ストアキューの拡張によるロードトラフィックの削減方式

吉瀬謙二†，斎藤英一‡，入江英嗣†，坂井修一†，田中英彦†
東京大学大学院工学系研究科†　株式会社日立製作所 日立工業専門学院‡

1　はじめに

アウトオブオーダ実行をおこなうプロセッサは、ストア命令をインオーダに処理するためにストアキューを利用する。このストアキューのエントリを数百という規模に拡大するとともに、命令がリタイアされた後もエントリに余裕がある限りデータを保持することで、多くのロード命令の要求をストアキューで処理することを検討する。この拡張により、今後予測されるデータ供給機構のポート数増加の問題緩和を目指す。

2　データキャッシュのポート数

データキャッシュのマルチ・ポート化には幾つかの戦略がある [1]。IBM Power2, DEC Alpha21264 ではキャッシュをプロセッサの倍速で動作させることで 2 ポートを実現する。MIPS R10000 ではキャッシュを 2 バンクに分割することで、異なったバンクへのアクセスを同時に可能としている。DEC Alpha21164 ではキャッシュを 2 つに複製し、一貫性を維持することで 2 ポートを実現している。このような中、更なる命令レベル並列性の利用を目指した研究が進められており [3]、それに見合うポート数の確保が課題となっている。

3　ストアキューとデータ参照の時間的局所性

アウトオブオーダ実行をおこなうプロセッサは、キャッシュの更新をインオーダに処理するためにストアキューを利用する。このような目的でストアキューを実装した場合には、キャッシュに値を書きこんだエントリはストアキューから開放してかまわない。一方、キャッシュにデータを書きこんだ後にもエントリを開放せず、データをストアキューから供給するように設計することもできる。この場合、新しくエントリが必要となった時点で、最も古く割り当てられたエントリを開放する。エントリが開放されるまで、ストアキューは各エントリのデータを供給できるバッファとなる。本研究ではストアキューを両方の目的で利用する。

ストアキューのエントリ数を 16〜512 に変化させて、ストアキューから供給できるデータの割合を測定した。スカラプロセッサ環境を用いて測定したため、n エントリのストアキューからデータを供給できる割合は、ロードするデータが、先行する n 個のストア命令において生成された値である割合となる。測定結果を図 1に示す。この結果から、先行する 256 個のストアからロードできる確率は平均で 51%となることがわかる。

Load Traffic Decrease with Extended Store Queue
Kenji KISE, Eiichi SAITO, Hidetsugu IRIE,
Shuichi SAKAI, and Hidehiko TANAKA
† Graduate school of Engineering, The University of Tokyo
‡ Hitachi, Ltd.

本評価で採用した Alpha AXP アーキテクチャの 1 ワードは 8byte であり、256 個のストア・データを保存するためには容量 2KB の連想記憶が必要となる。

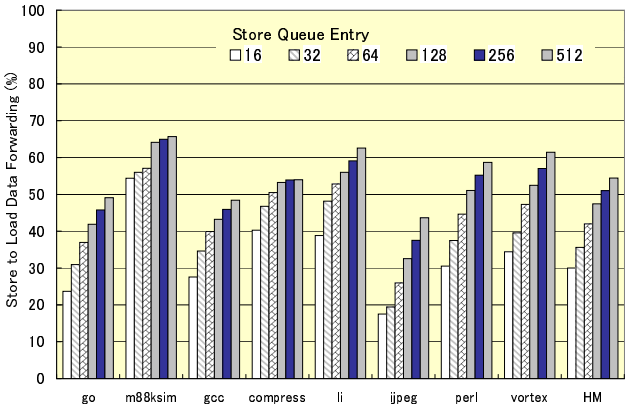

図 1: ストアキューのエントリ数と供給できるデータ

4　ストアキューの拡張によるロードトラフィックの削減

図 1の予備評価結果より、ストアキューのエントリ数を大きくすることで、約半分のデータをストアキューから供給できる可能性があることがわかった。しかし、ストアキューとキャッシュの両方にロードの要求を出していては、ロードユニットとキャッシュとのトラフィックを削減することはできない。このため、図 2に示す様に、ロード命令をストアキューからデータを得るものと、キャッシュからデータを得るものに分類し、それぞれのバッファにのみロードの要求を出す方式を検討する。

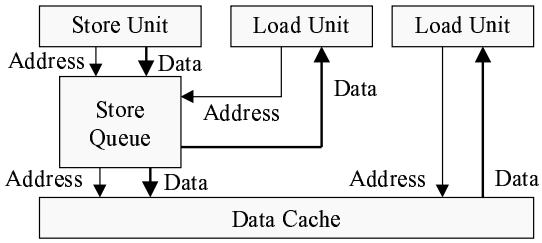

図 2: ストアキューの拡張とトラフィックの分散

利用するバッファの選択方法として、動的予測や静的情報の利用が考えられる。図 2の機構により、ロード・トラフィックを 2 つのバッファに分散させ、ポート数増加の効果を得るには、適切なバッファにロード要求を出す、それぞれのバッファに均等にロード要求を出す、という 2 つの条件を満たす必要がある。最初の条件を満たせない場合には、ストアキューにデータを要求したが、そのデータをストアキューからは供給できないといった状況

が発生する。この場合には、次のサイクルにキャッシュへのアクセスが必要となり、ロード命令の実行レイテンシを 1 サイクル増加させることになる。2 番目の条件を満たせない場合には、キャッシュにアクセスするロード命令が集中して実行されるといった状況が発生し、バッファの利用率が低下する。

5 ストアキュー利用の動的予測

利用するバッファを分類するために動的予測の利用を検討する。コンパイラ等を用いた静的な情報付加は今後の課題である。動的な予測機構として 2 ビット ・カウンタ方式 (4096 エントリのダイレクトマップ), JRS 予測器 [2] (4096 エントリの予測テーブル、閾値 8) を評価した。ストアキューのエントリは 256 エントリを想定した。

Program	2BC	JRS
go	37.54%, 2.84%	6.25%, 0.88%
m88ksim	63.84%, 0.43%	59.24%, 0.31%
gcc	41.39%, 1.77%	20.52%, 1.04%
compress	53.06%, 0.79%	47.73%, 0.28%
li	55.16%, 1.40%	48.33%, 0.45%
ijpeg	33.64%, 2.32%	18.08%, 0.89%
perl	50.83%, 1.77%	43.79%, 0.35%
vortex	51.25%, 2.65%	35.82%, 0.95%

表 1: ストアキュー利用の動的予測結果

表 1に結果を示す。2 ビット ・カウンタ方式 (2BC), JRS 予測器 (JRS) それぞれについて、ストアキューからデータを供給できると正しく予測した割合と、ストアキューからデータを供給できると予測してミスした割合を示した。割合は、実行した全ロード命令に対するものである。この結果から、分岐履歴を用いて予測する JRS の利点はそれほど大きくないことがわかる。特に go, ijpeg については、JRS で予測できる割合が大きく低下する。

6 ストアキューの拡張と性能評価

スーパースカラのシミュレータを用いて、拡張したストアキューを評価する。本評価では、ミスを起こさないデータキャッシュを仮定した。また、メモリシステムに厳しい負荷を与えるために、分岐ミスをおこさない理想的な分岐予測機構を仮定した。シミュレータの主な実行パラメタを表 2にまとめる。

Parameter	Value
Bandwidth of fetch and retire	16 instr. per cycle
Instruction window	128 entries
Memory latency	2 cycle / 3 cycle
The number of functional unit	No limitation

表 2: 評価に用いたプロセッサ ・モデルのパラメタ

キャッシュのポート数 1 とストアキューのポート数 2(設定 a)、キャッシュのポート数 2 とストアキューのポート数 2(設定 b)、ポートの数を 1(設定 c), 2(設定 d), 4(設定 e), 無限大 (設定 f)、という 6 つの設定においてプロセッサ性能を測定した。設定 a, b では、拡張されたストアキューを用いている。ストアキューの利用については 5章で検討した 2BC を用いた。設定 c から f は、両方のバッファにリクエストを送る通常のプロセッサ ・モデルであり、比較のために用意したモデルである。

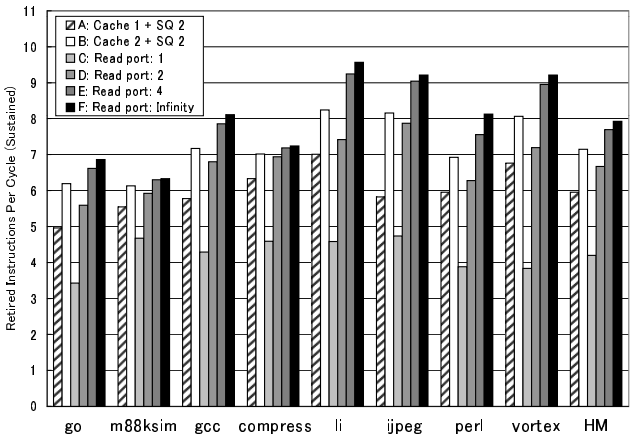

図 3: 拡張したストアキューによる性能向上

評価結果を図 3に示す。キャッシュのポート数 1 を想定した場合の評価は、設定 a と c の結果を比較すればよい。この時、拡張されたストアキューを用いることで IPC1.7 の向上を確認できた。同様に、キャッシュのポート数 2 を想定した場合、設定 b と d の結果を比較すればよい。この時、拡張されたストアキューを用いることで IPC0.5 の向上を確認できた。

7 まとめ

ストアキューをデータ参照の時間的局所性を利用するバッファとして利用するために、ストアキューのエントリを 256 という大きな値に設定するとともに、ロード要求をキャッシュとストアキューに分散させ、キャッシュとロードユニットのトラフィックを削減する方式を検討した。

ストアキュー利用の動的予測に関して、2 ビット ・カウンタ方式と JRS 予測器を評価し、前者が適していることを示した。スーパースカラのシミュレータを用いて本方式を評価し、特にキャッシュのポート数が少ない場合に、本方式が有効であることを確認した。

参考文献

[1] Jude A. Rivers, Gary S. Tyson, Edward S. Davidson, and Todd M. Austin. On High-Bandwidth Data Cache Design for Multi-Issue Processors. In Proc. of MICRO-30, pages 46-56, 1997.

[2] Erik Jacobsen, Eric Rotenberg, and J. E. Smith. Assigning confidence to conditional branch predictions. In Proc. of MICRO-29, pages 142-152, 1996.

[3] 中村友洋, 吉瀬謙二, 辻秀典, 安島雄一郎, 田中英彦. 大規模データパスプロセッサの構想. 情報処理学会計算機アーキテクチャ研究会, Arch 124, pp.13-18, Jun.1997.