

温度を考慮するアーキテクチャの検討のための物理 レジスタアクセス頻度の特徴調査

佐藤, 寿倫
九州大学 システムLSI研究センター

国武, 勇次
九州工業大学大学院 情報工学研究科

千代延, 昭宏
九州工業大学大学院 情報工学研究科

<https://hdl.handle.net/2324/6326>

出版情報：研究報告ARC. 2006 (62), pp.37-42, 2006-06. 情報処理学会
バージョン：

権利関係：ここに掲載した著作物の利用に関する注意 本著作物の著作権は（社）情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。

温度を考慮するアーキテクチャの検討のための 物理レジスタアクセス頻度の特徴調査

佐藤 寿倫[†] 国武 勇次[‡] 千代延 昭宏[‡]

[†]九州大学 システム LSI 研究センター 〒814-0001 福岡市早良区百道浜 3-8-33-3F

[‡]九州工業大学大学院 情報工学研究科 〒813-0036 飯塚市川津 680-4

E-mail: [†] toshinori.sato@computer.org, [‡] {y-kunitake,chiyo}@mickey.ai.kyutech.ac.jp

あらまし 動作周波数上昇と集積度向上の恩恵によってマイクロプロセッサは性能向上を図ってきたが、近年消費電力の増加とその過密化による温度上昇が大きな問題となっている。本研究ではレジスタアクセスによる温度の上昇に着目している。特定のレジスタやレジスタバンクへの集中的なアクセスは、著しい発熱を伴うからである。以前の調査で物理レジスタへのアクセスは均等に行われていることを確認したが、時間的な変動は考慮されていない。今回、時間を区切ってアクセス頻度を調査したところ、著しいアクセス集中が確認された。

キーワード レジスタファイル, レジスタリネーミング, 消費電力, ホットスポット, 熱設計

Physical Register Access Analysis for Temperature-Aware Architectures

Toshinori Sato[†] Yuji Kunitake[‡] and Akihiro Chiyonobu[‡]

[†] System LSI Research Center, Kyushu University, Japan

[‡] Department of Artificial Intelligence, Kyushu Institute of Technology, Japan

E-mail: [†] toshinori.sato@computer.org, [‡] {y-kunitake,chiyo}@mickey.ai.kyutech.ac.jp

Abstract While the improvements in clock frequency and transistor density have achieved the continuous increase in microprocessor performance, increasing power consumption and thus temperature are becoming a serious problem. We focus on register files, because strong access locality to particular registers or register banks causes serious heat dissipation. Our previous study did not show such an access locality, because register renaming transforms the logical register access into the uniform physical register access. However, it did not consider temporal locality but only spatial locality. This paper shows strong temporal and spatial locality on physical register access.

Keyword Register files, register renaming, power consumption, hot spots, thermal design

1. はじめに

マイクロプロセッサの設計において、近年発熱が問題視されつつある[1]。電力消費量が増加するとともに、微細化により電力の過密化が進んでいる。例えば、マイクロプロセッサ内の電力密度が、原子炉内部のそれに近づきつつあるという報告がある。インテルが次世代ペンティアム 4 の開発をキャンセルしたのは、消費電力の上昇とそれに伴う熱の上昇、そしてそれらの結果として起こる信頼性の低下、これらの問題を解決できなかったためであると想像されている。

中でも利用頻度の高さから、レジスタファイルにおける発熱が問題となっている[1]。温度は消費電力に比例するが、利用頻度が高いことは電力消費量も大きいことを意味するからである。このため、レジスタファイルの消費電力を削減するための様々な方式が検討されている[2, 3, 4, 5, 6]。さらには、高温となったレジスタ

を冷やすために、物理的に位置の異なるレジスタを利用することも検討されている[1, 7]。一方で、マルチスレッドプロセッサの出現などにより、レジスタファイルの容量とポート数に対する要求はますます大きくなっている。大容量かつ多ポートのレジスタファイルはアクセスレイテンシも大きくなるため、レイテンシ削減のためにマルチバンク化が提案されており[8]、市場には出なかったものの旧コンパック社の 21464 では実際に採用している[9]。加えてマルチバンク化は電力削減にも有効であるとされている[2, 3, 4]。

レジスタファイルにおける発熱で憂慮すべき問題は、特定のレジスタあるいはレジスタバンクへのアクセス集中である。温度は放熱部の面積に反比例するので、発熱部の面積が小さくなると温度上昇が激しくなる。つまり、特定のレジスタあるいはレジスタバンクへのアクセス集中は放熱部の面積が小さくなることに

相当するので、著しい温度上昇を伴うことになり問題である。バンク化は熱の観点からは考慮を要する。

以上を考慮して、本稿ではレジスタアクセスの特徴を調査し、温度を考慮するアーキテクチャ検討の足がかりとする。

2. レジスタアクセス

アウトオブオーダー実行するスーパーカラプロセッサが、逐次プログラムから並列実行可能な命令を抽出するためには、命令間の出力依存関係と逆依存関係を解消する必要がある。この目的でレジスタリネーミングが利用され、同一の論理レジスタに異なる物理レジスタが割り当てられる。本稿では、論理レジスタとリネーミング用レジスタを、同一の物理レジスタで共有するレジスタリネーミングを扱う。マップ表を用いて論理レジスタから物理レジスタへの変換を行うことを想定している。

プログラムでは、特定のレジスタを頻繁に使用することが起こり得る。例えば、スタックポインタを特定のレジスタに割り当てることが推奨される場合がある。スタックポインタは頻繁に使用されるので、特定のレジスタへアクセスが集中し得る。しかしレジスタがリネーミングされると、プログラム中で使用されるレジスタとプロセッサ内部で実際に使用されるレジスタとが異なることになる。例えば、スタックポインタも異なる物理レジスタへ割り当てられるので、そのためのレジスタへのアクセス集中が解消され得る。

上記を考慮して、私たちは[10]において物理レジスタへのアクセス頻度を調査した。その結果、物理レジスタへのアクセスは均等に行われていることを確認した。しかしこの調査では時間的な変動を考慮していなかった。本稿では時間的なレジスタアクセスの偏りを調査する。

3. シミュレーション環境

SimpleScalar ツールセットを拡張した sim-alpha[11] を利用してシミュレーションを実施した。sim-alpha は HP 社の 21264 プロセッサ[12]を忠実に再現している。そのプロセッサ構成を表 1 に示す。21264 はマップ表を用いてレジスタリネーミングを実現しており、本調査に適している。21264 は 80 エントリの整数レジスタファイル¹と 72 エントリの浮動小数点レジスタファイルを用意しているが、本調査ではいずれも 64 本の場合と 128 本の場合でシミュレーションを実施した。

¹ 21264 プロセッサは 80 エントリの整数レジスタファイルを二つ持ち、合計 160 本のレジスタを備えるが、それらは互いのコピーであり、物理レジスタとして使用可能なのは半分の 80 本である。

表 1: プロセッサ構成

フェッチ幅	4 命令
分岐予測	1K-local+4K-global
リネーミング幅	4 命令
イシュー幅	INT: 4 命令, FP: 2 命令
コミット幅	11 命令
命令キュー	INT: 20 命令, FP: 15 命令
ロードキュー	32 命令
ストアキュー	32 命令
リオーダーバッファ	80 命令
整数演算器	ALU: 4, 乗除算器: 4
浮動小数点演算器	ALU: 1, 乗算器: 1
L1 命令キャッシュ	64KB, 2 ウエイ
L1 データキャッシュ	64KB, 2 ウエイ
victim キャッシュ	8 エントリ
L2 キャッシュ	2MB, ダイレクトマップ

ベンチマークプログラムには、SPEC2000 より整数系プログラムから 6 本と、浮動小数点系プログラムから 11 本を選んだ。入力ファイルには ref を用いている。用いたプログラムとその入力ファイルを表 2 に示す。

表 2: ベンチマークプログラム

164.gzip	input.graphic	300M
186.crafty	crafty.in	0
197.parser	ref.in	1.6B
252.eon	rushmeier	1.8B
256.bzip2	input.program	900M
300.twolf	ref	3.1B
168.wupwise		58.4B
171.swim	swim.in	500M
172.mgrid	mgrid.in	600M
173.applu	applu.in	1.8B
177.mesa	mesa.ppm	8.9B
178.galgel	galgel.in	315B
179.art		6.7B
187.facerec	ref.in	13.6B
188.ammmp	ammmp.in	213B
189.lucas	lucas2.in	3.5B
200.sixtrack	inp.in	8.2B

SimPoint 2.0[13]の指定する 100M 命令を実行した。表 2 にはシミュレーション開始時からスキップする命令数も示されている。レジスタアクセス回数の収集は 1M サイクル毎に行った。

4. 結果

本節でシミュレーション結果を紹介する。まず、レジスタ数が 64 の場合と 128 の場合で大きな違いは見られなかった。そのため以下では、レジスタ数が 64 の場合のみを紹介する。またページ数制約の都合により、詳細な結果については代表的なプログラムの結果のみを示す。

興味深いことに、レジスタアクセスの特徴は、大きく以下の三種類に分けることができた。

- A. 全てのレジスタに均等にアクセスする
- B. 特定のレジスタにアクセスが集中する
- C. 特定のレジスタがアクセスされない

以上の観点でシミュレーション結果を分類したものを、表3にまとめる。

表3：レジスタアクセスの特徴

	整数レジスタ	FPレジスタ
164.gzip	B	
186.crafty	B	
197.parser	C	
252.eon	A	
256.bzip2	C	
300.twolf	A	
168.wupwise	A	B
171.swim	A	A
172.mgrid	B	C
173.applu	C	C
177.mesa	A	A
178.galgel	A	A
179.art	C	C
187.facerec	C	C
188.amp	C	C
189.lucas	C	C
200.sixtrack	A	A

まず整数系プログラムの結果を紹介する。図1は252.eonの結果である。横軸は時間の経過を表している、縦軸は、全整数レジスタのアクセス総数を100%として、64本の各レジスタのアクセス回数の割合を示している。全てのレジスタに対して時間的な偏りも無く均等にアクセスされていることが分かる。このようなアクセスであれば、特定のレジスタが高温になる様な事は無く、熱設計の観点から好ましい。

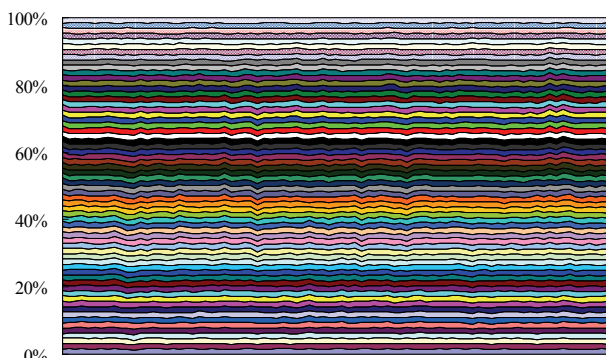


図1：eonにおける整数レジスタアクセス

図2は186.craftyの結果である。いくつかの区間で、ある特定のレジスタへ集中的にアクセスが発生していることがわかる。この場合、それら特定のレジスタがホットスポット的に高温になると予想される。ただし、他の区間と比較してアクセス総数が少なければ問題が無いと思われる。

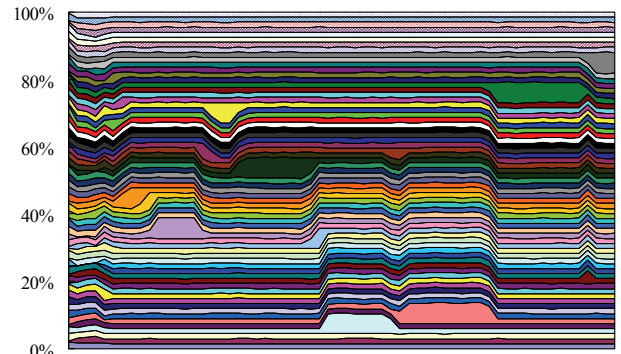


図2：craftyにおける整数レジスタアクセス

図3にレジスタアクセスの総数を示す。アクセス集中の観察される区間で、特にアクセス総数が減少してはいない。したがって、これらのアクセス集中は熱設計の観点からは問題となる。

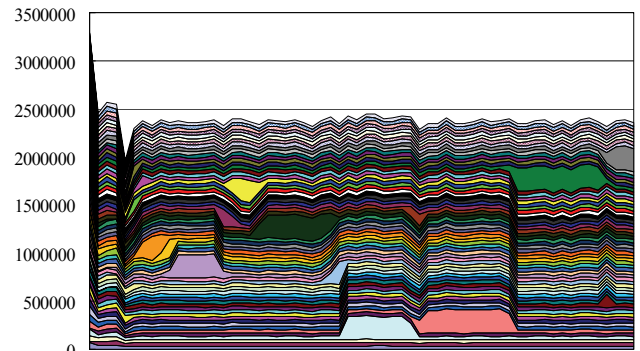


図3：craftyにおけるレジスタアクセス回数

レジスタリネーミングのアルゴリズムを考慮すれば、アクセス集中の原因はリードアクセスである。なぜなら、ある論理レジスタに割り付けられた物理レジスタには、ライトアクセスは一度しか生じないためである。図4にリードアクセスの分布を、図5にライトアクセスの分布を示す。これらの図より、リードアク

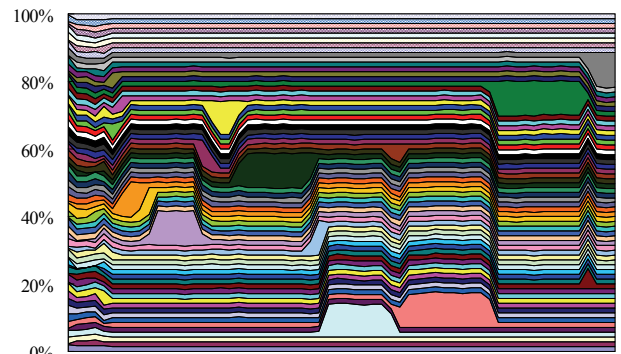


図4：craftyにおけるリードアクセス

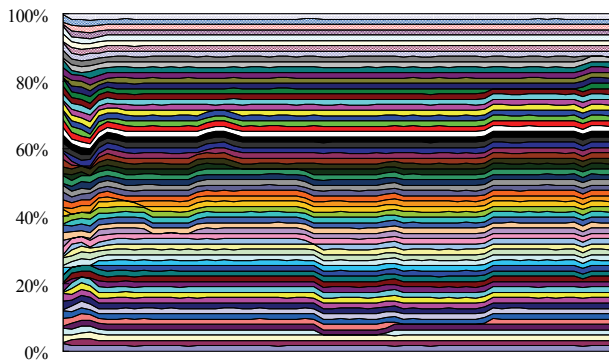


図 5 : crafty におけるライトアクセス

セスがアクセス集中の原因であることが明らかである。また図 5 より、アクセス集中の生じる区間で、該当するレジスタへのライトアクセスが生じていないことが観察される。正しくレジスタリネーミングが動作していることが確認できる。

図 6 と図 7 に 256.bzip2 におけるレジスタアクセスの分布と総数を示す。図 6 のほぼ中央以降で、各レジスタアクセスの割合を示す幅が広がっていることが観察される。これは、特定のレジスタがアクセスされなくなっているためである。しかし図 7 からは、アクセスの特徴に変化が生じる境界の前後で、アクセス総

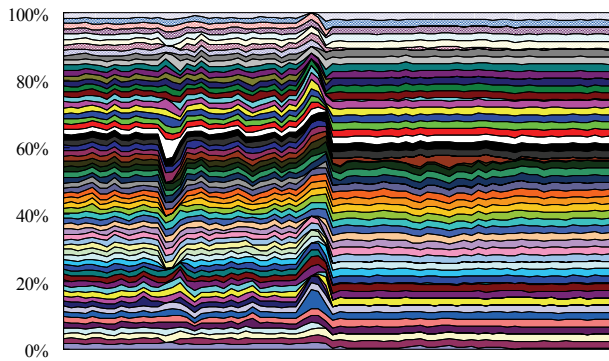


図 6 : bzip2 における整数レジスタアクセス

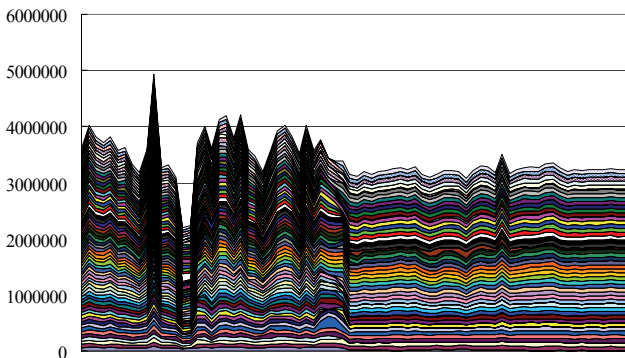


図 7 : bzip2 におけるレジスタアクセス回数

数が著しくは異なっていないことが確認できる。これは参照レジスタへのアクセス数増を意味している。

crafty に関する考察で述べたように、レジスタアクセスの特徴で支配的なのはリードアクセスである。該当するレジスタは参照されないにもかかわらず、他の論理レジスタに再割り当てされていない。再割り当てされないのは、現在割り当てられている論理レジスタへの書込みが起らないためである。つまり、該当する論理レジスタ内の値は、何らかの理由でオーバライトされることなく保持され続けているわけである。

続いて、浮動小数点系プログラムの結果を紹介する。図 8 と図 9 は 168.wupwise での浮動小数点レジスタの結果である。特定のレジスタにアクセスが集中している区間が認められる。図 3 の crafty の場合と異なるのは、区間によりアクセス総数に大きな違いがあることである。アクセス集中の観察される区間ではアクセス総数は小さいが、それでもなお、集中の生じるレジスタへのアクセス回数は大きく、該レジスタがホットスポットとなることが懸念される。整数レジスタでは均一的なアクセスとなっていることが確認されている。

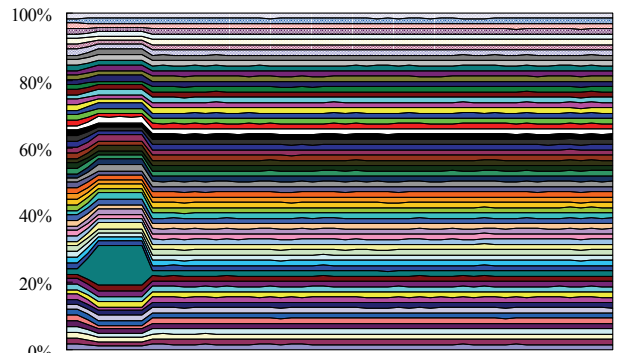


図 8 : wupwise における FP レジスタアクセス

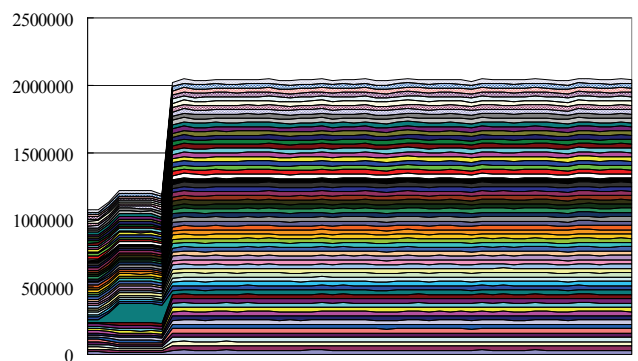


図 9 : wupwise における FP レジスタアクセス回数

図 10 と図 11 は 172.mgrid における整数レジスタの結果である。アクセス総数に変動があるが、特定のレジスタへのアクセス集中が観察できる。

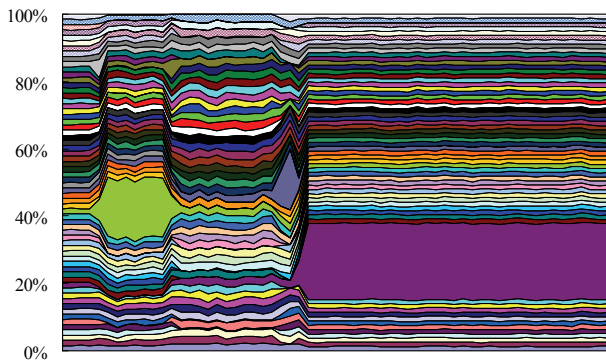


図 10 : mgrid における整数レジスタアクセス

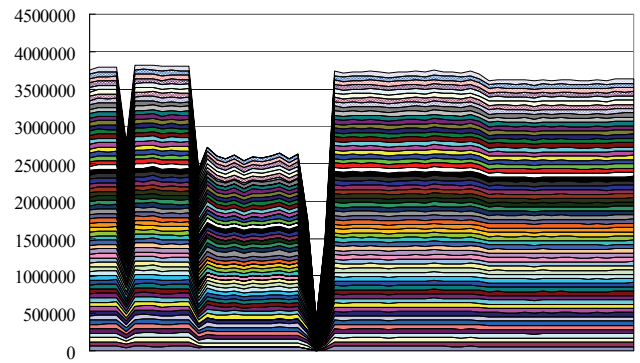


図 13 : mgrid における FP レジスタアクセス回数

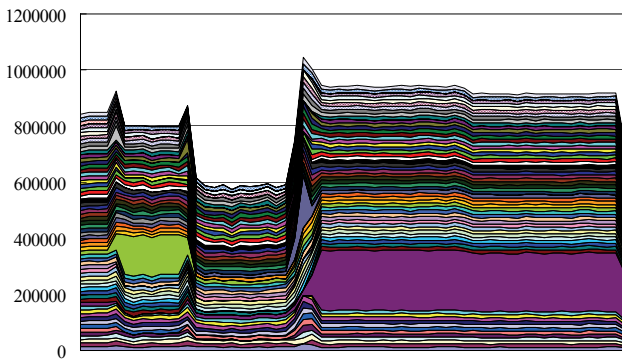


図 11 : mgrid における整数レジスタアクセス回数

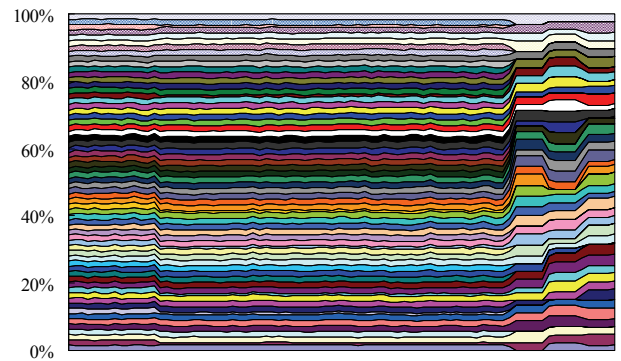


図 14 : lucas における整数レジスタアクセス

図 12 と図 13 は浮動小数点レジスタの結果である。整数レジスタでアクセス集中が観察されない区間で、使用されないレジスタが存在している。ただし、その区間ではアクセス総数も減少している。

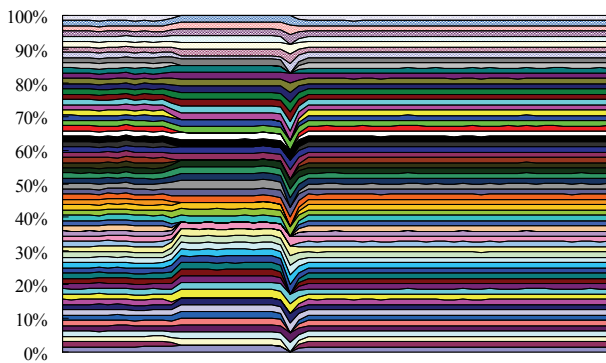


図 12 : mgrid における FP レジスタアクセス

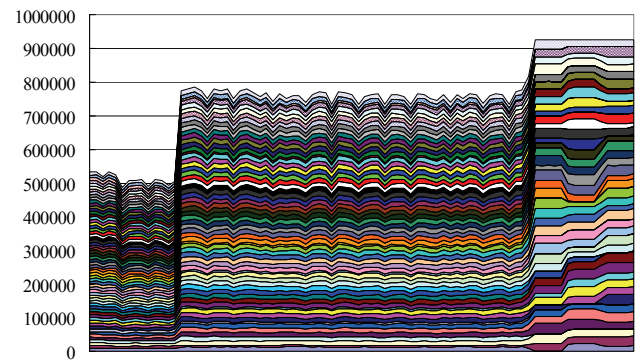


図 15 : lucas における整数レジスタアクセス回数

図 14 と図 15 は 189.lucas における整数レジスタの結果である。図の右端の区間で、特定の複数のレジスタへのアクセスが無くなっていることが観察される。興味深いのは、この区間でアクセス総数が増えている点である。つまり、特定のレジスタへのアクセス回数が増加しているわけである。これは熱設計の観点から好

ましくない。ただし、lucas においてはアクセス総数の絶対値が小さいため、ホットスポットが生じる可能性は高いが、熱設計が困難になるかどうかは詳細な調査を要する。

以上をまとめる。[9]での調査では不均一なレジスタアクセスは確認されなかったが、時間を区切るとアクセスの集中するレジスタや全くアクセスされないレジスタが存在すること観察された。つまり[9]の結果とは異なり、レジスタアクセスには時間的な不均一性が存在することが確認できた。

最後に、これらの不均一アクセスを解消する方法を

検討する.

アクセス集中の支配的な要因はリードアクセスである. これを緩和するには, レジスタ間でのマイグレーションが有効である. 離れた位置にあるレジスタ間でのマイグレーション[1, 6, 7]ではなく, 同じレジスタファイル内のレジスタ間でのマイグレーションである. レジスタアクセス回数を観察すれば, プログラムから透過的にマイグレーションするマイクロアーキテクチャを容易に実現できると思われる. しかし, 命令で容易にレジスタ間でデータ転送できる. コンパイラが多大なリードアクセスを発見し, 適切な場所にレジスタ間転送命令を挿入できることが望ましい. なぜなら, 追加されるハードウェアによる副次的な消費電力増などを抑制できるからである.

熱設計の観点からは, 利用されないレジスタが存在するよりも, 出来るだけ多くのレジスタが均等に利用される方が望ましい. あるレジスタが長期に渡って参照されないことを検出し, その再利用を促すようなマイクロアーキテクチャの考案は容易に思える. しかし, スピルコードを挿入するだけで, 容易にレジスタを再利用できる. コンパイラは該当するレジスタ内の値を長期間保持しなければならない理由を知っているはずであるから, スピルコードの挿入は容易と考えられる. 参照されないレジスタ内の値を積極的にメモリに待避すれば, ハードウェアを追加することなくレジスタの再利用を促すことが可能になる.

加えて浮動小数点系プログラムでは, 図 8~15 からわかるように, 整数系プログラムよりも特徴的なレジスタアクセスのパターンが認められる. これらの特徴を有効に利用できれば, コンパイラによる分析が容易になると期待できる.

5. まとめ

近年問題として認識されつつある発熱に着目し, 特定のレジスタやレジスタバンクへの集中的なアクセスは著しい発熱を伴うことを考慮して, 物理レジスタのアクセス頻度を調査した. 以前の調査では, 物理レジスタへのアクセスは均等に行われていることが観察されていた. しかし時間的な変動を考慮していなかったため, 今回は時間を区切ってアクセス頻度を調査した. その結果, レジスタアクセスには時間的な変動があり, 特定のレジスタへのアクセス集中が生じる区間が存在することが確認された. また, 全く参照されないレジスタが存在する場合も確認された.

今後は, まず今回報告できなかったマルチバンク化の影響を調査したい. 続いて, レジスタアクセスを時間的にも空間的にも均一にできるマイクロアーキテクチャやコンパイラ技術を検討したい. 例えば, リネー

ミングレジスタの整列手法[14]を利用できるかも知れない.

謝辞

本研究の一部は, 科学研究費補助金 (No.16300019, No.176549) の援助によるものです.

文 献

- [1] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture", 30th International Symposium on Computer Architecture, 2003.
- [2] V. Zyuban, P. Kogge, "The energy complexity of register files", International Symposium on Low Power Electronics and Design, 1998.
- [3] J. Abella and A. Gonzalez, "On reducing register pressure and energy in multiple-banked register files", 21st International Conference on Computer Design, 2003.
- [4] J. Abella and A. Gonzalez, "Power-aware adaptive issue queue and register file", International Conference on High Performance Computing, 2003.
- [5] J. L. Ayala, M. Lopez-Vallejo, A. Veidenbaum, "Energy-efficient register renaming in high-performance processors", Workshop on Application Specific Processors, 2003.
- [6] M. Goto and T. Sato, "Leakage energy reduction in register renaming", 1st International Workshop on Embedded Computing Systems, 2004.
- [7] S. Heo, K. Barr, and K. Asanovic, "Reducing Power Density through Activity Migration", International Symposium on Low Power Electronics and Design, 2003.
- [8] J. Tseng and K. Asanovic, "Banked multiported register files for high-frequency superscalar microprocessors", 30th International Symposium on Computer Architecture, 2003.
- [9] R. P. Preston, R. W. Badeau, D. W. Bailey, S. L. Bell, L. L. Biro, W. J. Bowhill, D. E. Dever, S. Felix, R. Gammack, V. Germini, M. K. Gowan, P. Gronowski, D. B. Jackson, S. Mehta, S. V. Morton, J. D. Pickholtz, M. H. Reilly, M. J. Smith, "Design of an 8-wide superscalar RISC microprocessor with simultaneous multithreading," International Solid-State Circuits Conference, 2002.
- [10] 国武, 千代延, 佐藤, "スーパースカラプロセッサにおけるレジスタアクセスの特徴調査", 第 13 回信学会九州支部学生会講演会, 2005.
- [11] R. Desikan, D. C. Burger, and S. W. Keckler, "Measuring experimental error in microprocessor simulation", 28th International Symposium on Computer Architecture, 2001.
- [12] R. E. Kessler, "The Alpha 21264 microprocessor", IEEE Micro, Vol.19, No.2, 1999.
- [13] G. Hamerly, E. Perelman, J. Lau, and B. Calder, "SimPoint 3.0: faster and more flexible program analysis", Workshop on Modeling, Benchmarking and Simulation, 2005.
- [14] 小林, 堀部, 島田, "リネーミングされるレジスタ番号の整列によるレジスタ・キャッシュの高精度化手法", 情処研報 2006-ARC-167, 2006.