Vol.2014-ARC-212 No.18
2014/10/7

レジスタ・キャッシュ・システムにおけるレジスタ・ファイルのマルチバンク化

山田 淳二[1,a)]　倉田 成己[1]　塩谷 亮太[2]　五島 正裕[1]　坂井 修一[1]

概要：巨大なレジスタ・ファイルの消費電力と熱の問題を解決するため，レジスタ・キャッシュが提案されている．レジスタ・キャッシュは，メイン・レジスタ・ファイル (MRF) からのリードを削減し，面積と消費電力を数分の 1 程度までに削減することができる．
本稿では，MRF にバンク分割を用いることで，MRF のポート数を削減する手法を提案する．バンク分割ではバンク・コンフリクトが課題となるが，(1) 複数演算器からの MRF の同一エントリに対するリードを，スイッチにより分配する (2) MRF からのリードに 2cycle に分割する (3) ライト・バッファによって MRF へのライトを平滑化するなどの手法を使うことで，バンク・コンフリクトの影響を軽減することが出来る．
評価の結果，相対 IPC≥99%の範囲で面積性能比が高いのは，2(R/W)×6 バンク,RC エントリゼロの条件であった．この時，通常の十分なポート数を持つレジスタ・ファイルの面積を 100%とした相対値で，相対面積は 19.8%, 相対 IPC は 99.0%であった．従来のバンク分割を行わない MRF では，相対 IPC≥99%となるのは，5R+5W/RC0 条件で 51.3%，4R+4W/RC16 条件で 52.1%であったので，より高い面積性能比を実現した．
現在の構成では，ライト・バッファには十分なリード・ポート数が割り当てられ，ライト・バッファから演算器へのフォワーディングも行われているため RC の効果は限定的である．しかし，このために面積の約 2/3 をライト・バッファが占める状態となっており，今後はライト・バッファを簡素化し，小容量の RC を用いるなど，よりバランスの取れた構成を検討する必要がある．

1. はじめに

物理レジスタ・ファイルは，最近のスーパスカラ・プロセッサの構成要素の中でも最も高コストなものの 1 つとなっている．

巨大なレジスタ・ファイル

この要因として，まず，レジスタ・ファイルの容量の増加がある．通常，物理レジスタ・ファイルには in-flight な命令数に応じた容量が必要となる．より多くの命令レベル並列性を抽出するため，近年ではスーパスカラ・プロセッサの in-flight 命令数を増やす傾向にあり，その一環として物理レジスタ・ファイルの容量も増大している．また，マルチスレッドをサポートするコアでは，スレッド数に比例した容量が必要となるため，更に数倍という規模での容量増を引き起こすことになる．

次に物理レジスタ・ファイルのポート数の増加がある．4 命令同時実行可能なスーパスカラ・プロセッサでは，レジスタ・ファイルに 8 つのリード・ポートと 4 つのライト・ポートが必要となる．レジスタ・ファイルは通常，多ポートの RAM によって構成されるが，RAM の回路面積はポート数の 2 乗に比例するため，その回路面積は容量の割に非常に大きなものとなる．

これら 2 つの理由により，レジスタ・ファイルは近年では L1 データ・キャッシュに匹敵するほど巨大な回路となっている．巨大なレジスタ・ファイルは，IPC の低下や回路の複雑さの増大に加えて，消費電力，熱の増大などの様々な問題を引きおこす．

消費電力と熱の増大

RAM の消費電力は，その回路面積に加えて，アクセス頻度にも比例する．ロード/ストア命令が L1 データ・キャッシュに対してそれぞれ 1 回しかアクセスを行わないのに対し，ほぼ全ての命令は通常，レジスタ・ファイルに対して 2〜3 回のアクセスを行う．このため，面積が同程度の L1 データ・キャッシュと比較して，レジスタ・ファイルはよ

[1] 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo
[2] 名古屋大学大学院工学系研究科
Graduate School of Engineering, Nagoya University
[a)] yamadaju@mtl.t.u-tokyo.ac.jp

り大きな電力を消費する．

消費電力とそれによって発生する熱は，最近のプロセッサ・コアにおける問題の中でも，もっとも深刻なものの一つである．レジスタ・ファイルを含む領域は，プロセッサ・コア内のホット・スポットであり，その動作周波数を制限する主な要因の一つになっている．

レジスタ・キャッシュ

レジスタ・キャッシュを導入すれば，巨大なレジスタ・ファイルに起因するこれらの様々な問題を解決することが可能である．

レジスタ・キャッシュのシステムは，主に，多ポート少エントリの**レジスタ・キャッシュ**（Register Cache，**RC**）と少ポート多エントリの**メイン・レジスタ・ファイル**（Main Register File，**MRF**）の組合せによって構成される．RC は少エントリゆえ，MRF は少ポートゆえに，元のレジスタ・ファイルより格段に小面積・低電力となる．

特に, 塩谷らが提案した**NORCS** (Non-latency-Oriented Register Cache System) では，IPC の低下を 2%程度に抑えながら，面積を 1/4 程度に，消費電力を 1/3 程度に削減することに成功している [1].

レジスタ・キャッシュ・システムにおけるメイン・レジスタ・ファイル

4 章で詳しく述べるが，NORCS では、メイン・レジスタ・ファイルのポート数は 2-read 2-write 程度必要であり、依然としてレジスタ・キャッシュ・システム全体の面積の半分程度は、メイン・レジスタ・ファイルが占めている。このことは特に，マルチスレッド・プロセッサで重要な問題となる．NORCS では MRF を 128 レジスタとしているが、マルチスレッド・プロセッサでは MRF はスレッド数分だけエントリ数を増やす必要がある．

マルチスレッド・プロセッサでは，スレッド数の増加に対して，RC の容量はほとんど増やす必要がないことが分かっている [2], [3] ことから、マルチスレッド・プロセッサ化によって、レジスタ・キャッシュ・システム全体に占めるメイン・レジスタ・ファイルの面積は増加することとなる。

マルチ・バンク化

筆者らは、この対策として、MRF のマルチ・バンク化を提案する。

詳細は 4 章で述べるが、マルチ・バンク化を行うと、バンク数分だけスループットを改善することができる．それに対して、マルチ・バンク化による面積増大は、SRAM アレイの分断に伴うオーバーヘッドであり、ポート数増加に見られるような、2 乗に比例した増加は発生しない。したがって、例えばリード/ライト共通 1 ポートの最小サイズの SRAM を用い、マルチ・バンク化によって、最大スループットを確保すれば、革新的に小さなメイン・レジスタ・ファイルを実現可能である。

マルチ・バンク化の課題は、当然、**バンク・コンフリクト**である。しかし、レジスタ・キャッシュ・システムにおいては、以下の理由によって問題になりにくいと考えられる。

メイン・レジスタ・ファイルへのライト ライト・バッファで平滑化されるためバンク・コンフリクトが直ちにストールを招かない

メイン・レジスタ・ファイルからのリード レジスタ・キャッシュが肩代わりするためスループットに余裕がある

これらは、レジスタ・キャッシュ・システムが持っているもともとの性質であり、レジスタ・キャッシュ・システムに追加要素を加える必要はない。

さらに，NORCS の「RC ミスを仮定したパイプライン」の考え方を拡張し、MRF からのリードを行うステージを 2cycle 設ければバンク・コンフリクトを更に軽減することも可能である．

また，NORCS では，MRF のリード・ポートが演算器が要求するソース・オペランドの数に対して不足することから，これを接続するセレクタが必須である．このセレクタは，MRF の同一エントリのリードを演算器に振り分ける目的も利用可能である．

MRF の同一エントリは必ず同一バンクであるので，同時に実行される複数の命令が，MRF の同一エントリを参照する場合バンク・コンフリクトが発生しやすくなるが，セレクタの制御で回避が可能である．似たような考え方は，SPARC T4 プロセッサ [4] のレジスタ・ファイルにも見られる．

以下，2 章では，NORCS を中心として従来のレジスタ・キャッシュ・システムについて，3 章では，本稿と考えたが異なる他のマルチ・バンク化の手法について述べ，4 章で，提案するマルチ・バンク化の手法について述べる。5 章で，面積, 消費電力及び性能の評価結果についてまとめる．

2. 通常のレジスタ・キャッシュ・システム

2.1 レジスタ・キャッシュ・システム

本節では，提案のベースとなる**レジスタ・キャッシュ・システム**（Register Cache System，**RCS**）について説明する．

2.2 レジスタ・キャッシュ・システムの概要

前章で述べたように，RCS は，多ポート少エントリの**レジスタ・キャッシュ**（Register Cache，**RC**）と少ポート多エントリの **MRF**（Main Register File，**MRF**）の組合せによって構成される．RC は少エントリゆえ，MRF は少ポートゆえに，元のレジスタ・ファイルより格段に小面積・低電力となる．

特に, 塩谷らが提案した**NORCS** (Non-latency-Oriented Register Cache System)[1] では，RC ヒット時にもレイテ

ンシを削減しない「RC ミスを仮定したパイプライン」の採用により，IPC の低下を 2%程度に抑えることに成功している．また，面積を 1/4 程度に，消費電力を 1/3 程度に削減することに成功している．

2.3 レジスタ・キャッシュ・システムへのライト

RC に対するライトは通常，ライト・スルー方式で処理される．すなわち，レジスタへの書き込みは，RC と同時に MRF に対しても行われる．これは以下の理由による．

通常のメモリ階層における，ライト・スルーに対するライト・バックのメリットは，2 回目以降のライトをもキャッシュ上のエントリに対して実行することによって，次の階層へのライト（バック）の回数を 1 回で済ますことにある．ライト・スルーでは，1 回のライトごとに，次の階層への 1 回のライト（スルー）が発生することになる．

しかしレジスタ・キャッシュにおいては，ライト・バックのこのようなメリットは生じない．これは，通常のメモリのロケーションとは異なり，レジスタに対してはリネーミングが施されるためである．前節で述べたように，各命令のデスティネーションには別の物理レジスタが割り当てられる．したがって，RC 上のあるエントリに対するライトは，割り当てられた命令が実行された時の 1 回のみである．「2 回目のライト」はそもそも存在しない．

RC をライト・バックとすると，ライト・バック時に RC を読み出すためのポートが余計に必要になり，かえって不利である．

以上の理由により，RC はライト・スルーとするのである．

2.4 通常のレジスタ・ファイルとレジスタ・キャッシュ・システム

本節では，通常のレジスタ・ファイルとレジスタ・キャッシュ・システムについてブロック図を示し，対比して説明する．

通常のレジスタ・ファイル

通常のレジスタ・ファイルの構成を**図 1** に示す．各演算器は，デスティネーション・オペランド ×1，ソース・オペランド ×2 を取り，演算器は 4 ユニットある．従って，レジスタ・ファイルは全体として 8R+4W が必要であり，2R+1W 毎に各演算器に直結されている．

命令はイミディエイトをソース・オペランドに取ることもあるため，レジスタ・ファイルの他に，イミディエイトを演算器に供給するためにスイッチが設けられている．

この通常のレジスタ・ファイルでは，レジスタ・ファイルのポート数不足によるストールは発生しない代わりに，メイン・レジスタ・ファイル全体を多ポートとする必要がある．

NORCS

次に NORCS の構成を**図 2** に示す．

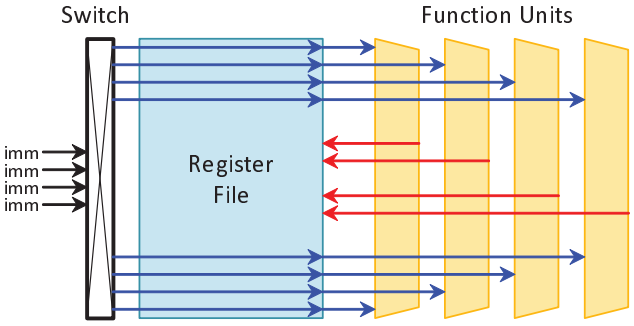

図 1 Block diagram of nomal register file

NORCS は，ポート数の少なくエントリ数の多い MRF と，ポート数が多くエントリ数の少ないレジスタ・キャッシュの組み合わせで実現されている．

レジスタ・キャッシュは 8R+4W であり，通常のレジスタ・ファイル構成の場合のレジスタ・ファイルと同様に，2R+1W 毎に各演算器に直結されている．この他にライト・バッファと MRF があり，図 2 では，ライト・バッファは 2R+4W，MRF は 2R+2W である．

MRF のポート数は，(1) レジスタ・キャッシュがソース・オペランドを供給できず，更に，MRF のリード・ポート数が不足した場合 (2) ライト・バッファから MRF へのライトが間に合わずにデスティネーション・オペランドでライト・バッファがあふれた場合にストールが発生することから，性能と面積のトレードオフで決定する必要がある．塩谷らの NORCS では，4-way のプロセッサに対して 2R+2W を，MRF に必要なポート数としていた．これについては，後に詳細に述べる．

ライト・バッファのポート数は他の要素から従属的に決定され，リード・ポート数は MRF のライト・ポート数，ライト・ポート数は演算器の数で決定されている．

ライト・バッファから演算器へのフォワーディング (ライト・バッファ・フォワーディング) を行う場合，ライト・バッファの役割は，レジスタ・キャッシュの役割と似ている．ライト・バッファとレジスタ・キャッシュの違いは，レジスタ・キャッシュが例えば LRU などのアルゴリズムでエントリの置換を行うのに対して，ライト・バッファは FIFO として動作し，かつ，積極的にエントリを MRF に書き出す制御を行う点である．

MRF のポート数

RC は MRF からのリードの大半を代替するが，依然として，メイン・レジスタ・ファイルのリード・ポートは一定程度必要である．また，RC はライト・スルーであるから，ライト・ポートを大きく減らすことは難しい．この点について検討する．

MRF のリード・ポート数

NORCS では，MRF からのリードが行われるのはキャッ

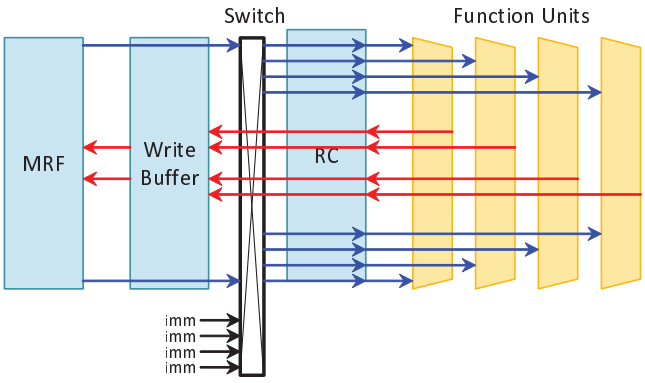

図 2 Block diagram of NORCS

シュ・ミスが発生した時のみである．キャッシュ・ミス時に，リード・ポートが不足するとストールが発生する．

従って，リード・ポート数は，キャッシュ・ミスによって，MRF に対して発生するリード数の分布を考慮して，ストールによる性能低下が十分小さくなるように決定する必要がある．塩谷らのシミュレーションでは，2-read が MRF に必要なリード・ポート数であった．

MRF のライト・ポート数

レジスタ・キャッシュ・システムはライトスルーキャッシュである．物理レジスタは，リネーミングによって命令のデスティネーション毎に新たに割り当てられ，上書きは発生しない．このため，レジスタ・キャッシュをライトバックキャッシュとする意味は無い．

従って，レジスタ・キャッシュ・システムには，ライトの総数を減らす効果は無い．NORCS では，ライト・バッファによる平滑化を行い，演算器 4 ユニットに対して，ライト・ポートを 2-write としていた．

3. 従来のマルチ・バンク化

本章では、マルチ・バンク化の先行研究について述べ、本研究との違いをまとめる。

[5] は、ライト・ポートのみをマルチ・バンク化した例である。リネーミング時に直ちに物理レジスタを割り当てず、依存性・タグを割り当て、ライト・バッファの瞬間にバンク・コンフリクトが発生しないように物理レジスタの割り当てを行う。この方式では、リード・ポートのマルチ・バンク化は当然不可能であり、また、依存性・タグと物理レジスタの対応をとるテーブルが必要となる。[5] では、リード・ポート削減手段としてオペランド・バイパスが行われた場合に MRF へのリードを行わないことで、リード・ポート数の削減を行っている。この手法は，NORCS 及び本研究でも踏襲されており，オペランド・バイパス及びレジスタ・キャッシュのいずれもがソース・オペランドを供給できない場合のみ，MRF からのリードが行われる．

他の研究としては、物理レジスタへのアクセスにアクセス・キューを用いる方式 [6] が提案されている。この方式では、リード・ライトともにマルチ・バンク化が可能であるが、アクセス・キューが追加要素として必要である。

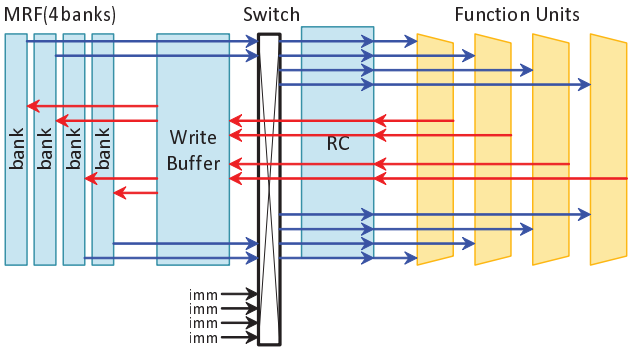

図 3 Block diagram of Multibanked MRF

筆者らの提案するマルチ・バンク化は、

- リード・ポート及びライト・ポートのいずれも、少ポートでマルチ・バンク化を行う。したがって、面積は最小限となる。
- バンク・コンフリクトを緩和するためにレジスタ・キャッシュ・システムの既存の仕組みのみを利用する。したがって、マルチ・バンク化に伴う追加要素は発生しない。

の 2 点で、先行研究とは大きく異なる。

4. 提案手法

4.1 マルチ・バンク化

本章では、マルチ・バンクについて述べる．

マルチ・バンク化 MRF の構成

図 3 に，4 バンク化された MRF の構成を示す．各バンクは 1R+1W であり，全バンクの合計で (1R+1W)×4 のリード・ポート及びライト・ポートが，演算器側のスイッチ及びライト・バッファに接続されている．仮にバンク・コンフリクトが発生しなければ，これは，4R+4W の MRF を使うのと同等である．

4.2 バンク・コンフリクトの軽減

実際には，何らかの形でバンク・コンフリクトが発生するため，その回避が課題である．

バンク・コンフリクトを減らす工夫として，[5] では、ライトのみバンク・コンフリクトを完全に回避していた．これは，リネーミング時に直ちに物理レジスタを割り当てず、依存性・タグを割り当てて起き、ライト・バッファの瞬間にバンク・コンフリクトが発生しないように物理レジスタの割り当てを行う手法による．

しかし，バンクをまたぐ依存関係があるためデスティネーション・オペランド及びソース・オペランドの双方で

完全にバンク・コンフリクトを防ぐのは困難である．

本研究では，バンク・コンフリクトの完全な回避は目指さず，ライト・バッファによりライトのバンク・コンフリクトを軽減し，レジスタ・キャッシュ及びパイプラインの工夫でリードのバンク・コンフリクトを軽減することを目指す．

ライト・バッファ

MRF へのライトでバンク・コンフリクトが発生した場合でも，空き容量がある限りライト・バッファはストールを発生させない．これは，従来の NORCS で，一時的なライト・ポート数不足を回避するためにライト・バッファが用いられ，空き容量がある限りライト・バッファがストールを発生させなかったのと同様である．

レジスタ・キャッシュ

レジスタ・キャッシュは，MRF へのリード要求を減らすことで，バンク・コンフリクトを減少させることができると考えられる．

同一エントリリードの回避

同時に実行される命令が，同一エントリの物理レジスタをソースに取っている場合，同一エントリは必ず同一バンクであるから，バンク・コンフリクトを発生させる．

この問題は，図 3 に示したスイッチの制御で回避可能である．

似たような考え方は，SPARC T4 プロセッサ [4] でも見られる．SPARC T4 は，マルチスレッド・プロセッサであり，1 コアあたり 8 スレッド 1280 レジスタの巨大なレジスタ・ファイルを持つが，レジスタ・ファイルのポートに 5R+3W が必要であるところ，リード・ポート側を 3 として，これをスイッチで 5 に拡張することで，面積を縮小している．

パイプラインの工夫 (2cycle-read)

NORCS では，レジスタ・キャッシュのキャッシュ・ミスを仮定することで，キャッシュ・ミス時に直ちにストールが発生しないパイプラインを構成していた．この考え方を拡張し，パイプラインに 2cycle の MRF リードステージを設けることで，バンク・コンフリクトを緩和することができる．

図 4 に，リードステージが，1cycle 及び 2cycle の場合のパイプラインを示す．SC, IS, RS, CR, RR, EX, CW, BW はそれぞれ，Instruction Scheduling, Register Scheduling, Cache Read, Register Read, Execution, Buffer Write を示す．このうち，RS は，NORCS の場合と同様レジスタ・キャッシュの Hit/Miss 判定を示す．また，BW は，ライト・バッファへのライトである．ライト・バッファから MRF へのライトは遅れて行われる．

図 4 上段は 1cycle の場合，下段は 2cycle の場合であるが，いずれも，I_2 で，ソース・オペランド 2 つがキャッシュ・ミスを起こし，かつ，バンク・コンフリクトによって

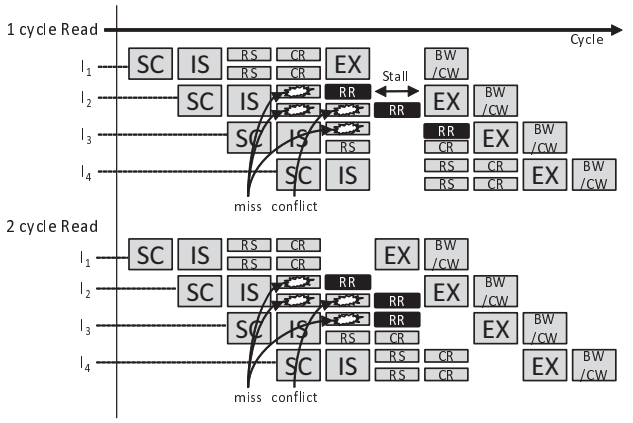

図 4 2cycle-read

RR ステージにソース・オペランドの片方が読み出せない．

この場合，上段の 1cycle ではストールとなるが，下段の 2cycle では，パイプラインに 2cycle 目の MRF リードが組み込まれていることから，ストールは発生しない．

2cycle リードは MRF の読み出しスループットを改善するわけではない．図 4 で，I_2 の 2cycle 目の MRF リードは，I_3 の 1cycle 目 MRF リードでもあり，単位時間当たりに可能なリードの数は変わらない．しかし，MRF の読み出しステージを I_2 と I_3 で共有し，バンク・コンフリクトが起きないようにスケジュールすることで，バンク・コンフリクトの確率を下げることができる．

4.3 面積への影響

本章では，マルチ・バンク化の面積に対する影響を，プロセッサのレジスタ・ファイルのレイアウトを前提として検討する．

マルチ・バンク化の面積に対する影響について検討するために、通常のプロセッサ及び、マルチ・バンク化されていないレジスタ・キャッシュ・システムのレイアウトを理解する必要がある。まず、これらについて、それぞれ説明し、その後に、マルチ・バンク化の影響を考察する。

通常のレジスタ・ファイル

通常のプロセッサでは，レジスタ・ファイルには**ビット・スライス**構成が用いらている。

図 5 に，レジスタ・ファイルと演算器のレイアウトの一例を示す．演算器 ×4 に対して，全体として 8R+4W のレジスタ・ファイルが 2R+1W 毎に各演算器に直結されている．各演算器は 64bit のソースおよびデスティネーションを取り，レジスタ・ファイルは 128 エントリある．

この 8R+4W の合計 12 ポートの単位で 1bit が構成されており，64bit あるので 12 ポートの単位が 64 回繰り返されている．この構成を、ビット・スライスと呼んでおり、従来のプロセッサのレジスタ・ファイルの基本的なレイアウトである。

1bit の高さは，最低でも 12 ポートのデータ線に対応す

る高さが必要であり，図 5 では，ライト・ポートに相補信号を用いているため，合計 16 本が 1bit の高さとなる．

従来のレジスタ・キャッシュ・システム

次に従来のレジスタ・キャッシュ・システムのレイアウトの一例を**図 6** に示す.

左側から，メイン・レジスタ・ファイル, ライト・バッファ, レジスタ・キャッシュの順に並んでおり、4.3 で示した通常のプロセッサと同様に、1bit 毎のビット・スライス構成となっている。

右端の **RC** のポート数は，4.3 のプロセッサのレジスタ・ファイル同様に 8R+4W であるため，1bit の高さも同じになる．このことは，レジスタ・キャッシュ・システムは，大幅なレイアウトの変更を行わずに実現可能であることを示している．

左端の **MRF** は，2R+2W であり，1bit の高さが RC より小さくなる．具体的には，RC が 16 本分の高さであったのに対して，MRF は 6 本分である．これを利用して面積を削減するため，MRF のメモリ・セルは，2 段に積み重ねて配置されている．

積み重ねたレイアウトでは，ライト動作に工夫が必要となる．MRF のワード線は，2bit 分を貫いてレイアウトされており，2bit で共有され、アクセスする 1bit を選ぶために**カラム・セレクタ**が必要である。

リードについては，単に必要な 1bit のみをリードすればよいが、ライトでは、非選択側にライトしてはいけない。このため，MRF に使用するメモリ・セルは，1 ライト・ポートあたり 2 本の相補データ線を用いる形式のメモリ・セルが望ましい．相補データ線であれば，ライトを Disable にする動作を実現できるためである．

中央の**ライト・バッファ**のリード・ポート数はメイン・レジスタ・ファイルのライト・ポート数，ライト・バッファのライト・ポート数は，演算器の数で決定されている．

これらに加え，MRF 又はライト・バッファから演算器へのリード・ポート数は，演算器のリード数に対して不足しているため，**スイッチ**が必要である．図 6 では，MRF が 2R であるので 2 to 8 のスイッチを使用している．

階層データ線

マルチ・バンク化は、例えば 128 エントリといった数のレジスタ・ファイルをバンクに分割する構造である。分割に伴い、当然、面積のオーバーヘッドが発生する。従来のレジスタ・ファイルでも，階層データ線構造のために分割が行われており、この構造との対比が理解の助けとなる。この点について説明する。

従来のプロセッサの MRF では，メモリ・セルのリークに対処するため，階層データ線構成が用いられていた [7]。この階層データ線のレイアウトを**図 7** の上段に示す。ビット・スライス構成を前提としているため、1bit のみを示している。識別のため、上層配線を太い線、下層配線を細い線で示すが、ここでは、ピッチは同じであるとしている。これ以外に、ワード線の配線があるため、最下層から少なくとも 3 層が同一ピッチであることが前提である。

下層配線が Local データ線であり、[7] では、Local データ線に接続できるメモリ・セルは 16 程度であるとしていた。

マルチ・バンク化

マルチ・バンク化をした場合の MRF のレイアウトを図 7 の下段に示す。

上段の階層データ線及び下段のマルチ・バンクのいずれも、レジスタ・ファイルを 2 分割しており、それに伴うオーバーヘッドが存在している。

両社の、最も大きな違いは、分割されたアレイが同時にいくつ動くかである。階層データ線では、分割されたアレイの 1 つのみが同時に動けばよいのに対して、マルチ・バンクでは、分割されたアレイがバンクであり、全バンク同時の動作が求められる。

このため、図 7 では、Global データ線, カラム・セレクタ及びライト・アンプの扱いが異なる。階層データ線では、分割された全アレイが共有しているのに対して、マルチ・バンクでは各アレイ毎に存在する。

したがって、マルチ・バンク化は階層データ線より大きな面積オーバーヘッドを発生させる。しかし、Global データ線が各アレイ (バンク) 毎に存在することによって、バンク・コンフリクトが発生しない場合には、2R+2W のポートを 2 組使うことができる。

マルチ・バンク化に伴う面積オーバーヘッドについて、厳密には、後の評価で明らかにする。

4.4 メモリ・セル

図 8 に，ポート構成の異なる各種メモリ・セルの回路図とそれぞれの 1 ビットあたり面積を示す．1 ビットあたり面積は，$(\text{単位配線ピッチ}/2)^2$] を単位としている．

塩谷らが提案した NORCS[1] では，メイン・レジスタ・ファイルのポート数について 2R+2W を IPC の大きな低下を招かないポート数としていた．この 2R+2W のメモリ・セルの面積は 128 である．

本稿では，マルチ・バンク化によって，これよりも少ないポート数のメモリ・セルを使って MRF を構成することを目指している．このためのもっとも単純な解は，ポート数を半減した 1R+1W のメモリ・セルである．1R+1W のメモリ・セルは 32 の面積を持ち，2R+2W の 1/4 となる．

この他に，リード・ポートとライト・ポートのデータ線が共有されるメモリ・セルとして, 1(R/W), 2R/1W, 2(R/W) などが考えられる．1(R/W) は，SRAM として考えられる最小構成であり面積は 16 である．2R/1W は，最大 2-read まで可能なシングルエンドリード・ポートと 1-write のライト・ポートが共有となる構成であり面積は 48，2(R/W) は，1(R/W) を 2-port 化した構成であり面積は 64 となる．

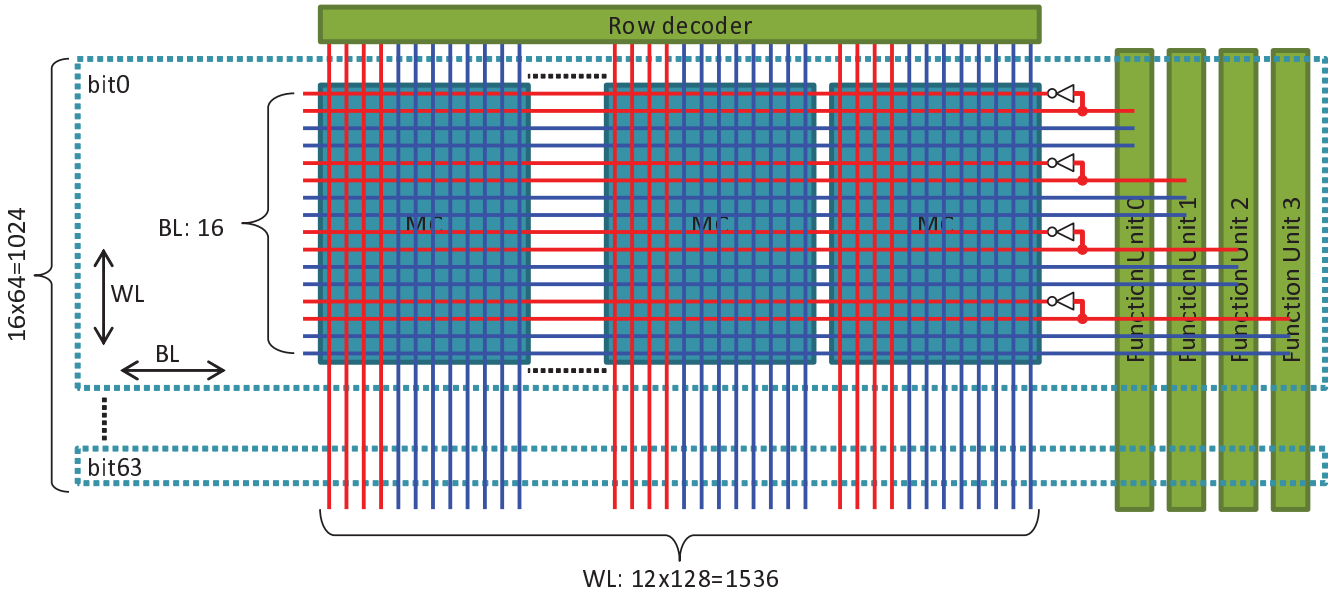

図 5 レジスタ・ファイルの Bit-slice レイアウト

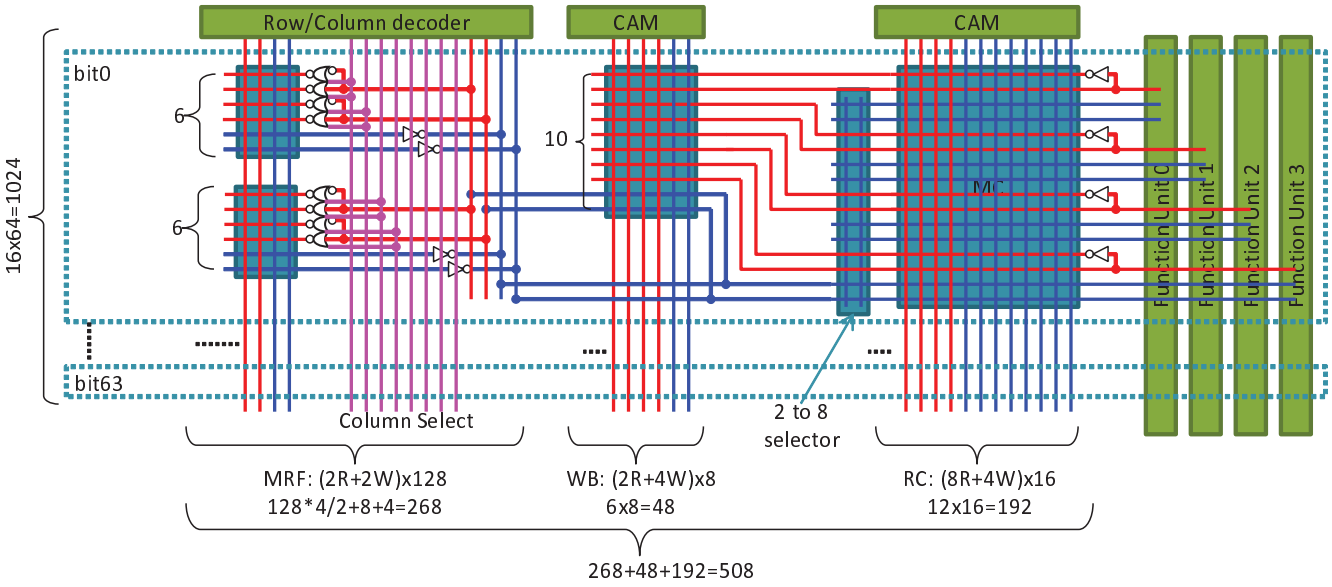

図 6 レジスタ・キャッシュ・システムのレイアウト

レジスタ・キャッシュ・システムでは，これらのリード・ポートとライト・ポートを共有するメモリ・セルが以下の理由で有利と考えられる．

- データ線共有によって面積が削減できる
- レジスタ・キャッシュによってリードのみが削減されるため，スループットが必要なのはライトのみである．
- ライトはライト・バッファが平滑化を行うため，リードと競合した場合に待たせることができる．

ただし，図 8 に示す面積は，配線本数から算出した値であり，特に，小ポートの小さなメモリ・セルでは，配線ではなくトランジスタのサイズで面積が決定されることには留意が必要である．[8] によれば，22nm プロセスで，最も密度の高い SRAM の面積は $0.092um^2$，最下層 M1 配線のピッチは 90nm であり面積は 45.4 程度となる．

図 8 に示した面積について**表 1** にまとめる．配線本数から計算した面積では 1(R/W) は圧倒的に小さいが，トランジスタサイズから決まる下限面積が 45.4 程度であるとすると，1(R/W) の面積は非現実的である．本稿では，この点は問題とせず，配線本数のみで計算された面積を用いるが，この点は，さらに検討が必要である．

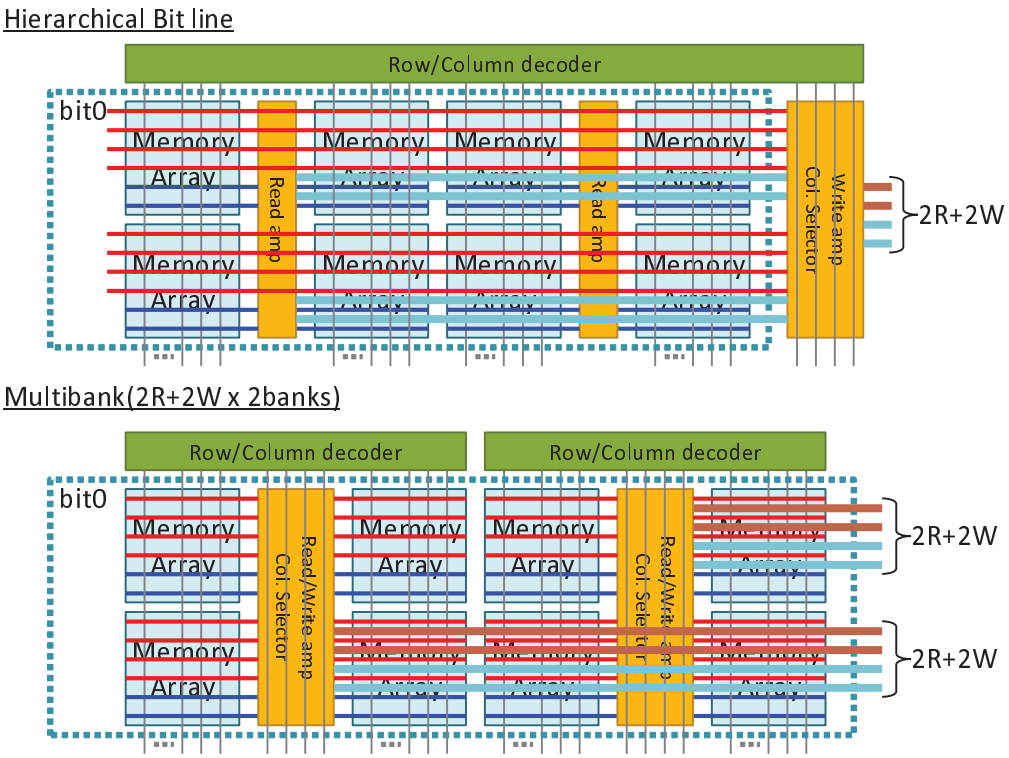

図 7　階層データ線及びマルチ・バンクのレイアウト

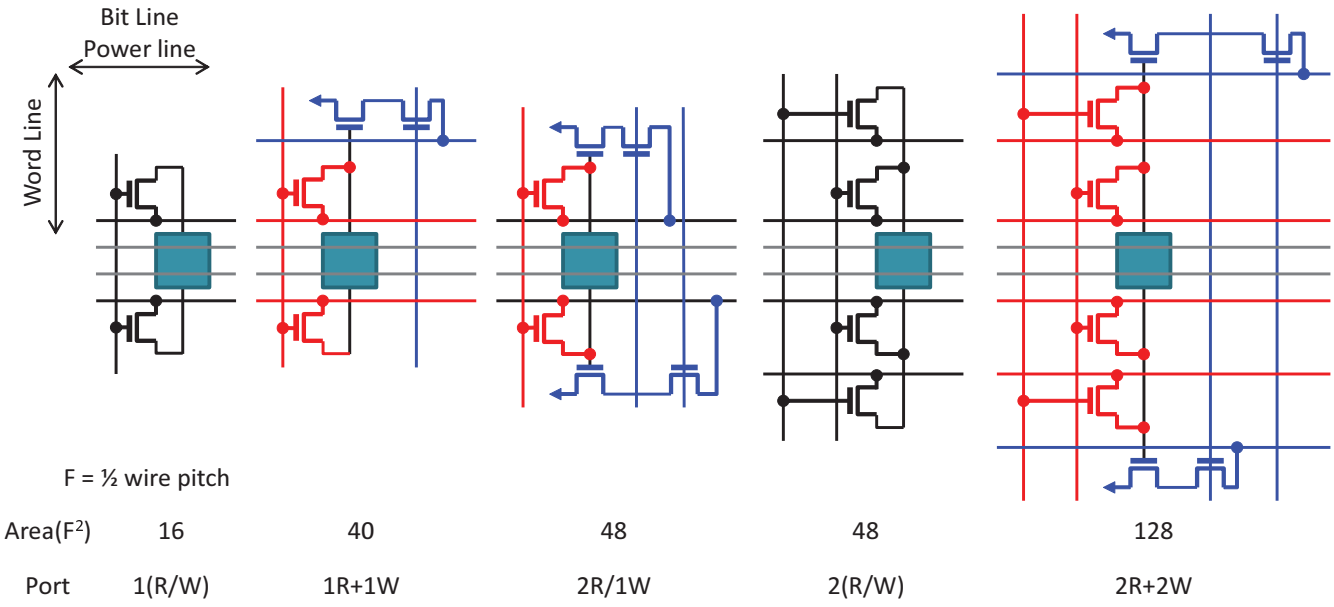

図 8　メモリ・セルの回路図

表 1　SRAM のメモリ・セル

ポート	面積 [(単位配線ピッチ/2)^2]
22nm 1(R/W)[8]	45.4
1(R/W)	16
1R+1W	32
2R/1W	48
2(R/W)	64
2R+2W	128

5. 評価

5.1 評価手法

5.1.1 面積評価

従来，RAM の面積及び消費電力の評価には CACTI [9] などが使用されていた．しかし，バンク，アレイの分割，ビット・スライス構成などについて自由な構成で見積もりを行うことは困難である．このため，本稿では，ポート数，

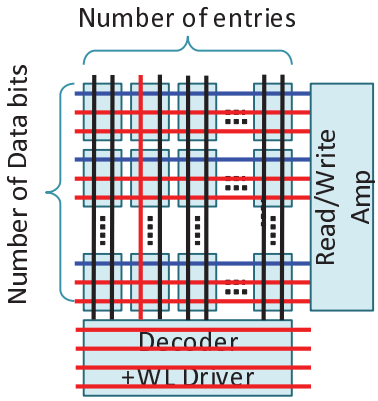

図 9 RAM のアレイ構造

エントリ数などから計算した配線本数と配線ピッチの積で，相対的な面積を算出することとした．

RAM の面積

メモリ・セルの構造については，4.4 で述べたとおりであり，ポート数によって配線本数が決まる．このメモリ・セルが**図 9** に示す構造のアレイを構成しているとして，RAM 全体の面積を見積もった．

アレイ上の X 方向配線はデータ線であり，ビット幅数 × ポート (×2) 数の配線が存在する．ライトのデータ線が面積計算で (×2) となるのは，相補信号を前提としているためである．

アレイ上の Y 方向配線はワード線であり，エントリ数 × ポート数の配線が存在する．

マルチ・バンク化 MRF の面積

マルチ・バンク化を行った MRF は，基本的には RAM と同様であるが，以下の 3 点について考慮が必要である．

- 1bit 高さを変えない状態で，レジスタ・キャッシュよりは少ないポート数の RAM を用いるため，メモリ・セルが折り畳まれカラム・セレクタが必要となる
- バンク毎にデータ・アンプ及びカラム・セレクタが必要である
- 分割された各バンクにアレイ上を上層の配線で接続する Global データ線が存在する

面積の見積もりの根拠となるバンクの構造を**図 10** に示す．図 10 は，合計 128 エントリの MRF が 4 バンクに分割されており，1bit の高さに 1(R/W) のメモリ・セルが 8 段折り畳まれている．

各バンクのデータ・アンプ及びカラム・セレクタの制御信号として，折り畳まれたメモリ・セルの選択信号とデータ・アンプのイネーブル制御信号が必要であり，これは面積の増大要因となる．折り畳まれたメモリ・セルの選択信号の配線本数は，$\lceil \log 2 \text{ 折り畳み数} + 1 \rceil \times$ ポート数 で与えられ，更に，データ・アンプの制御信号が必要である．図 10 の例では，8 段の選択信号に 3 本，データ・アンプの制御信号としてリード及びライトに 1 本ずつの合計 5 本となる．この制御信号本数は，ワード線の本数 4 本より多く面積見積もりにあたって無視できない要素である．各ポート構成毎に前提とした制御信号本数を**表 2** にまとめた．

Global データ線については，上層の配線を用いるため面積への影響は無いものとした．

5.1.2 性能評価

シミュレーションを行い，提案手法の性能について評価を行った．シミュレーションには本研究室で開発したシミュレータ「鬼斬弐」[10] を用いている．ベンチマークには，SPEC2006 の全プログラムについて，ref.0 パターンを使い，最初の 1G 命令をスキップし直後の 10M 命令を実行した．

評価条件

シミュレーションしたプロセッサの構成は**表 3** の通りである．ISA は ALPHA プロセッサであるが，プロセッサの構成は，POWER 7[11] など，近年の高性能プロセッサの構成を参考として決定した．プロセッサの構成の内，pipeline の regread が 2 or 3cycles，miss penalty が 15 or 16cycles となっているのは，後述の MRF Read cycle が 1 or 2cycle の場合にそれぞれ対応している．

レジスタ・キャッシュ・システムの評価条件は**表 4** に示す通りである．MRF 本体部分については，ポート構成とバンク及び，読み出しサイクル数の組み合わせを見ている．ポート構成の内，3R+3W, 4R+4W, 5R+5W は，バンク分割を行わない従来のレジスタ・キャッシュ・システムを想定した構成である．

レジスタ・キャッシュ部分については 0,4,8,16 エントリの 4 通りを見ている．RC0 エントリは，専用のレジスタ・キャッシュを使わないケースであるが，この場合でも，ライト・バッファからのライト・バッファ・フォワーディングは有効でありレジスタ・キャッシュ的に機能する．

5.2 評価結果

評価結果について簡単にまとめる．詳細は，次節以降に説明する．

面積の見積もり

ライト・バッファ及びレジスタ・キャッシュの占める比率が高いのが課題である．現在の評価範囲では，レジスタ・キャッシュエントリゼロとライト・バッファの組み合わせ

表 2 カラム・セレクタ及びデータ・アンプの制御信号

ポート	折り畳み数	配線本数		
		選択	データ・アンプ	合計
1(R/W)	8	3	2	5
1R+1W	6	6	2	8
2R/1W	8	6	2	8
2(R/W)	5	6	4	10

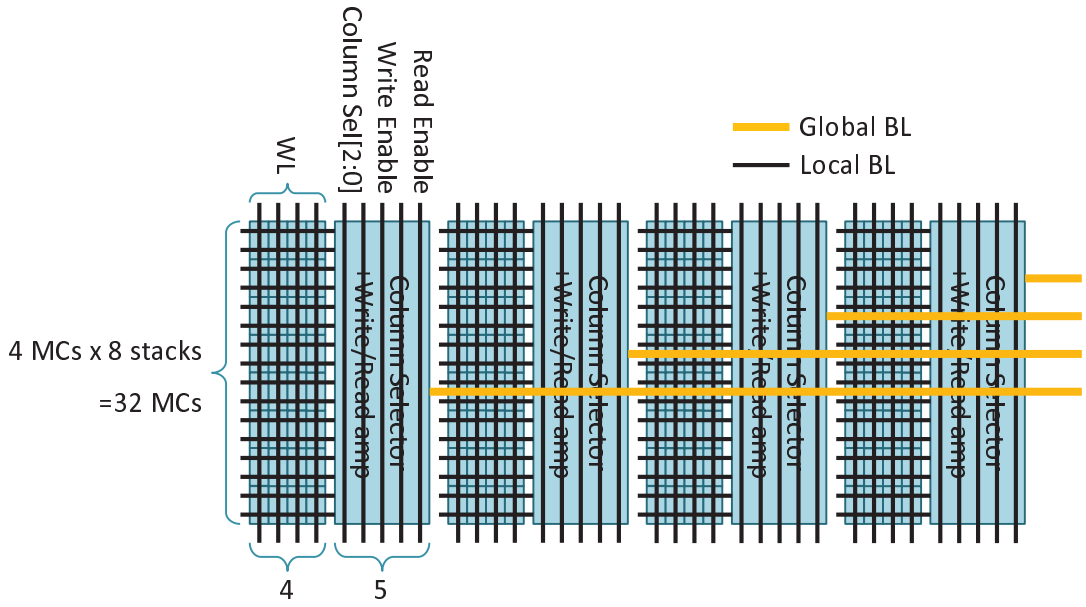

図 10 マルチ・バンク MRF の構造 1(R/W) × 32entries × 4 banks

が裁量となったが，バランスよく縮小できる構成を検討する必要があると考えられる．

同一エントリリードへの対応

直接 MRF に送らず 1 エントリのみリードして copy する構成が必須である．

表 3 シミュレーション・コンフィギュレーション

ISA	ALPHA
pipelins stages	fetch+decode:5, rename:2, dispatch:2, select:1, issue:2, regread:2 or 3
fetch/issue width	8 instructions
inst. window	64(unified)
inorer list	256 entries
branch pred.	64K:gshare+32K:local hybrid
miss penalty	15 or 16 cycles
BTB	2 Kentries, 4-ways
L1C	64KB, 8-way, 64B/line, 2cycles
L2C	512KB, 8-way, 64B/line, 8cycles
L3C	8MB, 8-way, 64B/line, 24cycles
main memory	200 cycles

表 4 レジスタ・キャッシュ・システム・パラメーター

論理レジスタ数	INT/FP=32/32 レジスタ
物理レジスタ数	INT/FP=160/160 レジスタ
WB エントリ数	16×2 本 (INT/FP)
RC エントリ数	0,4,8,16×2 本 (INT/FP)
MRF ポート	1(R/W), 1R+1W, 2R/1W, 2(R/W) 3R+3W, 4R+4W, 5R+5W
MRF バンク数	1,2,3,4,6,8,12,16 バンク
RC access (hit/miss dicision)	1 cycle
MRF Read cycle	1 or 2 cycle

2cycle-read の効果

1-read のメモリ・セルを使った MRF では効果があるが，2-read のメモリ・セルを使った MRF では，Latency の増大による性能低下の影響が上回る．

最適なレジスタ・キャッシュ容量とポート構成

相対 IPC≥99%となるのは，2R/1W 又は 2(R/W) の場合のみであり，相対 IPC≥99%の範囲で，面積性能比が高いのは，2(R/W)×6 バンク,RC0 で，この条件の面積は 19.8%，相対 IPC は 99.0%であった．

なお，ここで，面積は，16-read+8-write のポート制約が発生しない巨大な MRF 面積を 100%とした場合の相対面積で表している．相対 IPC も，同様にポート制約が発生しない MRF を使用した際の IPC を 100%とした相対 IPC で示す．

5.2.1 面積の見積もり

面積を，ライト・バッファ MRF(メモリ・セル及びデータ・アンプ)，レジスタ・キャッシュの項目ごとに見積もった結果を**図 11** に示す．データ・アンプには，リード・アンプ, ライト・アンプ及びカラム・セレクタの面積が含まれる．

この結果によると，ライト・バッファ及びレジスタ・キャッシュの占める比率が高く，MRF の本体部分であるメモリ・セルやデータ・アンプの割合は相対的に小さい．

後述の面積性能比の評価では，評価した範囲では，巨大なライト・バッファと RC エントリゼロが最良であったが，ライト・バッファを簡素化して小容量の RC を用いるようなバランスの良い構成も検討する必要がある．

5.2.2 同一エントリリードへの対応

MRF の同一エントリに対するリードが複数発生した場

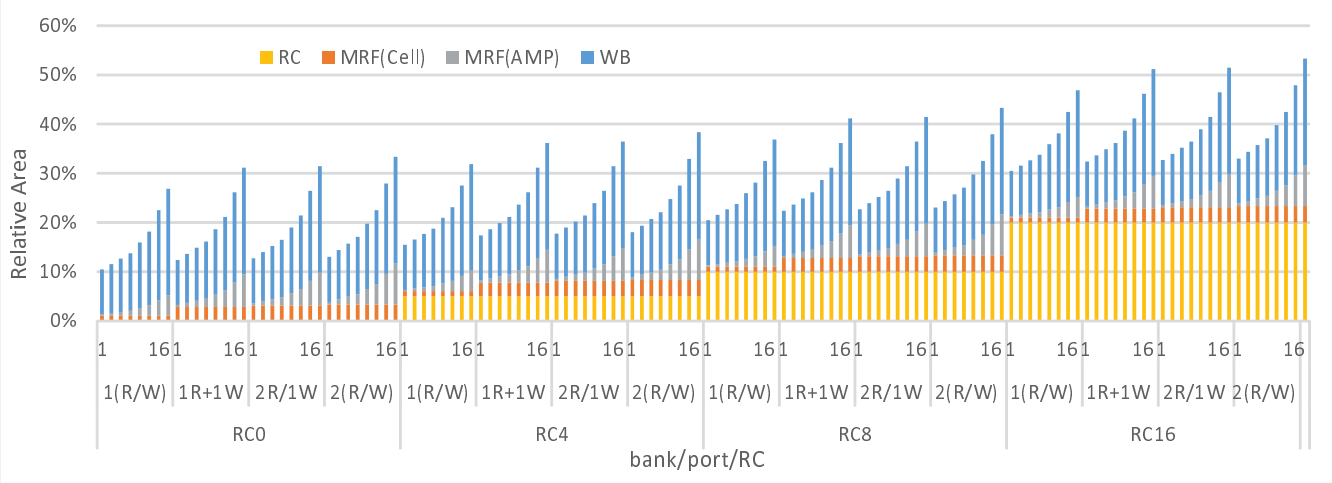

図 11 Area breakdown

合，それぞれを別々に MRF からリードするか，セレクタなどで結果をコピーするかは，バンク・コンフリクトに大きく影響すると考えられる．なぜなら，同一エントリに対するリードは必然的に同一バンクへのリードとなり，バンク分割数を増やしても軽減しないためである．

結果を**図 12** に示す．グラフ中の mrfread が MRF からそのままリードする場合，copy が 1 エントリ分のみリードして結果をコピーする場合である．

1-read/bank である 1(R/W) 及び 1R+1W と，2-read/bank である 2R/W 及び 2(R/W) では性能向上の程度が異なるが，いずれにせよ，結果をコピーする方式の方が，性能が高くなる．

MRF の少ないリード・ポートから，多数ある演算器へソース・オペランドを供給するセレクタは，いずれにせよ必須の要素であり，コピーは制御のみの変更で行え特に副作用も無いため，同一エントリリードは，1 エントリのみリードしてコピーする方針が正しいと考えられる．

このため，次節以下の結果では，全て，”1 エントリのみリードしてコピー”の結果を示す．

5.2.3 2cycle-read の効果

次に MRF のリードを 1cycle で行う場合と 2cycle で行う場合の効果について検討する．2cycle で行う方がバンク・コンフリクトが減少すると期待される一方で，MRF Read Latency の増加による IPC 低下が副作用として考えられる．

結果について，相対 IPC を**図 13** に，ストールの内訳を**図 14** に示す．図 13 は，X 軸がバンク数，Y 軸に相対 IPC を取っている．図 14 は，同様に X 軸がバンク数であり，Y 軸には，ストールの発生回数を，MRF Read, WB Full, WB Forwarding の 3 通りの分類で示している．

ストール発生回数は実行サイクル数で割った相対値としており，この値は，概ね IPC の低下率と対応するが，同一のサイクルに複数の原因でストールした場合は重複してカウントしているため厳密ではない．

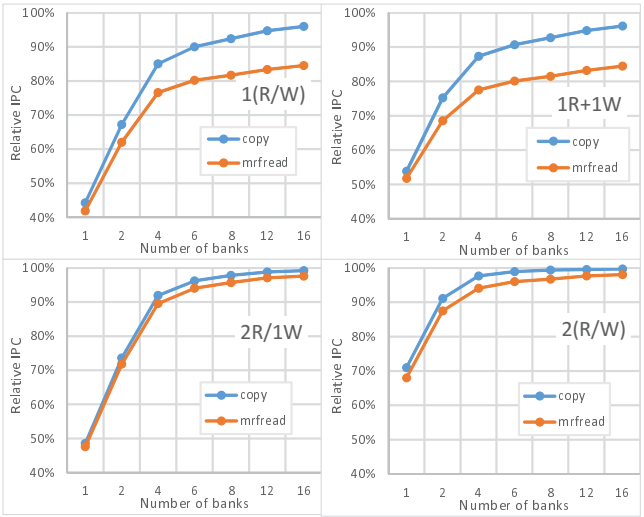

図 12 Effect of same entry read

図 13 の相対 IPC を見ると，1bank あたりのリード・ポート数によって結果が異なり，1-read/bank である 1(R/W) 及び 1R+1W では，2-cycle read の方が性能が高い．これに対して，2-read/bank である 2R/W 及び 2(R/W) では，2-cycle read によってかえって性能が低下している．

図 14 のストール内訳を見ると，青で示す MRF Read 起因 Read は，いずれの条件でも 2-cycle read で減少している．従って，2-read/bank である 2R/W 及び 2(R/W) でかえって性能が低下するのは，Latency 増加に伴う性能低下に起因していると考えられる．

5.2.4 最適なポート構成及びレジスタ・キャッシュ容量

次に，最適なポート構成とレジスタ・キャッシュ容量について検討する．この検討に当たっては，性能だけでなく，面積を考慮する必要がある．

ポート構成・バンク数・RC 容量毎の相対 IPC

ポート構成毎の相対 IPC のグラフを，**図 15** に示す．ポート構成は，左上，右上，左下，右下の順に，それぞれ，

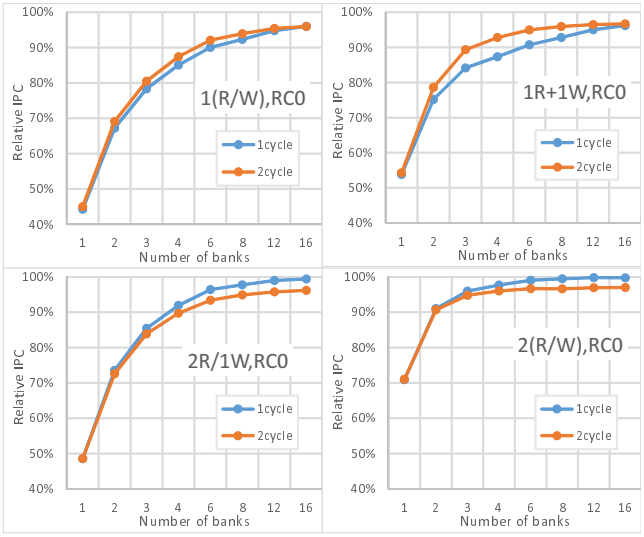

図 13 2cycle-read の効果 (相対 IPC)

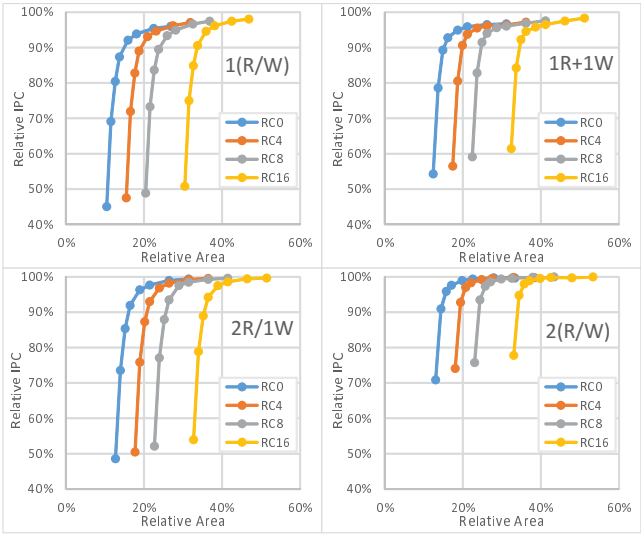

図 15 面積性能比 (ポート構成・RC 容量別)

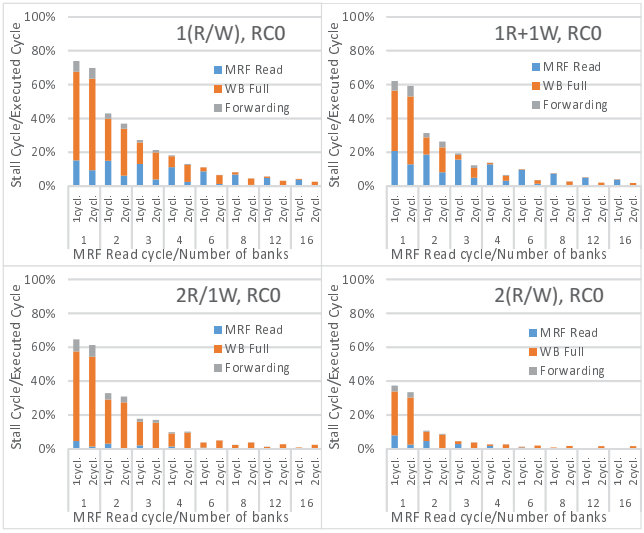

図 14 2cycle-read の効果 (ストール内訳)

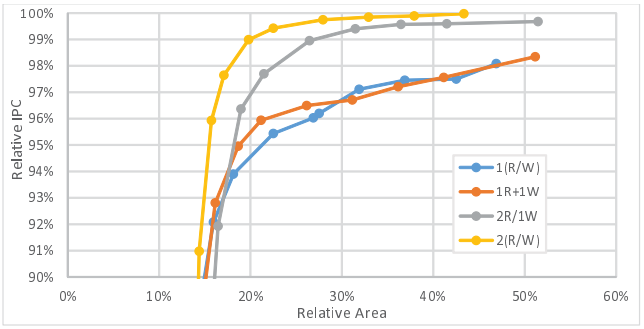

図 16 面積性能比 (各ポート構成の最良ポイント)

1(R/W), 1R+1W, 2R/1W, 2(R/W) である.

グラフの各系列はレジスタ・キャッシュ容量に対応しており，各系列の点がそれぞれ 1...16 バンクまでの結果を示している．横軸は面積であり，当然バンク数が多い側及びレジスタ・キャッシュ容量が多い側で面積は大となるため，各系列の右側が 16 バンク, RC エントリ 16 である．相対 IPC は，それぞれの結果について，1cycle-read 及び 2cycle-read の良い側の数字を選択している．

レジスタ・キャッシュ容量に対して相対 IPC は向上している．特に，バンク数が多い場合に相対 IPC が飽和する位置が上昇する．しかし，レジスタ・キャッシュ容量増加で面積も大きく増大している．

ポート構成毎の相対 IPC

図 15 の結果から，4 通りのポート構成別に，同じ面積に対して相対 IPC が最も良いバンク数及び RC 容量の組み合わせを抽出した結果を，**図 16** に示す．

相対 IPC≥99%となるのは，2R/1W 又は 2(R/W) の場合のみであり，相対 IPC≥99%の範囲で，面積性能比が高いのは，2(R/W)×6 バンク,RC0 で，この条件の面積は 19.8%, 相対 IPC は 99.0%であった．

これを，バンク分割を行わない従来のレジスタ・キャッシュ・システムと比較する．

バンク分割を行わない従来のレジスタ・キャッシュ・システムについて，同様に求めたポート構成毎の相対 IPC を**図 17** に示す．バンクはいずれも 1 バンクであり，RC 容量は同様に 0,4,8,16 の 4 通りを見ている．

図 17 では，相対 IPC≥99%となる構成は，4R+4W の RC16 または，5R+5W の RC0 以上であった．この場合の面積は，5R+5W/RC0 について 51.3%，4R+4W/RC16 について 52.1%であった．また，4+4W/RC8 の構成も相対 98.9%とわずかに 99%に及ばないが, 面積は 42.1%である．

いずれの結果と比較しても, 本稿のバンク分割 MRF では，相対 IPC≥99%の面積が 19.8%であるから，より高い面積性能効率を実現したといえる．

各構成の性能低下要因

ポート構成とレジスタ・キャッシュ容量の組み合わせについて，ストールの内訳を**図 18** 及び**図 19** に示す．

図 18 は，各ポート構成の全バンクの結果を示した．いずれのポート構成，RC 容量であっても，バンク数の少な

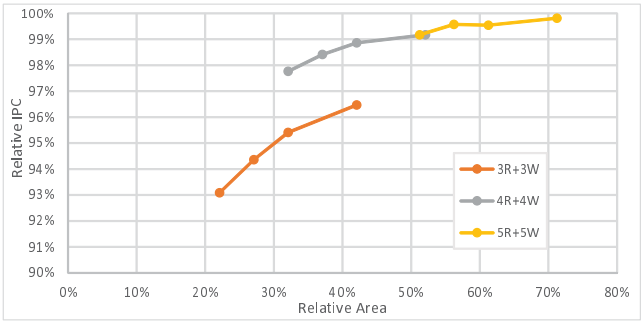

図 17 面積性能比 (バンク分割無し)

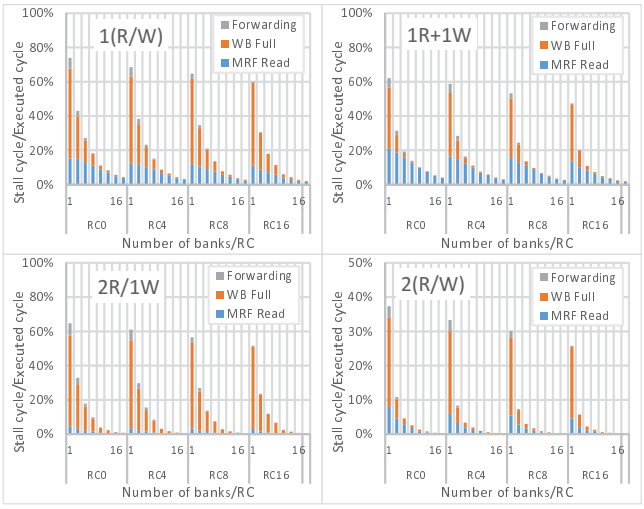

図 18 ストールの内訳 (全バンク数)

い領域ではオレンジで示す WB Full 起因のストールが多い．これは，RC は，ライト・スルーであるため，MRF にライト・ポート数 × バンク数で表されるライト・スループットが一定程度必要であることを示している．

図 19 は，図 18 から，ポート構成毎に，相対 IPC≥95%となる最小のバンク数を抜き出した．具体的には，1(R/W), 1R+1W, 2R/1W, 2(R/W) に対して，それぞれ，12, 8, 6, 3 バンクの場合の結果である．RC の容量増加に，青で示す MRF Read 起因ストールを減少させる効果があることが分かる．

ソース・オペランドの供給元

次に，ソース・オペランドの供給元について，**図 20** に示す．

評価したプロセッサ構成でソース・オペランドを供給できるのは，オペランド・バイパス, レジスタ・キャッシュ, ライト・バッファ・フォワーディング,MRF であり，ソース・オペランドをこれら 4 つの要素が，どのような割合で供給したかを，INT 系/FP 系レジスタ別に，全プログラムの平均で示した．条件は，面積性能比が高かった，2(R/W)×6 バンクの場合であり，RC 容量 0,4,8,16 の各条件を示している．

結果を見ると，INT/FP ともに，全体の約半分程度はオペランド・バイパスで供給されている．残りは，RC 容量

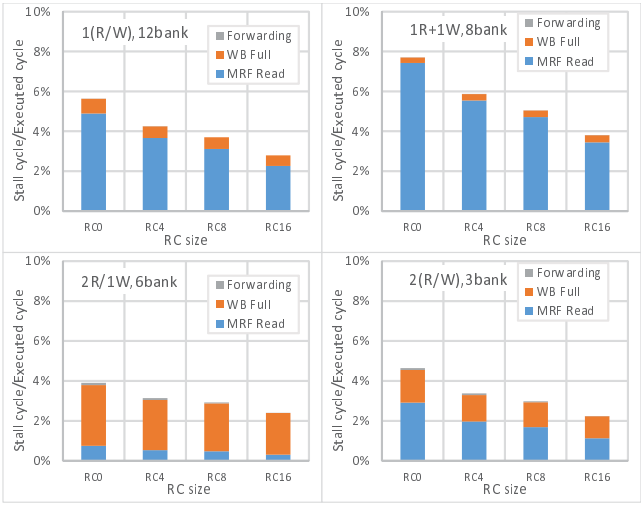

図 19 ストールの内訳 (相対 IPC≥95%となる最小のバンク数)

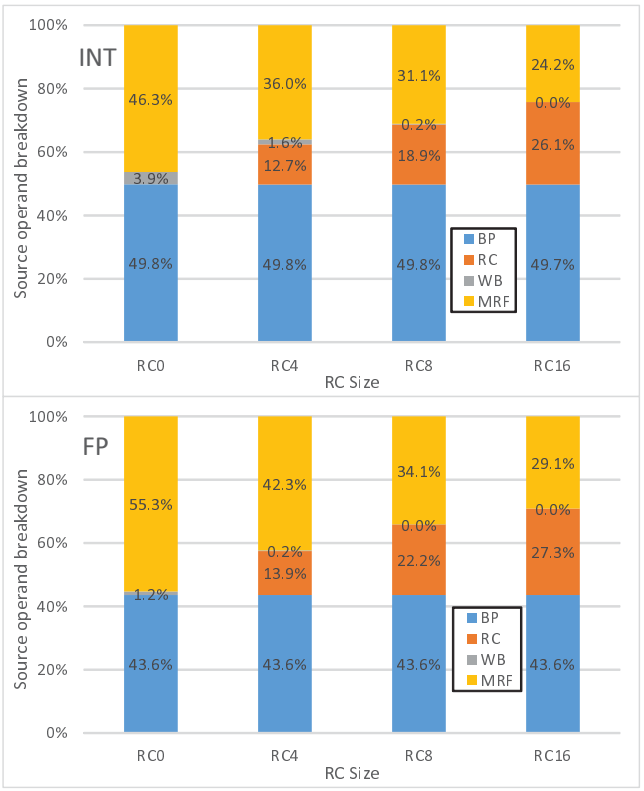

図 20 ソース・オペランドの供給元内訳

によって異なり，RC 容量 16 では，全体の約 1/4 が RC から供給され，残りが MRF である．

RC が少ない領域では，当然 MRF からの供給が増加するが，この他に WB からの供給が発生する．特に INT 側で目立ち，RC 容量 0 の INT 側では，3.9%を WB が供給している．本稿の評価ではライト・バッファ・フォワーディングを行っているため，この 3.9%は，直接的に IPC の低下要因とはならないが，ライト・バッファを簡素化してライト・バッファ・フォワーディングを行わない場合には IPC 低下要因となる．

6. おわりに

本稿では，マルチ・バンク化によって，メイン・レジスタ・ファイルの面積を削減する手法を提案した．

今後の課題を 3 点述べる．

第一に，面積に占めるライト・バッファの割合が高すぎる問題について，再検討が必要である．現在は，ライト・バッファはポート数不足とならない十分なポート数を割り当て，更にライト・バッファ・フォワーディングも行っているため，巨大である．ライト・バッファへのライトが止まると直ちにストールするが，リード側は柔軟性を持たせることが可能と考えられる．また，ライト・バッファ・フォワーディングによってライト・バッファが RC 的に動作することで，RC の効果が見えにくくなっているが，全体の構成としては，ライト・バッファを簡素化し，代わりに小容量の RC を使う方が効率的であることが考えられる．

第二に，物理レジスタ数の多いマルチスレッド・プロセッサなど，より大規模な物理レジスタ・ファイルを持つプロセッサへの対応が挙げられる．マルチスレッド・プロセッサでは，スレッド数分に対応してサイズの大きい物理レジスタ・ファイルを持つ．このようなプロセッサでは面積の削減効果はより大きくなることが期待される．

最後に，これらに加え，消費電力の観点からも評価を行う必要がある．

謝辞 本論文の研究は一部，文部科学省科学研究費補助金 No. 23300013, 26280012 による．

参考文献

[1] Shioya, R., Horio, K., Goshima, M. and Sakai, S.: Register Cache System Not for Latency Reduction Purpose, *2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 301 –312 (online), DOI: 10.1109/MICRO.2010.43 (2010).

[2] 西川卓，倉田成己，塩谷亮太，五島正裕，坂井修一：レジスタ・キャッシュのマルチスレッド・プロセッサへの適用，情報処理学会第 75 回 全国大会 (2013).

[3] 西川卓，倉田成己，塩谷亮太，五島正裕，坂井修一：マルチスレッド・プロセッサにおけるレジスタ・キャッシュ・システムの評価，情報処理学会研究報告 (発表予定) (2013).

[4] Shin, J. L., Golla, R., Li, H., Dash, S., Choi, Y., Smith, A., Sathianathan, H., Joshi, M., Park, H., Elgebaly, M., Turullols, S., Kim, S., Masleid, R., Konstadinidis, G., Doherty, M., Grohoski, G. and McAllister, C.: The Next Generation 64b SPARC Core in a T4 SoC Processor, *Solid-State Circuits, IEEE Journal of*, Vol. 48, No. 1, pp. 82–90 (online), DOI: 10.1109/JSSC.2012.2223036 (2013).

[5] Park, I., Powell, M. and Vijaykumar, T. N.: Reducing register ports for higher speed and lower energy, *Microarchitecture, 2002. (MICRO-35). Proceedings. 35th Annual IEEE/ACM International Symposium on*, pp. 171–182 (online), DOI: 10.1109/MICRO.2002.1176248 (2002).

[6] Hironaka, T., Maeda, M., Tanigawa, K., Sueyoshi, T., Aoyama, K., Koide, T., Mattausch, H. and Saito, T.: Superscalar processor with multi-bank register file, *Innovative Architecture for Future Generation High-Performance Processors and Systems, 2005*, pp. 10 pp.– (online), DOI: 10.1109/IWIA.2005.42 (2005).

[7] Patwary, A., Greub, H., Wang, Z. and Geuskens, B.: Bit-Line Organization in Register Files for Low-Power and High-Performance Applications, *Electrical and Computer Engineering, 2006. ICECE '06. International Conference on*, pp. 505–508 (online), DOI: 10.1109/ICECE.2006.355679 (2006).

[8] Jan, C.-H., Bhattacharya, U., Brain, R., Choi, S.-J., Curello, G., Gupta, G., Hafez, W., Jang, M., Kang, M., Komeyli, K., Leo, T., Nidhi, N., Pan, L., Park, J., Phoa, K., Rahman, A., Staus, C., Tashiro, H., Tsai, C., Vandervoorn, P., Yang, L., Yeh, J.-Y. and Bai, P.: A 22nm SoC platform technology featuring 3-D tri-gate and high-k/metal gate, optimized for ultra low power, high performance and high density SoC applications, *Electron Devices Meeting (IEDM), 2012 IEEE International*, pp. 3.1.1–3.1.4 (online), DOI: 10.1109/IEDM.2012.6478969 (2012).

[9] Thoziyoor, S., Ahn, J.-H., Monchiero, M., Brockman, J. and Jouppi, N.: A Comprehensive Memory Modeling Tool and Its Application to the Design and Analysis of Future Memory Hierarchies, *Computer Architecture, 2008. ISCA '08. 35th International Symposium on*, pp. 51–62 (online), DOI: 10.1109/ISCA.2008.16 (2008).

[10] 塩谷亮太，五島正裕，坂井修一：プロセッサ・シミュレータ「鬼斬弐」の設計と実装，先進的計算基盤システムシンポジウム SACSIS，pp. 120–121 (2009).

[11] Sinharoy, B., Kalla, R., Starke, W. J., Le, H. Q., Cargnoni, R., Van Norstrand, J. A., Ronchetti, B. J., Stuecheli, J., Leenstra, J., Guthrie, G. L., Nguyen, D. Q., Blaner, B., Marino, C. F., Retter, E. and Williams, P.: IBM POWER7 multicore server processor, *IBM Journal of Research and Development*, Vol. 55, No. 3, pp. 1:1–1:29 (online), DOI: 10.1147/JRD.2011.2127330 (2011).