計算機アーキテクチャ 150－9
（2002．11．28）

マルチバンク構成レジスタファイルを用いたスーパスカラプロセッサの構成方式に関する検討

三谷陽介[†] 平松健[††,*] 内田裕志[†††]
弘中哲夫[††] マタウシュハンスユルゲン[†††] 小出哲士[†††]

スーパスカラ，SMTなどの技術による並列度の増加に伴い，レジスタファイルのポート数，エントリ数は増加する傾向にある．ポート数やエントリ数の増加は，面積増大，アクセスタイムの増加といった問題を引き起こす．この問題に対処する方法として，高速かつ小面積で構成できるマルチバンク構成レジスタファイルの利用を考え，従来のマルチポートレジスタファイルに比べて70%の面積削減，48.6%のアクセスタイム削減を実現した．しかし，マルチバンク構成レジスタファイルには，バンクのアクセス競合による性能低下の可能性がある．本稿では，マルチバンク構成レジスタファイルのためのアクセス競合回避手法を提案し，シミュレーションによる性能評価を行う．評価結果より，提案手法を適用することで，マルチバンク構成レジスタファイルを通常のマルチポートレジスタファイルの代わりに利用してもほぼ同等の性能を得られることがわかった．

Access Conflict Resolution Methods for Superscalar Processors with Multi-Bank Register File

YOSUKE MITANI,[†] TAKESHI HIRAMATSU,[††,*]
HIROSHI UCHIDA,[†††] TETSUO HIRONAKA,[††]
MATTAUSCH HANS JÜRGEN[†††] and TETSUSHI KOIDE[†††]

Superscalar and SMT processors require a large register file which increases chip size and access time. To solve this problem, we use multi-bank register file which reduces 70% in chip size and 48.6% in access time. However, multi-bank regsiter file cause performance degradation by bank conflictions. This paper proposes access conflict resolution methods for superscalar processors with multi-bank register file in order to reduce bank conflicts. Our experimental evaluation shows that a multi-bank register file supported by access confilct resolution methods can achieve nearly the same performance as a ideal multi-port register file.

1. はじめに

計算機の性能向上の手法として，機械命令レベルの並列性を利用するスーパスカラプロセッサや，スレッドレベルの並列性を利用するSMT(Simultaneous Multithreading)などの技術が利用されている．

同時発行命令数や同時走行スレッド数は今後も増加すると考えられ，それに伴って必要なレジスタファイルのポート数やエントリ数も増加する．特にSMTプロセッサにおいては，スレッドごとにレジスタファイルを用意する必要があるため，大容量のマルチポートレジスタファイルが必要となる．例えば，8-issue/4-SMTをサポートするプロセッサの設計では，512エントリのレジスタファイルを備えている[1)]．

マルチポートレジスタファイルのポート数，エントリ数の増加により，面積，アクセスタイム，消費電力の増大といった問題が生じる．これらの問題に対処する方法の一つに，レジスタファイルをバンク化したマルチバンク構成レジスタファイルの利用が考えられる．このレジスタファイルを用いることで，ポート数を増加させることなくエントリ数を増やすことができる．本稿では，マルチバンク構成レジスタファイルを用いたスーパスカラプロセッサの構成方式を提案する．

本稿の構成を次に示す．2節では，従来のマルチポートレジスタファイルの問題点について説明する．3節では，マルチバンク構成レジスタファイルの構成を示す．4節では，マルチバンク構成レジスタファイルをスーパスカラプロセッサに適用するための方式について提案する．5節では，提案方式についての定量的な評価結果を示す．6節では，関連研究について説明する．最後に7節でまとめを行う．

2. マルチポートレジスタファイルの問題点

マルチポートレジスタファイルのポート数およびエントリ数を増加させた場合に生じる問題として，面積，アクセスタイム，消費電力の増大がある．それぞれについて次項で説明する．

2.1 面　積

一般に，レジスタファイルを構成するマルチポートメモリの面積はポート数の2乗に比例して増大する[3)4)]．このため，12ポートのレジスタファイルは1ポートレジスタファイルの15倍の面積となる．従って，ポート数が増加すればするほど，ハードウェア面積の劇的な増大が実現上の大きな問題となる．

† 広島市立大学大学院 情報科学研究科 情報工学専攻
Graduate School of Information Sciences, Hiroshima City University
†† 広島市立大学 情報科学部 情報工学科
Faculty of Information Sciences, Hiroshima City University
††† 広島大学ナノデバイス・システム研究センター
Research Center for Nanodevices and Systems, Hiroshima University
* 現在，(株) ソフト開発
Presently with Soft Kaihatsu Co.,Ltd.

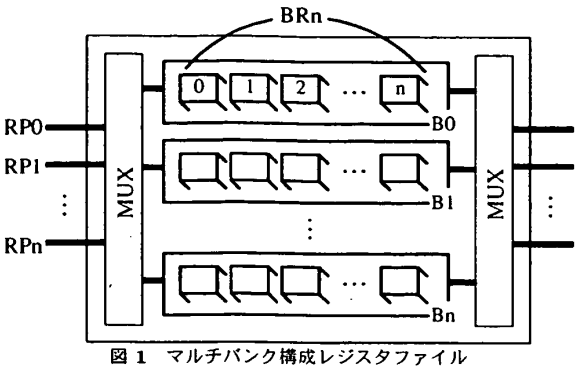


図 1 マルチバンク構成レジスタファイル

2.2 アクセスタイム

多数のポートやエントリを備えたマルチポートレジスタファイルは，内部に多数のマルチプレクサやバッファを必要とするためアクセスタイムの増加が著しい．加えて近年のプロセスでは配線遅延が支配的であり，微細化によるアクセスタイムの削減には限界がある．0.18μm プロセスにおける 80 エントリ，24 ポートのレジスタファイルの設計例では，アクセスタイムが 1.5ns を上回り，プロセッサのクリティカルパスになる可能性がある[12]．

アクセスタイムの問題を解決する方法として，レジスタファイルのパイプライン化が考えられる．しかしレジスタファイルのパイプライン化は RAM の構成から容易ではない．たとえパイプライン化しても，バイパスロジックの複雑化という別の問題が生じる[6]．また，分岐予測ミスペナルティの増加や，レジスタ寿命の増加によるレジスタプレッシャーが性能を低下させる要因となる．

2.3 消費電力

レジスタファイルの消費エネルギーはポート数の 2 乗に比例して増加するため[9]，ポート数の増加は消費電力を増加させる大きな要因となる．かつて，レジスタファイルの消費電力は無視できる程度であったが，近年のプロセッサではチップ全体の消費電力の 10%を占めるまでに上昇している[7]．また，8-issue/4-SMT プロセッサの設計例において，マルチポートレジスタファイルのリーク電力は 64KB の L1 キャッシュより数倍大きい[8]．

3. マルチバンク構成レジスタファイル

前節に示したマルチポートレジスタファイルに起因する問題を緩和する方法として，マルチバンク構成レジスタファイルの利用を検討する．マルチバンク構成レジスタファイルは，バンク化によりポートに対する配線領域を減少させ，高速かつ小面積を実現する．

3.1 構　　成

図 1 に，マルチバンク構成レジスタファイルの構成を示す．RPn はレジスタファイルのポートを表す．Bn はバンクを表し，BRn はバンク内レジスタを表す．各バンク内のレジスタに対しては，同時に複数のアクセスはできない．バンク数が多いほど同時にアクセス可能なレジスタ数は増加するが，面積も増加する．

3.2 設 計 例

マルチバンク構成レジスタファイルの設計を行った結果について示す[10][11]．設計結果より，4 バンク構成・128 エントリのマルチバンク構成レジスタファイルは，0.35μm プロセスにおいて，従来のマルチポートレジスタファイルに比べ 70%の面積削減，48.6%のアクセスタイム削減が実現できることがわかった．

3.3 バンクのアクセス競合による性能低下

マルチバンク構成レジスタファイルは内部をバンク化しているため，同一バンクのレジスタに対して同時に複数のレジスタアクセスができないという制約がある．この制約のため，従来のマルチポートレジスタファイルの代替としてマルチバンク構成レジスタファイルを用いても，バンクに対するアクセス競合が頻繁に発生し性能低下を招く．アクセス競合はバンク数を増加させることである程度回避可能だが，バンク数の増加は面積およびアクセスタイムの増大を招き，バンク構造の利点を活かせなくなる．バンク数を増やさずにアクセス競合を削減するには，レジスタリネーミングを工夫する方法がある．リネーム先レジスタを単一のバンクに偏らないように複数バンクに分散して配置することでアクセス競合を削減できる．

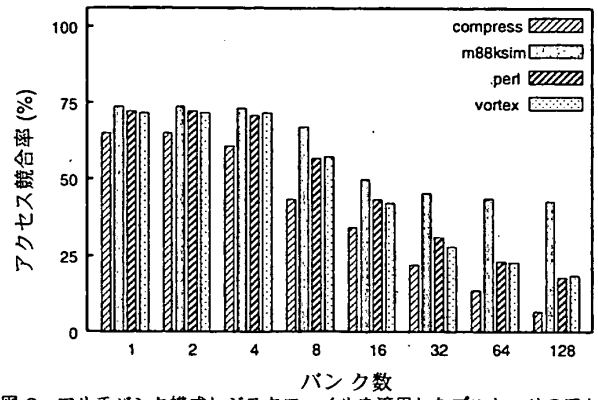


図 2 マルチバンク構成レジスタファイルを適用したプロセッサのアクセス競合率

3.4 予備評価

ここで，アクセス競合が性能に及ぼす影響を調査した．プロセッサの命令発行数を 4，総レジスタ数を 128 とした場合の，総レジスタアクセス数に占めるアクセス競合率，およびマルチポートレジスタファイルに対する実行時間比を調べた．レジスタリネーミングは，バンクを考慮した特別なリネーミングは用いていない．結果を図 2，図 3 に示す．

図 2 は，総レジスタアクセス回数に占めるアクセス競合の発生率である．1〜4 バンク構成では 65.0%〜73.7%のアクセス競合が発生する．また，バンク数を最大の 128 まで増加させたとしてもアクセス競合は完全に削減できないことがわかる．

図 3 に，マルチポートレジスタファイルを用いた場合の実行時間を 1 とした，マルチバンク構成レジスタファイルの実行時間比を示す．バンク数を最大の 128 まで増加させたとしても，マルチポートレジスタファイルを用いた方式に比べ，アクセス競合が原因で 1.1〜2.6 倍ほど実行時間が増加している．以上より，マルチポートレジスタファイルをマルチバンク構成レジスタファイルで置き換えるにはペナルティが大きく，そのままでは難しい．バンク数を増加させる方法以外に，アクセス競合を回避する方法が必要となる．

4. アクセス競合回避手法の提案

マルチバンク構成レジスタファイルのアクセス競合を回避する手法として，ハードウェアによる競合回避手法と，ソフトウェアによる競合回避手法が考えられる．これらの手法について示す．

(a) バンク構造を考慮したレジスタリネーミング
(b) レジスタアクセスのスケジューリング
(c) レジスタアクセス数の削減
(d) コンパイラによる，バンク構造を意識したレジスタ割り付けおよびコードスケジューリング

本稿では，ハードウェアによって行う (a)，(b)，(c) のアクセス競合回避手法に注目し，提案・検討を行う．

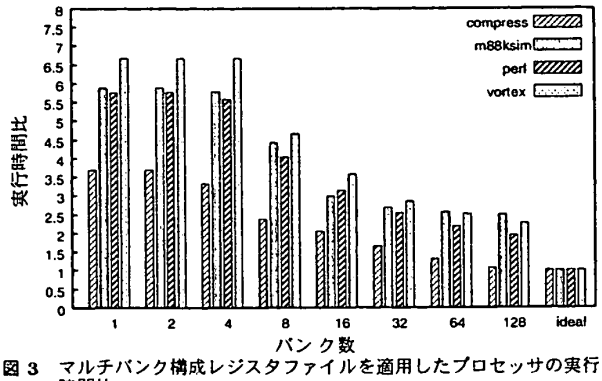


図 3 マルチバンク構成レジスタファイルを適用したプロセッサの実行時間比

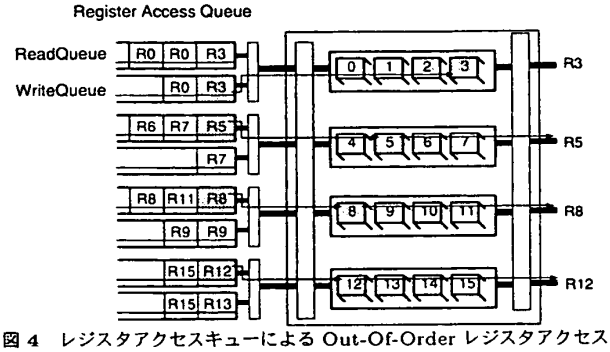


図 4 レジスタアクセスキューによる Out-Of-Order レジスタアクセス

4.1 (a) バンク構造を考慮したレジスタリネーミング

レジスタリネーミングにおけるリネーム先レジスタは，マルチバンク構成レジスタファイルのバンク構造を考慮して選択する．割り付け先レジスタの選択アルゴリズムは，ラウンドロビンを使用する．

4.2 (b) レジスタアクセスのスケジューリング

レジスタファイルに対するアクセスをスケジューリングすることによって，アクセス競合を削減する．これには，次に示す 2 つの方法を提案する．

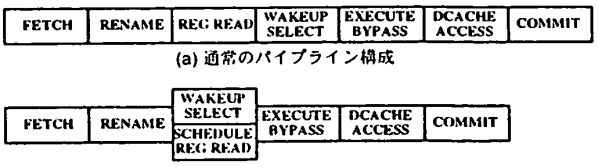
4.2.1 Out-Of-Order レジスタアクセス

スケジューリングを行うために，命令をレジスタアクセス（オペランド）単位に分解し，レジスタアクセスが可能になるまで保持しておくようにする．この情報を保持するバッファとして，レジスタアクセスキューを用意する．図 4 にレジスタアクセスキューの構成とスケジューリングの動作を示す．レジスタアクセスキューは，各バンクごとのリードキュー，ライトキューで構成される．リードキューは，読み出しアクセスを保留する FIFO キューで構成される．ライトキューは，書き込みアクセスおよび書き込みデータを保留する．書き込みデータは演算器およびデータキャッシュより供給される．

次に，動作を説明する．命令はリネームステージでリネーミングされると同時にレジスタアクセス単位まで分解され，各バンクごとのアクセスに振り分けられて各アクセスキューへ登録される．その後，先頭エントリの条件がそろった時点で，レジスタファイルに対して発行される．真の依存関係を保証するために，ライトキューからの発行はリードキューに優先して発行される．出力依存関係，逆依存関係はレジスタリネーミングにより保証される．

4.2.2 スケジューリングに適したパイプライン構成

Out-Of-Order レジスタアクセス手法によって 1 つの命令をレジスタアクセス単位に分解することで，レジスタアクセスのスケジューリングに適したパイプライン構成が可能になる．図 5 に従来のパイプライン構成との比較



(a) 通常のパイプライン構成

(b) 提案するパイプライン構成

図 5 レジスタアクセス・スケジューリングに的したパイプライン構成

を示す．本手法では，スケジューリングおよびレジスタアクセスのステージをリザベーションステーションと同一ステージに位置づけることで，命令発行の自由度を増大できる．

提案するパイプライン構成の動作を説明する．まずリネームステージで，デコードおよびリネーミングされた命令はレジスタアクセス単位に分解される．その後，すぐにリザベーションステーションへ登録され，オペランド待ちの状態となる．その一方で，分解されたレジスタアクセス要求はレジスタアクセスキューへ登録され，バンク構成レジスタファイルに対して Out-Of-Order 発行される．リザベーションステーションへは，レジスタファイルおよび演算結果のフォワーディングによってオペランドが供給され，演算器に対して Out-Of-Order 発行される．このようにしてスケジューリングの自由度を増大させている．

4.3 (c) レジスタアクセス数の削減

レジスタファイルへのアクセス数を削減することで，アクセス競合を回避する．これには次に示す 2 つの方法が利用できる．

4.3.1 フォワーディングによるアクセス回避

真の依存関係にある命令で使用されるソースレジスタは，フォワーディングされた値を使用する．すなわち，このソースレジスタは読み出す必要がなくなる．

4.3.2 同一レジスタアクセスのコンバイン

複数の命令で同一レジスタ番号の読み出しがあった場合，先行命令のレジスタ読み出しで得られた値を渡すことで，後続命令の同一レジスタは読み出す必要がなくなる．実現方法について説明する．まずリネームステージにおいて，新規に登録される命令のソースレジスタが，先行命令で使用するレジスタと同じレジスタかどうかを検出する．同じレジスタであれば，リザベーションステーションに登録するレジスタのタグを，先行命令がアクセスするタグと同一にする．

4.4 提案手法の面積およびアクセスタイム

提案手法による付加回路の増加がマルチバンク構成レジスタファイルによる面積削減効果を上回るならば，提案手法は意味を持たなくなる．今回は提案手法のハードウェアを設計していないが，必要とされるハードウェアについて示す．

(a) バンク構造を考慮したレジスタリネーミング

リネーミング方法の変更のみで対処できるため，特に付加回路は必要としない．

(b) レジスタアクセスのスケジューリング

Out-Of-Order レジスタアクセスに利用するレジスタアクセスキューの構成が，面積およびアクセスタイムを決定する要因となる．面積に関して，FIFO キューや RAM で構成できるレジスタアクセスキューの利用は，ポート数の 2 乗に比例して増大するレジスタファイルに比較して小面積で構成できるため有効である．アクセスタイムに関して，仮にレジスタアクセスキューのエントリ数の増加により遅延が増大した場合，問題の生じる可能性がある．アクセスタイムの増加によりバンク構成レジスタファイルを含めて 1 サイクルでアクセスできない場合，パイプラインが 1 ステージ長くなる．すなわち，バンク構成

レジスタファイルに対するアクセスは，全てレジスタアクセスキューを介した2サイクルで行われるようになる．このとき，リザベーションステーションはフォワーディングされた値を受け取れない可能性がある．その場合，レジスタファイルから最新の値を得ることになるが，もし最新の値がレジスタファイルに書き込まれていなければ，書き込みが完了するまで読み出しを待つ必要がある．加えて，フォワーディングされた値を受け取れるかどうかはリネームステージの時点ではわからないため，レジスタアクセスキューの読み出し要求はリザベーションステーションがオペランドを受け取るか，もしくはレジスタファイルに書き込みが行われ依存関係が解決し，なおかつレジスタファイルに対して読み出し要求が発行できるまで保留される．このためにレジスタアクセスキューはリザベーションステーションより多くのエントリ数を必要とする．しかし不用意なエントリ数の増加はさらなる面積増大やアクセスタイムの増加をもたらす可能性があるため，エントリ数を増やす以外の方法を用いる必要がある．

このステージ数の増加に対しては，4.2.2節で提案した，スケジューリングに適したパイプライン構成を用いることで解決できる．提案したパイプライン構成では，フォワーディングの値を受け取れない場合が存在しないために，上記のような問題は生じない．

(c) レジスタアクセス数の削減

レジスタアクセス数の削減のうち，同一レジスタアクセスのコンバインを追加するための付加回路について検討する．この手法を実現するためには，リネームステージからのアクセス要求と，リードキュー内のエントリのタグとの比較器が必要となる．リードキューのタグビット幅は，総レジスタ数128，4バンク構成であれば5ビットで，エントリ数は後述の評価結果によれば4命令発行の場合，2〜3で済む．従って，チップ面積を圧迫するほどの大きなハードウェアは必要とならないと考えられる．

5. 性能評価

本節では，提案したマルチバンク構成レジスタファイルおよびアクセス競合回避手法を備えた方式と，従来のマルチポートレジスタファイルを備えた方式との比較評価を行った結果について示す．

5.1 評価環境

評価には，C++で作成したトレースドリブンシミュレータを使用した．入力トレースデータは，SimpleScalar Tool Set[13]上でSPECint95の8つのベンチマークプログラムを実行して得られたものを用いた．測定時間が過大にならないようにするため，トレースデータは最初の1000万命令を使用した．なお，SPECint95のバイナリはSimpleScalarのWEBページからダウンロードしたものを使用した．

レジスタファイルの性能比較評価であるため，分岐予測は100%的中するものとし，キャッシュは命令キャッシュ・データキャッシュともヒット率100%の理想的なものを仮定した．

5.2 評価項目

評価を行った項目を次に示す．

(1) アクセス競合回避手法の効果

提案する3種類のアクセス競合回避手法を個別に適用した場合，および全ての手法を適用した場合におけるアクセス競合率と実行時間比を調べた．紙面の都合により，グラフ中にはcompress，m88ksim，perl，vortexの結果のみ示す．レジスタアクセスのスケジューリングによる競合回避を適用した場合は競合率を0%に抑えることができるため，実行時間比のみ示す．

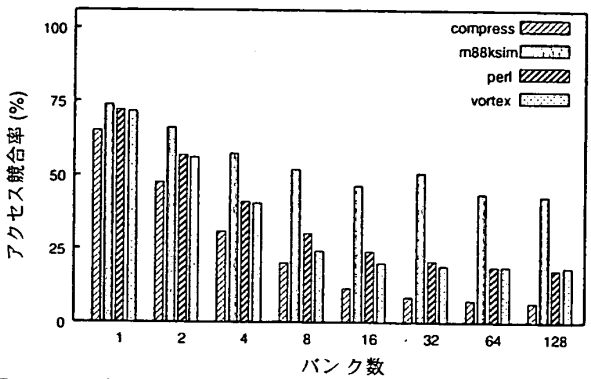


図6 バンク構造を考慮したレジスタリネーミングを適用した場合のアクセス競合率

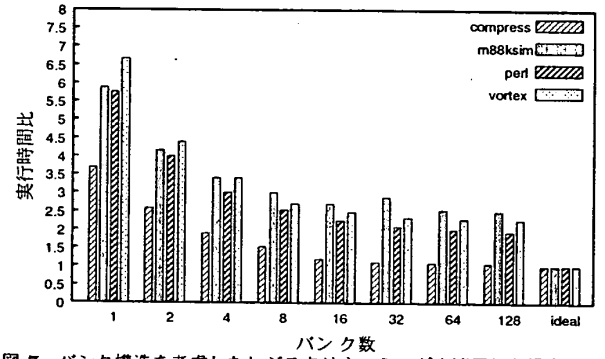


図7 バンク構造を考慮したレジスタリネーミングを適用した場合の実行時間比

(2) レジスタアクセスキューの深さ

最適なレジスタアクセスキューのサイズを調べるため，キューの深さを1から5まで変化させた場合のアクセス競合率と実行時間比を調べた．

5.3 評価結果

5.3.1 (1) 提案したアクセス競合回避手法の評価

バンク構造を考慮したレジスタリネーミング

図6に，4節で提案した3つのアクセス競合回避手法のうち，バンク構造を考慮したリネーミングを適用した場合のアクセス競合率を示す．縦軸は全てのレジスタアクセス要求に対する競合の発生率を表す．横軸はバンク数を表す．図7は，同手法を適用した場合の，マルチポートレジスタファイルを用いた場合と比較した際の実行時間比を示す．項目のidealは従来の理想的なマルチポートレジスタファイルを用いた場合である．

128バンク構成において，compressは6.5%と低い競合率で，実行時間比は1.07倍とマルチポートレジスタファイルに近い性能を示した．一方で，m88ksimは競合率42.8%，実行時間比2.5倍を示した．m88ksimの結果に関する考察は後述する．ベンチマークの特性から，レジスタアクセスの競合率が実行時間に比例するわけではないが，競合率の高さが性能に悪影響を与えていることがわかる．

レジスタアクセスのスケジューリングによる競合回避

図8に，レジスタアクセスのスケジューリングによる競合回避手法を適用した場合の実行時間比を示す．レジスタアクセスキューは十分な深さを持っているものと仮定する．実行時間比はm88ksimを除き，バンク数を増加させることで，ほぼマルチポートレジスタファイルと同等の性能を達成している．しかし，バンク数1〜8の範囲では実行時間は4倍以上のベンチマークが多く，依然として性能は低いままと言える．

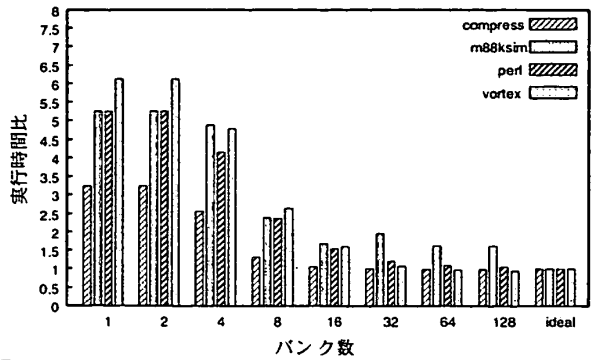


図 8 レジスタアクセスのスケジューリングを行った場合の実行時間比

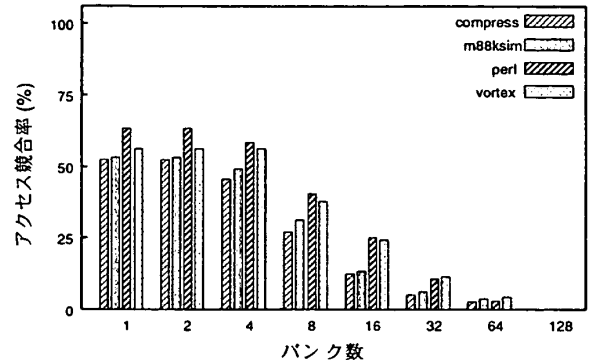


図 9 レジスタアクセス数削減手法を適用した場合のアクセス競合率

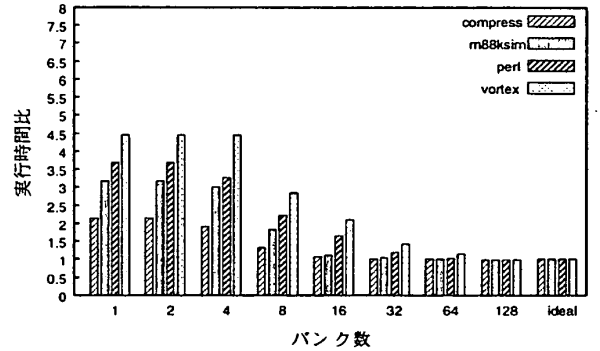


図 10 レジスタアクセス数削減手法を適用した場合の実行時間比

レジスタアクセス数の削減による競合回避

図 9, 図 10 に，レジスタアクセス数削減手法を適用した場合のアクセス競合率，実行時間比を示す．128 バンク構成においては，アクセス競合を完全に削減できており，マルチポートレジスタファイルと同等の性能を実現できている．しかし，1〜8 バンク構成では実行時間比は大きく，マルチポートレジスタファイルと同等の性能とは言えない．

全てのアクセス競合回避手法を適用

図 11 に，アクセス競合回避手法を全て適用した場合の実行時間比を示す．実行時間は，4 バンク構成で 1.03〜1.14，8 バンク構成ではマルチポートレジスタファイルとほぼ同等の実行時間を実現できる．少ないバンク数でマルチポートレジスタファイルと同等の性能を実現するには，全てのアクセス競合回避手法を適用することが必要であると言える．また，ベンチマークプログラムによっては，マルチポートレジスタファイルより実行時間の短いものがある．この理由については後述する．

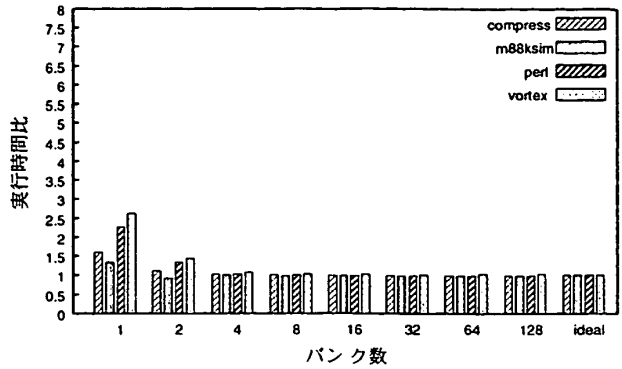


図 11 アクセス競合回避手法を全て適用した場合の実行時間比

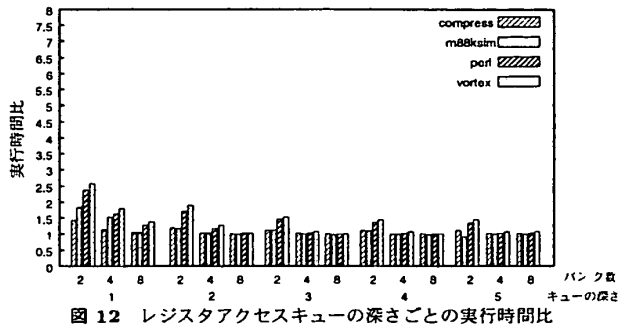


図 12 レジスタアクセスキューの深さごとの実行時間比

```
.L1:
        sw      r7,   0(r8)
        sw      r7, -24(r2)
        sw      r7, -20(r2)
        sw      r7, -16(r2)
        sw      r7, -12(r2)
        sw      r7,  -8(r2)
        sw      r7,  -4(r2)
        sw      r7,   0(r2)
        addiu   r2, r2, 32
        addiu   r8, r8, 32
        addiu   r3, r3, -1
        bne     r3, r0, .L1
```

図 13 m88ksim のコードの一部

5.3.2 (2) アクセスキューの深さの評価

図 12 に，レジスタアクセキューの深さを 1〜5 まで変化させた場合における，実行時間比を示す．アクセス競合回避手法は全て適用している．なお，グラフにはバンク数が 2，4，8 の場合のみを示す．結果より，レジスタファイルが 4 バンク構成の場合の深さは 3，8 バンク構成の場合の深さは 2 で，マルチポートレジスタファイルとほぼ同等の性能を実現できる．

5.4 考　　察

m88ksim の競合率削減

図 6 を見ると，バンク構造を考慮したリネーミングによるアクセス競合回避手法を適用しても，m88ksim ではバンク数 128 構成でアクセス競合率は 76%と高い割合を示している．原因は，m88ksim は同一のレジスタに対し連続的に読み出しを行う命令を含むループが多く繰り返されるからである．図 13 に，そのループ部分のコードを示す．このループは，250〜950 万命令まで繰り返される．このような場合，アクセス競合回避手法の “同一レジスタアクセスのコンバイン” を用いることで，競合を回避できる．図 9 では，この同一レジスタに対するアクセス競合を削減した効果が確認できる．

バンク構成レジスタファイル方式の性能が高い理由

図 11 で，アクセス競合回避手法を全て適用した場合に，バンク構成レジスタファイルの実行時間がマルチポート

レジスタファイルより短いことがある．例えば 128 バンク構成では，vortex を除く全てのベンチマークで 0.3%～4.7%実行時間が短縮されている．この原因は，レジスタアクセス数の削減手法の一つである “同一レジスタアクセスのコンバイン” によるものである．この手法の効果として，複数命令で同一のソースレジスタを利用する場合に，先行命令が後続命令のソースレジスタを同時に得ることができ，後続命令のレジスタアクセスを待つことなく命令発行が可能になることがある．この手法に加えて，レジスタアクセスのスケジューリングに向いたパイプライン構成の効果によって，レジスタアクセスすることなく命令が発行可能になる場合が存在する．このため，アプリケーションによってはバンク構成レジスタファイルのほうが性能が高くなる．

6. 関連研究

レジスタファイルの面積削減手法やアクセスタイムの削減手法についての関連研究を説明し，我々の方式と比較する．

Alpha プロセッサ[2] では，レジスタファイルおよび機能ユニットをあわせた実行ユニットをクラスタ化することで各レジスタファイルのポート数を減少させ，面積およびアクセスタイムの問題を緩和している．しかしこの手法では，クラスタ間の同期を保つための通信によるオーバヘッドが存在する．また，ポート数が増大すればクラスタ数をさらに増やすことが必要になり，面積増大が著しい．我々の方法は，よりスケーラブルな構成であると考えられる．

文献 12) では，2 レベルレジスタファイル構成，バンク構成レジスタファイルの利用を提案している．バンク構成のレジスタファイルを用いて，通常のレジスタファイルとほぼ同等の IPC を保ったままアクセス時間を 50%減少させ，エネルギー消費量を 18 分の 1 に減少できるとしたシミュレーション結果を得ている．バンク構成レジスタファイルに対するアクセスの方法は，バンク構造を考慮したリネーミング手法と，競合が発生した場合には機能ユニットの前段のバッファに値を保持しておき，複数サイクルで読み出す方法を用いている．

我々の方法との違いは，バンクの構成とアクセス競合の回避手法である．彼らはバンク構成レジスタファイルを設計してはいないが，バンクごとにリード/ライト独立のポートを備えたものを想定している．一方，我々のバンクはリード/ライト共用のポートを 1 本備えたものを想定しており，制約は厳しいが，より面積効率のよいレジスタファイルである．また，我々の想定したレジスタファイルはバンクの制約が厳しいが，レジスタアクセスのスケジューリングを導入することにより積極的な競合回避手法を実現している．一方で，競合回避手法によりハードウェア資源が増大する可能性がある．

7. おわりに

本稿では，マルチポートレジスタファイルをマルチバンク構成レジスタファイルで置き換える方法について検討した．マルチバンク構成レジスタファイルは，従来のマルチポートレジスタファイルに比べて小面積化・高速化を実現できる．設計結果より，4 バンク，128 エントリのレジスタファイルでは 70%の面積削減，48.6%のアクセスタイム削減を実現した．しかし，バンクに対するアクセス競合のため，性能低下を招く．このペナルティに対処する手法として，次のようなアクセス競合回避手法を提案した．

- バンク構造を考慮したレジスタリネーミング
- レジスタアクセスのスケジューリング
- レジスタアクセス数の削減

性能評価の結果，提案手法を用いたマルチバンク構成レジスタファイルは，マルチポートレジスタファイルとほぼ同等の性能を実現できることがわかった．また，同等の性能を実現するための付加回路として，レジスタアクセスキューの増設が必要となる．必要なキューの深さは，4 命令発行プロセッサにおいて，4 バンク構成では 3，8 バンク構成では 2 である．提案手法による付加回路の増加は，マルチバンク構成レジスタファイルの利用で削減できた面積より十分小さくなると考えられる．今後の予定として，提案したアクセス競合回避手法の設計を行い，面積およびアクセスタイムを評価する．

謝辞 本研究の一部は半導体理工学研究センターとの共同研究 “大きなランダムアクセスバンド幅を持つスーパコンパクト・マルチポートメモリ，及びそれを用いたシステム・オン・チップ／パッケージ向け高性能アプリケーション” による．

参考文献

1) S. S. Mukherjee, et al., “Detailed Design and Evaluation of Redundant Multithreading Alternatives”, ISCA-29, 2002.
2) R. Kessler, “The Alpha 21264 Microprocessor”, IEEE Micro, 19(2):24-36, March/April 1999.
3) H. J. Mattausch, “Hierarchical N-port memory architecture based on 1-port memory cells”, Proceedings of the 32nd European Solid-State Circuits Conference (ESSCIRS 97), pp.348-351, 1997.
4) H. J. Mattausch, et al., “Area-efficient multiport SRAMs for on-chip data-storage with high random-access bandwidthand large storage capacity”, IEICE Trans. Electron., Vol.E84-C, No.3, p410, 2001.
5) D. Tullsen, et al., “Simultaneous Multithreading: Maximizing On-Chip Parallelism” In Proceedings of ISCA-22, pp.392-403, 1995.
6) D. Tullsen, et al., “Exploiting Choice: Instruction Fetch and Issue on an Implementable Simultaneous Multithreading Processor,” In Proceedings of MICRO-32, pp.186-192, Nov. 1999.
7) D. Brooks, et al., “A Framework for Architectural-Level Power Analysis and Optimizations”, In Proceedings of ISCA-27, June 2000.
8) R. Preston, et al., “Design of an 8-Wide Superscalar RISC Microprocessor with Simultaneous Multithreading”, ISSCC-2002, pp.334-335, 2002.
9) V.Zyuban, et al., “The Energy Complexity of Register Files”, In Proceedings of the International Symposium on Low-Power Electronics and Design, Aug. 1998.
10) 内田裕志ほか，“バンク構造を用いた高並列プロセッサ向き小面積多ポートレジスタファイル”，信学技報 CAS-2002-47, pp.31-36, 2002.
11) 内田裕志ほか，“マルチバンク構造による小面積多ポートレジスタファイル”，信学技報 ICD-2002, 2002.
12) R. Balasubramonian, et al., “Reducing the Complexity of the Register File in Dynamic Superscalar Processors”, MICRO-34, 2001.
13) D. Burger, et al., “The simplescalar toolset, version 2.0”, Technical Report TR-97-1342, University of Wisconsin-Madison, 1997.