

研究種目：若手研究（B）  
 研究期間：2006～2008  
 課題番号：18700042  
 研究課題名（和文） 投機技術を積極的に利用するチップマルチプロセッサに関する研究  
 研究課題名（英文） Research on the chip multiprocessors using aggressive speculative technology  
 研究代表者  
 吉瀬 謙二（KISE KENJI）  
 東京工業大学・大学院情報理工学研究科・講師  
 研究者番号：50323887

## 研究成果の概要：

マルチプロセッサに適した投機技術の一つとして、従来はハードウェアでおこなっていた分岐予測をソフトウェアで実現するソフトウェアの性能改善に取り組んだ。特に、複数の予測手法の適応可能性について検討し、個別の分岐命令毎に予測手法を変更することで予測精度を改善できることを明らかにした。チップマルチプロセッサ時代のコアの利用法の一つとして、ソフトウェアによりキャッシュを実現するキャッシュコアと呼ぶ技術を提案・評価をおこなった。この他にも、動的に計算能力を変更することを目指した CoreSymphony 機構を提案し、その性能を定量的に明らかにした。

## 交付額

(金額単位：円)

	直接経費	間接経費	合計
2006年度	1,400,000	0	1,400,000
2007年度	1,300,000	0	1,300,000
2008年度	700,000	210,000	910,000
年度			
年度			
総計	3,400,000	210,000	3,610,000

## 研究分野：総合領域

科研費の分科・細目：情報学 計算機システム・ネットワーク

キーワード：チップマルチプロセッサ，投機技術，マルチコア

## 1. 研究開始当初の背景

(1) 半導体技術の進歩により、10億トランジスタを超える大規模なハードウェア資源を活用できる時代が迫っている。我々は、大規模なハードウェア資源をシングルスレッドのアプリケーションの高速化のために利用することを検討してきた。具体的には、命令レベル並列性の抽出による高速化を目指して、データ値予測、分岐予測などの投機技術、制御の流れに注目するスーパー命令フローアーキテクチャの検討を進めてきた。

(2) 一方、現状の市場をみると、プロセッサチップに2個程度のプロセッサコアを搭載するチップマルチプロセッサの構成を持つチップが提供され始めている。また、1つのチップに16個から64個程度のプロセッサコアを搭載する構成が現実的な選択肢として議論されている。高性能なチップマルチプロセッサを実現するためには、プロセッサコアの構成方式、集積するプロセッサコアの個数、プロセッサコアを接続するネットワーク構

成、共有キャッシュの方式などを考慮した最適化が必要となる。また、従来のプロセッサアーキテクチャにおいて検討されている投機技術の有効性をチップマルチプロセッサの構成において再検討する必要がある。

## 2. 研究の目的

(1) 速度と消費電力とのトレードオフに注目しながら、投機技術を積極的に利用するチップマルチプロセッサの方式を検討することを本研究の目的とする。一般的な RISC 方式を利用するプロセッサコアに加えて、大規模なプロセッサコアを構成要素とするヘテロなチップマルチプロセッサの高性能化を試みる。

## 3. 研究の方法

(1) プロセッサ設計の上位工程において利用される詳細なプロセッサシミュレータを構築し、これを利用してチップマルチプロセッサの性能に与えるプロセッサコアの方式及び投機技術の影響を定量的に評価する。

## 4. 研究成果

(1) 本研究における基盤環境として、スカラプロセッサのサイクルレベルのプロセッサシミュレータの実装をおこなった。また、これらを拡張してチップマルチプロセッサの性能評価のための幾つかのシミュレータの構築に成功した。

その1つとして取り組んだ成果が、チップマルチプロセッサ研究のインフラストラクチャとして、Cell Broadband Engine をモデルとしたプロセッサシミュレータ SimCell の開発である。これは、ソースコードのシンプルさを優先させており、その実用性は高い。また、SPE の数に着目したシミュレーション等の結果から、シミュレーション時間、可視化ツールによるローカルストアの表示例を示すことで、SimCell の実用性を明らかにした。

また、SimCell のシミュレーション例から、ネットワークの構成を変更しながら SPE を 512 個まで増加させたときのプロセッサ性能を評価し、想定したシミュレーション結果が得られること、512 個のシミュレーションが問題なく行えることを確認した。SimCell によるシミュレーション時間について検証し、0.2~5 MIPS で動作することを確認した。最適化を施していないため、高速という訳ではないが、アプリケーションのデータセットを小さくするといった使い方の工夫により十分に実用的である。

(2) マルチプロセッサに適した投機技術の

一つとして、従来はハードウェアでおこなっていた分岐予測をソフトウェアで実現するソフトウェアの性能改善に取り組んだ。特にハードウェア分岐予測機構をもたないプロセッサに注目し、分岐ミスを削減するために分岐ヒント命令を用いたソフトウェア分岐予測を提案した。また、ハードウェア分岐予測機構をもたないプロセッサを対象に、適切な予測コード挿入のための分岐命令プロファイリング手法と、予測コードの挿入機会を高めるアーキテクチャ技術である Dual-Hint 機構を提案した。評価の結果、予測コードの挿入前と比較してプロファイリングにより平均 18%、Dual-Hint 機構により平均 22%の性能向上を達成することができた。

(3) チップマルチプロセッサ時代のコアの利用法の一つとして、ソフトウェアによりキャッシュを実現するキャッシュコア (図 1) と呼ぶ技術を提案・評価をおこなった。

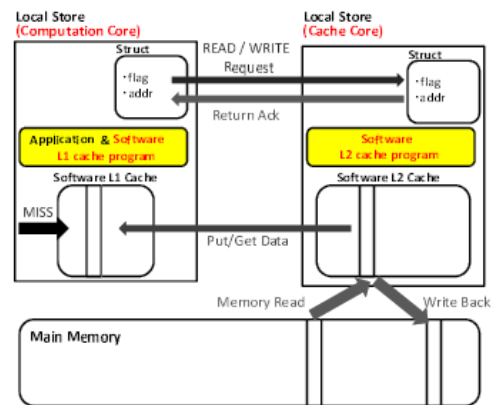


図 1 キャッシュコアアーキテクチャ

マルチコアプロセッサではアプリケーションの並列性を高め、複数のコアに上手にタスクを配分することで処理の高速化を望むことができる。しかし、この方法ではコアの増加にアプリケーションの並列性が追いつかなければ、期待される速度向上を得ることはできない。またメインメモリへのデータアクセスの集中がボトルネックとなり、性能がコア数の増加と共に線形に増加しない。

キャッシュコアでは、ソフトウェア実装のキャッシュプログラムを実行する。また、アプリケーションを実行するコア (計算コア) の L2 データキャッシュとして機能する。すなわち、キャッシュコアは計算コアに対してデータを供給することで、通信レイテンシの大きいメモリアクセスを減少させ、処理速度の向上を目指している。

シミュレータを駆使して、プロセッサ内部の通信レイテンシを変化させて提案方式のキャッシュコアを評価することにより、キその

可能性を明らかにした。

最適化を行ったキャッシュコアを実機としてのマルチコアプロセッサを搭載する Cell/B. E. と独自開発の M-Core アーキテクチャ上で評価した。その結果、コア間レイテンシの大きい Cell/B. E. では、キャッシュコア使用により、平均で 5% の性能低下が見られた。一方、コア間レイテンシの小さい、M-Core アーキテクチャでは平均で 25% という高い性能向上が見られた。この 2 つの結果から、コア間レイテンシが平均的なマルチコアプロセッサにおいて、キャッシュコアが有効であることが示された。さらに Cell/B. E. 上で Victim キャッシュをキャッシュコアに追加した。その結果、追加前と比較して平均 10% の性能向上が見られ、キャッシュコアを使用しない場合と比較して、平均 5% の性能向上を達成した。

最後に、メモリアクセス集中時におけるキャッシュコアの評価を行った。その結果、キャッシュコア使用により最大 2.0 倍、平均 1.6 倍の性能向上を確認した。また、メモリ負荷量とキャッシュコアの相対性能の関係を調べ、負荷量に応じてほぼ線形にキャッシュコアの性能が向上していることを確認した。これらの調査から、メモリアクセス集中時におけるキャッシュコアの有効性を示すことができた。

(4) この他にも、コア同士の協調(融合)によりスレッドあたりの実行能力を高める CoreSymphony アーキテクチャを提案した。このアーキテクチャでは、2 命令/cycle 程度の狭い発行幅のコアを多数有するメニーコアプロセッサを想定する。各コアは均質であり、スレッドレベルの並列性が十分な場合には標準的なチップマルチプロセッサ(CMP)と同様に動作する。しかし、十分なスレッドレベル並列性が得られない場合、あるいはシングルスレッド性能を要求するタスクが存在する場合には、幾つかのコアが融合し、あたかも発行幅の大きな一つのコアであるかのように振舞う。これにより、命令レベル並列性を積極的に抽出し、スレッドあたりの性能向上を図る。融合するコアの個数は、融合可能な最大個数以内で任意に選択することが可能であり、また融合/分離は動的に行われるため、様々な粒度のタスクへの柔軟な対応が可能となる。

この検討では、CoreSymphony の基本方針を明確にするとともに、2 命令発行のコアを 16 個もつプロセッサを想定して CoreSymphony アーキテクチャをソフトウェアシミュレータとして実装した。5 つのベンチマークを用いた評価の結果、1 コア時に比べ 4 コアの融合で平均 110%、最大 167% の命令レベル並列性の向上を達成した。同等の発行幅を有する標

準的なアウトオブオーダー・スーパースカラとの比較では 8 命令発行時の IPC の低下を平均で 23% に抑えることができた。また、コア間の通信のレイテンシを変化させて評価し、通信レイテンシが 3 サイクル程度の現実的な遅延とした場合でも、融合により性能が向上することを確認できた。

## 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 1 件)

① Shimpei Sato, Naoki Fujieda, Akira Moriya, and Kenji Kise: SimCell: A Processor Simulator for Multi-Core Architecture Research, IPSJ Transactions on Advanced Computing Systems, Vol. 2, No. 1, pp. 146-157 (March 2009). 査読有

[学会発表] (計 10 件)

① 森洋介, 森谷章, 吉瀬謙二, マルチコアプロセッサの高速化を目指したキャッシュコアの最適化, 先進的計算基盤システムシンポジウム SACSIS2009 論文集, 2009 年 5 月 29 日, 広島国際会議場

② 森谷章, 藤枝直輝, 佐藤真平, 吉瀬謙二, メニーコアプロセッサに向けたデータ供給を支援する多機能キャッシュコア, 先進的計算基盤システムシンポジウム SACSIS2008 論文集, 2008 年 6 月 13 日, 筑波国際会議場

③ 若杉祐太, 吉瀬謙二, メニーコアプロセッサに向けたシンプルで柔軟なコア融合機構 CoreSymphony, 先進的計算基盤システムシンポジウム SACSIS2008 論文集, 2008 年 6 月 13 日, 筑波国際会議場

④ 森洋介, 森谷章, 藤枝直輝, 吉瀬謙二, マルチコアにおけるオーバーヘッド解析を用いたキャッシュコアの最適化, 情報処理学会研究報告 2009-ARC-181, pp.105-110, 2009 年 1 月 14 日, パナソニック株式会社松心会館

⑤ Shimpei Sato, Naoki Fujieda, Akira Moriya, and Kenji Kise: Processor Simulator SimCell to Accelerate Research on Many-core Processor Architectures, Workshop on Cell Systems and Applications (WCSA 2008) held in conjunction with the ISCA-2008, 2008 年 6 月 21 日, Beijing China

⑥ 藤枝直輝, 佐藤真平, 吉瀬謙二, 二重分

岐ヒントを考慮したソフトウェア分岐予測の可能性検討, 情報処理学会研究報告 2008-ARC-177, pp. 121-126, 2008年3月5日, 北海道大学

⑦ 森谷章, 藤枝直輝, 佐藤真平, 吉瀬謙二, 多機能メニーコアにおけるデータ供給を支援するキャッシュコアの提案, 情報処理学会研究報告 2008-ARC-176, pp. 53-58, 2008年1月15日, 慶應義塾大学

⑧ 田原慎也, 吉瀬謙二, 予測精度の限界を目指すパターンマッチングを用いた分岐予測手法の検討と評価, 情報処理学会研究報告 2007-ARC-175, pp. 33-38, 2007年11月21日, 北九州国際会議場

⑨ 吉瀬謙二, 佐々木豊, Cell プロセッサの分岐ペナルティを軽減するソフトウェア分岐予測の可能性検討, 情報処理学会研究報告 2007-ARC-172, pp. 245-250, 2007年3月2日, 北海道大学

⑩ 吉瀬謙二, 現実的な構成を保ちながら大幅な予測ミスの削減を図る gshare+分岐予測, 情報処理学会研究報告 2006-EMB-2, pp. 13-18, 2006年9月27日, 産業技術総合研究所関西センター

## 6. 研究組織

### (1) 研究代表者

吉瀬 謙二 (KISE KENJI)  
東京工業大学・大学院情報理工学研究科・講師  
研究者番号: 50323887

### (2) 研究分担者

なし

### (3) 連携研究者

なし