

Instruction Caches in Instruction/Thread-Level-Parallelized Microprocessors(命令・スレッドレベル並列性を考 慮したマイクロプロセッサの命令キャッシュに関する研究)

著者	多田 十兵衛
号	248
発行年	2002
URL	http://hdl.handle.net/10097/12944

氏名（本籍）	ただ じゅうべえ 多田 十兵衛	（岩手県）
学位の種類	博士（情報科学）	
学位記番号	情博第248号	
学位授与年月日	平成15年3月24日	
学位授与の要件	学位規則第4条第1項該当	
研究科，専攻	東北大学情報科学研究科（博士課程）情報基礎科学専攻	
学位論文題目	Instruction Caches in Instruction/Thread-Level-Parallelized Microprocessors（命令・スレッドレベル並列性を考慮したマイクロプロセッサの命令キャッシュに関する研究）	
論文審査委員	（主査）	
	東北大学教授 中村 維男	東北大学教授 亀山 充隆
	東北大学教授 小林 広明	東北大学教授 青木 孝文

論文内容要旨

1 結論

近年、コンピュータの需要の広がりや留まる事を知らず、さまざまな分野への応用が行われている。コンピュータの性能への要求は益々増加しており、コンピュータの高性能化に関するさまざまな研究が行われている。高性能化の手法の一つとして、プログラム中の並列性を利用するものが挙げられる。

VLIW プロセッサは、プログラム中の命令レベル並列性を静的に抽出して利用するプロセッサである。マイクロプロセッサにおいて、命令を高速にフェッチするためには、高性能な命令キャッシュが必要不可欠であり、これは VLIW 命令を実行する VLIW プロセッサにおいても同様である。1つの VLIW 命令は複数の演算から構成されるため、非常に語長が長い。従って、VLIW 命令を高速にフェッチするためには、広いメモリバンド幅が必要となる。一方、すべての VLIW 命令を常に演算で満たすことは困難であるため、VLIW 命令中には NOP が挿入される。NOP を含む VLIW 命令をそのまま命令キャッシュに格納すると、命令キャッシュ中に格納可能な有効演算数が減少する。これにより、キャッシュヒット率が低下するという問題が発生する。従って、VLIW プロセッサには、これらの問題を解決する命令キャッシュが必要となる。

SMT プロセッサは、複数のスレッドからスレッドレベル並列性を抽出することで、命令レベル並列性のみを利用するプロセッサよりも高い機能ユニットの稼働率を得ることが可能なプロセッサである。しかし、SMT プロセッサでは、複数のスレッドがプロセッサ資源を共有するため、資源競合が大きな問題となる。命令キャッシュは特に負荷の大きな資源であり、同時実行するスレッド数の増加により性能が大きく低下することが知られている。従って、SMT プロセッサでは、より効率的に利用可能な命令キャッシュが必要となる。

本研究では、VLIW プロセッサおよび SMT プロセッサのための命令キャッシュの提案およびその評価を行なう。

2 VLIW プロセッサのための従来の命令キャッシュ

本章では、VLIW プロセッサにおいて命令キャッシュの性能に影響を与えるパラメータについて議論し、それらを用いて従来 VLIW プロセッサのために用いられてきた命令キャッシュの問題点を明らかにする。VLIW プロセッサでは、同時に実行可能な複数の演算をコンパイラが抽出し、それらを一つの VLIW 命令として実行することで命令レベル並列性を利用する。ここで、VLIW 命令は複数の演算から構成されるため、語長が長いという特徴がある。このため高速な命令フェッチが困難となる。また、VLIW 命令中の使用されない機能ユニットに対応した部分には、演算を実行しないという無効な演算である NOP が挿入される。この NOP を命令キャッシュ中に格納すると、キャッシュ中に格納可能な有効な命令数が減少し、キャッシュヒット率が低下するという問題が発生する。

ここで VLIW プロセッサのための従来の命令キャッシュについて述べる。NOP キャッシュは、VLIW 命令を機能ユニットに直製発行可能な NOP を格納した形式で格納するキャッシュである。NOP を格納しているため機能ユニットへの命令の発行は容易となるが、高速な命令フェッチには広いバス幅が必要となる。また、NOP の格納によりキャッシュの使用効率が低下する。

COMPRESS キャッシュは、VLIW 命令を NOP を除いた形式で格納するキャッシュである。NOP を格納しないため、キャッシュの使用効率は高くなる。しかし、複数のキャッシュラインに命令がまたがる可能性があり、この場合命令フェッチに必要なサイクル数が増加するという問題がある。

SILO キャッシュは、機能ユニット毎にその機能ユニットに発行される命令を格納するサイロと呼ばれるそれぞれ独立した命令キャッシュを用意したキャッシュである。VLIW 命令中の演算は、その演算が使用する機能ユニットに対応したサイロに格納される。SILO キャッシュは、キャッシュ中に NOP を格納しないため高い使用効率が得られ、また各サイロから読み出されるのは VLIW 命令中の演算であるため、高速な命令フェッチに必要なバス幅が狭くすむという特徴がある。しかし、特定の機能ユニットを利用する命令が集中する場合、その機能ユニットに対応したサイロのみが使用されるという状況が発生し、キャッシュの使用効率が低下する。

次章では、これら従来の命令キャッシュの問題点を解決するために提案された MULHI キャッシュの設計および評価を行なう。

3 VLIW プロセッサのための命令キャッシュ

本章では、VLIW プロセッサのための命令キャッシュとして提案された MULHI (MULTiple-Hit) キャッシュの設計および評価を行なう。

MULHI キャッシュは、VLIW 命令を演算単位で分割し、分割された演算をセットアソシアティブキャッシュの各セットの異なるウェイにそれぞれ格納する。ここで、同一セット上で、1つの VLIW 命令の格納後にも空きウェイがある場合には、異なる VLIW 命令を格納可能である。1つの VLIW 命令に含まれる演算は同じタグを持って格納されるため、命令フェッチの際には、要求された VLIW 命令を格納している複数のウェイでヒットが起これ、それらのウェイから出力された演算が、それぞれ適切な機能ユニットに発行される。MULHI キャッシュでは、複数の VLIW 命令を柔軟に同一セットに格納することが可能であり、SILO キャッシュに見られる特定のサイロへの集中格納という状況を回避することができる。その結果、命令キャッシュ中に格納可能な VLIW 命令が増加し、高いヒット率が期待できる。

しかし、MULHI キャッシュでは、複数のウェイに同時に書き込みを行なうため、1つのウェイのみに書き込みを行なう従来のキャッシュよりもウェイを選択する機構が複雑になる。このハードウェアの複雑さがサイクル時間に与える影響を評価するため、CAD を用いて設計を行い、その動作時間の測定

を行った。その結果、ハードウェアの複雑さは MULHI キャッシュの性能に影響を与えないことが示された。

次にソフトウェアシミュレーションによる OPC の評価を行なった。その結果、全ベンチマークの平均 OPC では、全ての構成において、MULHI キャッシュが他のキャッシュ機構を上回った。また、MULHI キャッシュは、特にキャッシュサイズの小さい構成において、SILO キャッシュと比較して高い性能が得られることがわかった。しかし、コードサイズの大きいベンチマークにおいては、COMRESS キャッシュよりも OPC が低下した。最も性能が向上した構成では、NOP キャッシュ、COMPRESS キャッシュ、SILO キャッシュと比較してそれぞれ平均 120%、4.6%、13%の性能向上が得られた。

4 SMT プロセッサのための命令キャッシュ

本章では、スレッドレベル並列性を利用する SMT プロセッサのための命令キャッシュの提案およびその評価を行なう。SMT プロセッサでは、複数のスレッドからの同時アクセスに対応するため、マルチポートの命令キャッシュが必要となる。

SMT プロセッサのための命令キャッシュとして、I-MULHI (Interleaved-MULHI) キャッシュを提案する。I-MULHI キャッシュは MULHI キャッシュを 1 つのバンクとして用い、それらをインターリーブ化することでマルチポートを実現した命令キャッシュである。I-MULHI キャッシュは、複数のスレッドの命令をキャッシュ中に効果的に格納することが可能であるため、高いヒット率が期待できる。

しかし、マルチバンク化マルチポートキャッシュにおいては、バンク競合の発生により使用可能なポート数が減少するという問題が発生する。そこで、バンク競合を起こさないスレッドの組み合わせをポートに割り当てることでバンク競合を回避するスレッドスケジューリング手法の提案を行なった。

シミュレーションによる性能評価の結果、I-MULHI キャッシュは、従来の命令キャッシュと比較して最大で 187%の性能向上が得られた。また、提案するスレッドスケジューリングを行なった I-MULHI キャッシュは、行なわない場合と比較して最大で 75%の性能向上が得られた。

5 結論

本研究では、命令レベル並列性を利用する VLIW プロセッサのための命令キャッシュである MULHI キャッシュの設計および評価、そして SMT プロセッサのための命令キャッシュである I-MULHI キャッシュの提案および評価を行った。

ハードウェア設計の結果、MULHI キャッシュのハードウェアの複雑さは性能に影響を与えないことが示された。シミュレーションによる評価の結果、MULHI キャッシュは VLIW プロセッサのための従来の命令キャッシュと比較して高い性能を得られることが示された。また、シミュレーションによる評価の結果、I-MULHI キャッシュは、従来の SMT プロセッサのための命令キャッシュよりも高い性能を得られることが示された。さらに、I-MULHI キャッシュは、バンク競合を回避するスケジューリングにより、性能がより向上することが示された。

論文審査の結果の要旨

近年プロセッサ-メモリ間の速度差は拡大傾向にあり、これを隠蔽する技術であるキャッシュの高性能化が重要な課題となっている。特に、プログラム中の並列性を利用するプロセッサでは、命令キャッシュの性能低下が大きな問題となる。本論文は、この問題を解決する手法として、命令レベル並列性とスレッドレベル並列性をそれぞれ利用するVLIW (Very Long Instruction Word)およびSMT (Simultaneous Multithreading)方式のプロセッサのための命令キャッシュを提案しその評価を行ったもので、全編5章からなる。

第1章は緒論である。

第2章では、命令レベル並列性を利用するVLIWプロセッサにおける命令キャッシュの問題点を論じている。特に、VLIWプロセッサにおいては命令フェッチサイクル数の増加、および使用効率の低下が問題となることを示している。これは、VLIWプロセッサの命令キャッシュを設計する際の重要な指針を与えるもので、実用上有益な指摘である。

第3章では、前章で明らかにした問題点を解決するVLIWプロセッサのためのキャッシュとして提案されているMULHI (MULtiple-HIt)キャッシュについて論じている。MULHIキャッシュの制御機構を提案してそのハードウェア設計を行い、MULHIキャッシュの制御の複雑さが性能に影響を与えないことを示している。また、シミュレーションを行い、MULHIキャッシュが高い性能を達成できることを示している。これは実用上有効な成果である。

第4章では、スレッドレベル並列性を利用するSMTプロセッサのための命令キャッシュとしてI-MULHI (Interleaved-MULHI)キャッシュを提案し、その評価を行っている。マルチバンク化マルチポートキャッシュにおいては、バンク競合がその性能に大きな影響を与えることを示し、さらに、バンク競合を回避するスケジューリング手法を提案している。また、シミュレーションにより、I-MULHIキャッシュが高い性能を得られることを示している。本成果は、スレッドレベル並列性を利用するプロセッサの命令キャッシュを設計する際に役立つものであり、非常に重要な成果である。

第5章は結論である。

以上要するに本論文は、命令およびスレッドレベル並列性を利用するプロセッサにおいて、それらの性能を高める命令キャッシュを提案し、その設計をする上での有力な指針を与えたもので、情報科学並びに計算機科学の発展に寄与するところが少なくない。よって、本論文は博士(情報科学)の学位論文として合格と認める。