消費エネルギー

ⓒ 2012 Information Processing Society of Japan

4.3.3 状態遷移ログ

本シミュレータは，タスクの重さをステップ数に変換し，動作状態に応じたメモリアクセス・レイテンシを積み上げていくことでタスク処理時間を求める，ソフトウェア・ステップ単位での実効時間精度を持つシミュレータである．

このシミュレータが生成する状態遷移ログの可視化結果を図 12 に示す．

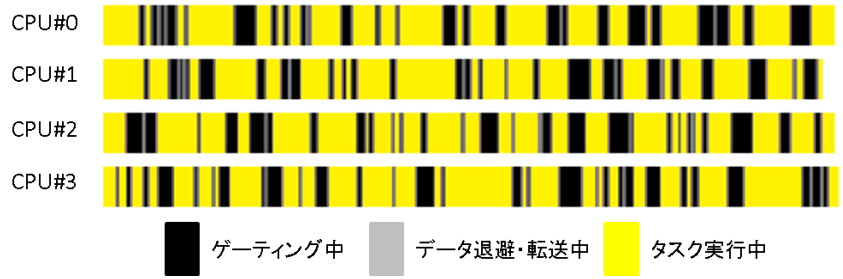

図 12 状態遷移ログ可視化の例

4.3.4 消費エネルギー

以下のようにして消費エネルギーを算出する．

- CPU，ローカルメモリの消費エネルギー

  シミュレーション結果から CPU とローカルメモリの稼働時間を取得し，これにそれぞれの消費電力を乗じることで算出している．提案手法では，データ領域のほかに冗長情報領域が必要となる．冗長情報領域の容量は次の式で求められ，これも計上する．

$$\text{冗長領域の容量} = \frac{\text{ローカルメモリの容量}}{\text{冗長構成を適用する CPU 数} - 1} \quad \cdots(1)$$

- 冗長化コントローラの消費電力

  図 9 に示す回路を構成するトランジスタ数は数千～1 万数千程度であり，40nm プロセスルールを想定するとこの消費電力は数十～数百 μW 程度と予想される．CPU やメモリに比べ十分に小さいため，冗長化コントローラは常時動作するものとしてこの消費電力は定数とした．

- 冗長化コントローラによるメモリアクセスの消費電力

  提案手法では，冗長化コントローラの働きにより従来手法に比べメモリアクセスが増加する．動作状態に応じて増加したメモリアクセスの消費エネルギーを算出し，計上した．

5. 評価

開発したシミュレータを用いて提案手法の有効性を検証した．入力には公開ベンチマーク「STG（Standard Task Graph set）[4]」を使用し，タスク構成の違う 180 本のタスクグラフをシミュレーションした．タスク構成の特徴において提案手法の効果の傾向をベンチマークした．

5.1 パラメータ

今回のシミュレーションで用いた想定ハードウェアのパラメータを表 2 に示す．

表 2 評価に用いたパラメータ

CPU 数	4
CPU 動作周波数	1GHz
内部バス動作周波数	500MHz
システムバス幅	32bit
ローカルメモリ容量	256KB
ローカルメモリのアクセスレイテンシ	5 サイクル
メインメモリのアクセスレイテンシ	100 サイクル
(*)タスク中のメモリアクセス命令の割合	~ 50%
(*)ローカルメモリアクセスの割合	~ 90%
CPU の消費電力	1W / 個
ローカルメモリの消費電力	0.5mW / KB
メモリアクセスの消費エネルギー	1nJ / 回

タスク中のメモリアクセス命令の割合は，様々なベンチマークやアプリケーションに含まれる命令を分析した論文[5]を参考に設定した．ただし，(*)印の項目については実行アプリケーションレベルでは一定値を取るとは限らないため，実機・実アプリケーションから取得されるプロファイル情報をもとにした確率分布を与える．

5.2 評価結果

180 本のタスクグラフのうち，特徴的な結果となった rand0000.stg と rand0177.stg の実行結果を以下に示す．

rand0000.stg は，並列性があまり多くないタスクグラフで平均の並列度が 4 程度となっている．タスク処理時間を図 13 に，消費エネルギーを図 14 に，それぞれ示す．

ⓒ 2012 Information Processing Society of Japan

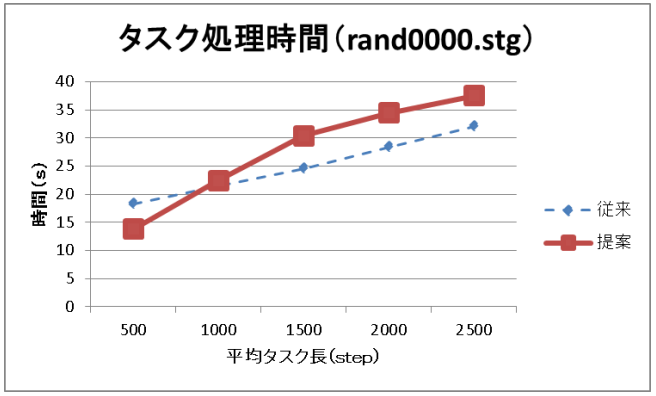

図 13 rand0000.stg のタスク処理時間

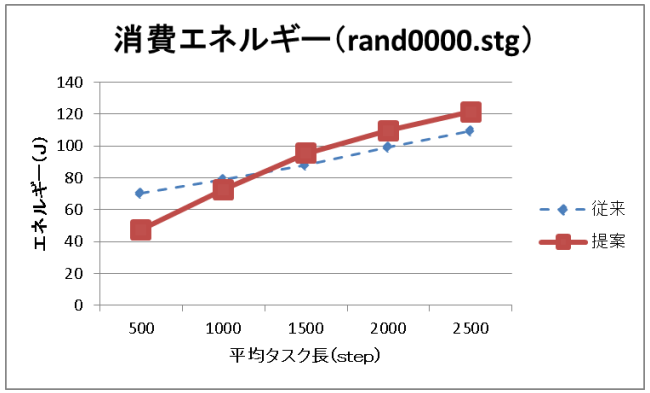

図 14 rand0000.stg の消費エネルギー

図 13 は, タスクグラフ rand0000.stg の平均タスク長を変化させた時のタスク処理時間を示している．平均タスク長が 500step の時，提案手法は従来手法に対して約 25% タスク処理時間を短縮しているが，平均タスク長が 1000step を超えると逆転する．

図 14 は, タスクグラフ rand0000.stg の平均タスク長を変化させた時の消費エネルギーを示している．平均タスク長が 500step の時，提案手法は従来手法に対して約 32% 消費エネルギーを削減しているが，平均タスク長が 1500step を超えると逆転する．

一方 rand0177.stg は，並列性が非常に高いタスクグラフで，平均の並列度が 130 程度となっている．タスク処理時間を図 15 に，消費エネルギーを図 16 に，それぞれ示す．

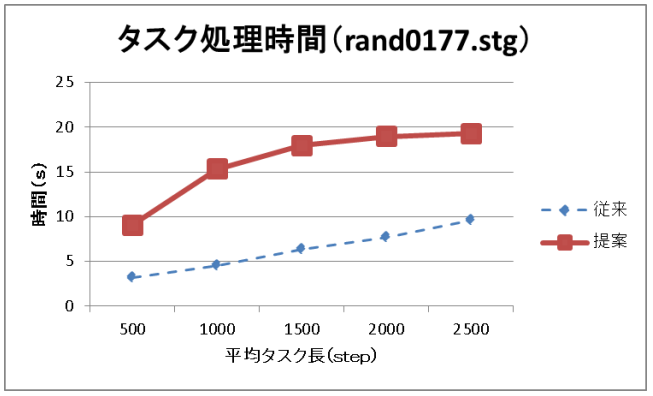

図 15 rand0177.stg のタスク処理時間

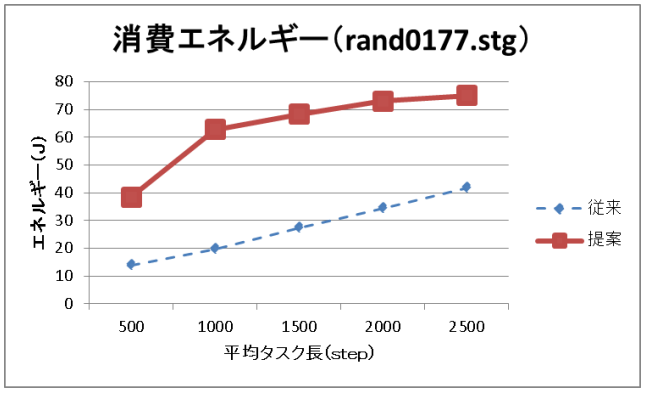

図 16 rand0177.stg の消費エネルギー

図 15 は, タスクグラフ rand0177.stg の平均タスク長を変化させた時のタスク処理時間を示している．このタスクグラフで測定範囲では平均タスク長によらず従来手法が短い時間でタスク処理を終了している．

図 16 は, タスクグラフ rand0000.stg の平均タスク長を変化させた時の消費エネルギーを示している．このタスクグラフで測定範囲では平均タスク長によらず従来手法が少ないエネルギーでタスク処理を行っている．

5.3 考察

rand0000.stg は並列度が比較的小さい, 言い換えればタスク間の依存が多いタスクグ

ⓒ 2012 Information Processing Society of Japan

ラフである．そのため，同期待ちなどによりアイドル状態となる CPU が多く発生し，ゲーティングが頻繁に起こっていると考えられる．提案手法はゲーティング時に正の効果が生じるため，このようなタスクグラフを得意とする．しかし，図 13, 図 14 によると, タスクの粒度を大きくすると提案手法の優位性は失われていくことが分かる．タスクの粒度が大きくなることで CPU がタスク処理を行っている時間が長くなるが，提案手法の冗長化による負の効果は CPU がタスク処理を行っている時間が長ければ長いほど増える．このため，平均タスク長が長くなることで提案手法の負の効果が正の効果を超えてしまっていると考えられる．

rand0177.stg は並列度が大きい, 言い換えるとタスク間の依存が少ないタスクグラフである．そのため，同期待ちなどによる CPU のアイドル状態が発生しづらくゲーティングがあまり行われない．前述のとおり，提案手法はゲーティング時に正の効果がある．そのため，rand0177.stg のようなタスクグラフは提案手法には向いていないと考えられる．

以上をまとめると，図 17 のようになる．

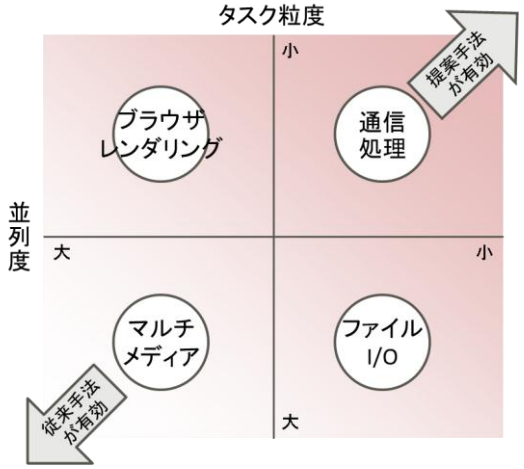

図 17 アプリケーション特性と提案手法の親和性

並列度が小さくタスク粒度が細かいアプリケーションとして，例えば通信パケット処理などが挙げられる．並列度が小さくタスク粒度が粗いアプリケーションとして，例えばファイル I/O などが挙げられる．

一方，並列度が大きくタスク粒度が細かいアプリケーションとして，例えばブラウザが挙げられる．また，並列度が大きくタスク粒度が粗いアプリケーションとして，マルチメディア処理が挙げられる．

6. まとめ

本稿では，階層メモリ型 CMP のシステム設計において，メモリの冗長化とパワーゲーティングを組み合わせたアーキテクチャを提案し，開発したシミュレータを用いて提案手法の効果検証を行った．その結果，提案手法は実行するタスクの粒度が細かくタスク間の依存関係が多い場合に有効であることが分かったが，アプリケーションの性質によっては提案手法が有効でないことも分かった．

そこで今後は，どのような条件で提案手法が有効となるかを定量的に表現する判別式の立式，提案手法と従来手法を動的に切り替える手法，ランタイムで提案手法と従来手法を切り替えるための実装方法，をそれぞれ検討することで，どのような性質のアプリケーションに対しても高信頼・低消費電力・高性能を実現するシステムを目指す．

参考文献

1) Timothy J. Dell, “A White Paper on the Benefits of Chipkill-Correct ECC for PC Server Main Memory,” IBM Microelectronics Division, 1997.
2) V. De La Luz, M. Kandemir, I. Kolcu, “Reducing Memory Energy Consumption of Embedded Applications That Process Dynamically Allocated Data”, Computer-Aided Design of Integrated Circuits and Systems, pp.1855-1860, 2009.
3) Shan Li, Edmund M-K. Lai, Mohammed Javed Absar, “Minimizing Embedded Software Power Consumption Through Reduction of Data Memory Access”, International Conference on Information, Communications and Signal Processing(ICICS), Vol.1, pp.309-313, 2003.
4) “Standard Task Graph Set” Home Page
http://www.kasahara.elec.waseda.ac.jp/schedule/
5) Matthew R. Guthaus, Jeffrey S. Ringenberg, Dan Ernst, Todd M. Austin, Trevor Mudge, Richard B. Brown, “MiBench: A free, commercially representative embedded benchmark suite,” Workload Characterization, pp.3-14, 2001.

ⓒ 2012 Information Processing Society of Japan