

# 複数パス実行プロセッサのための キャッシュメモリの提案

高野 直樹 田中 英彦 坂井 修一

{takano, tanaka, sakai}@mtl.t.u-tokyo.ac.jp

東京大学大学院 工学系研究科 \*

## 1 はじめに

プロセッサの処理性能向上に対し、主記憶に使われている DRAM の処理速度が追い付かなくなり、プロセッサにデータ供給が十分に行われなくなってきた。このプロセッサとメモリーとのパフォーマンスギャップを解決するために、プロセッサと主記憶の間にキャッシュメモリが用いられてきた。

今後プロセッサの 1cycle に実行された命令数 (IPC<sup>1</sup>) が向上するに従って、今まで以上に十分なデータを供給できるキャッシュの研究が不可欠である。本稿では次世代プロセッサとして現在研究が進められている複数パス実行プロセッサのためのキャッシュメモリについてポート数に着目し提案を行う。

## 2 複数パス実行プロセッサ

現在使われている Super Scalar プロセッサでは、分岐予測を行い単一パスの投機的実行を行っているが、3~4 IPC が性能の限界と言われている。更に IPC の向上を計るには、分岐予測ミス時のペナルティーを削減を行うために、分岐成立と不成立の投機的な両パスの実行を行う必要がある。これが複数パス実行プロセッサである。

図 1 に単一パス実行と複数パス実行の違いを示す。図中の単一パス実行では分岐予測失敗時にペナルティーが発生しているが、複数パス実行では分岐成立と不成立の両パスを実行するのでペナルティーは存在しない。また各パスは投機的に実行されるので IPC は単一パス実行よりも向上することがわかる。

## 3 複数パス実行プロセッサにおけるキャッシュのポート数の考察

キャッシュが 1cycle で同時にデータの読み書きが行える数をポート数と呼ぶ。ここでは複数パス実行プロセッサにおけるキャッシュメモリをポート数の観点から考える。

### 3.1 ポート数増加の必要性

複数パス実行は単一パス実行と比較して、無駄なパスの命令も投機的に実行されるので、実行される命令数は

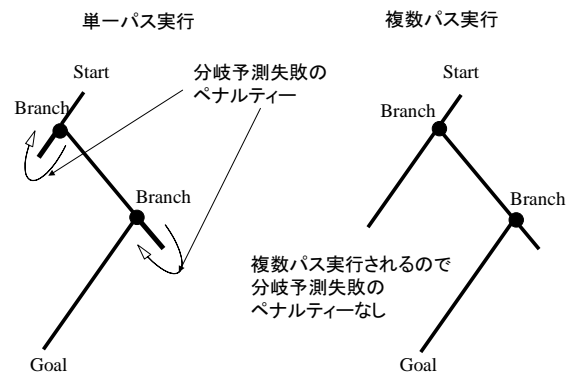


図 1: 単一パス実行と複数パス実行の違い

多くなる。当然 1cycle あたりの Load/Store の数もマルチパス実行によって増えることが考えられる。

キャッシュへのアクセスを考えると Store は実行のパスが確定しない限り、すなわち Store 命令が In order 状態にならない限りキャッシュへの書き込みは起らない。Store がキャッシュへ書き込みを行うタイミングはキャッシュの上流に存在する Load/Store キューで吸収するので、キャッシュでそれを考える必要はない。プログラムはそれぞれ一定の割合で In Order になる Store が含まれているので、同じプログラムを実行すれば、実効 IPC に比例したキャッシュへの Store による書き込みが起ると考えられる。

一方 Load は投機的に実行されたパスからも要求されるので、Load/Store キュー内にそのデータが存在すれば良いが、見つからないとキャッシュへ読み込みの要求が送られる。Load/Store キューの大きさや複数パス実行のアルゴリズムによるが、実行 IPC に比例した数以上にキャッシュへの読み込みの要求が増える可能性がある。つまり従来の Super Scalar プロセッサよりも 1cycle で同時にキャッシュへの読み込み要求が増えることが考えられる。

以上のように複数パス実行プロセッサにおいては Load/Store とともに従来の Super Scalar で用いられてきたキャッシュよりもポート数を増やす必要性が考えられる。

### 3.2 従来のポート数の増加手法

SRAM を理想的に多ポート化するとポート数の自乗分の実装コストがかかる。そのため従来のキャッシュの多ポート化には 3 つの手法が取られてきた。1 つは時分割法、1 つは複製法、そして Bank 分割法である。

\* Naoki Takano, Hidehiko Tanaka and Shuichi Sakai  
University of Tokyo, Graduate School of Engineering,  
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

<sup>1</sup>Instrucion Per Cycle の略

時分割法はキャッシュに使われている SRAM のクロックをプロセッサの複数倍率で駆動することによりポート数を増やす手法である。しかしこの方法で増やせるのはせいぜい 2~3 ポートが限界であり、スケーラビリティに欠けるので、多ポート化には向かない。

複製法は同じ内容のキャッシュをポート数だけ用意し、読み込みのポート数を増やす手法である。これは実装コストがポート数倍かかるのと、書き込みの際一貫性を保つため書き込みのポート数は 1 のままであるのが欠点である。

Bank 分割法は SRAM を複数 Bank に分割し、アドレスによって分散配置を行うことでポート数を増やす方法である。しかしこの方法は Bank 化されたキャッシュと Load/Store ユニットの接続するため Crossbar 等の相互結合網が必要であり、同じ Bank への衝突が起る可能性がある。

現在の半導体の集積度の問題から考えると、複製法よりも Bank 分割法の方がトランジスタを有効に利用できるため、利点が大きいように思われる。しかし今後 10 年間の半導体テクノロジー予測 [2] を参照すると、将来キャッシュに使えるトランジスタ数は数億個オーダーになり、Bank 分割が必ずしも有利な手法とは言えなくなる。

### 3.3 複製法のキャッシュを使った Lazy Store

ここで、複製法のキャッシュに対して新しく Store 専用の Buffer を設けることで、遅延書き込みを行えるようにしたキャッシュを新たに提案する。このキャッシュを図 2 に示す。

Bank 分割法より複製法を選んだのは Bank 衝突を起こさず効率良く Load 要求に答えるためである。過去の研究 [1] においては Bank 分割法をベースにした手法が提案されていたが、この手法では Load/Store とともに Latency が増加する可能性があり、異なったラインの Bank 衝突が起るのは回避できない。また複数パス実行プロセッサにおいては、Super Scalar プロセッサと比較して Store より Load の方が数が増えることを想定して、Load と親和性が高い複製法を選択する。

本手法は複製法におけるキャッシュへの書き込みは 1 ポートしか行えない欠点を補うため、Store の際起る書き込みを Lazy に行うことで、Store の書き込みのポート数を仮想的に増やす。

Store データはキャッシュに書き込みの要求を行った後でもしばらくの間 Load/Store キューに留まっており、その間はキャッシュにアクセスする必要はない。そこでポートが塞がっている時は、Store 要求が出されても一度にキャッシュに書き込みを行わずに、一時的に Lazy Store Buffer に保存する。キャッシュは常に全てのポートを占有されている訳ではなく、一時的に Load/Store ユニットからの要求が来ない時があるので、空いた時間を使って Lazy Store Buffer にある古いデータから Store を行う。Store の実行後そのキャッシュの Stored Bit をセットし、どのキャッシュにすでに書き込まれたかを保

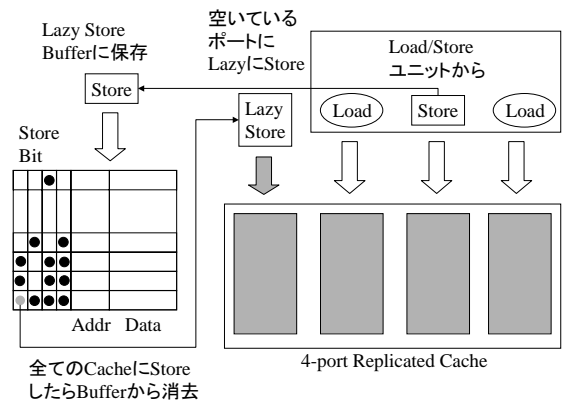


図 2: 複製法をベースにした新しい多ポートキャッシュ

存する。Stored Bit 全てがセットされたらそのエントリは Lazy Store Buffer から消去する。Load/Store キューから Store データのエントリが削除される時にまだ Lazy Store Buffer に残っている時は、強制的にキャッシュに対して書き込みを行うことで一貫性を保つ。

本手法は Load/Store キューからデータが追い出されるまでの時間を使うので、Load/Store キューのエントリ数が小さい時には Lazy に Store を行う時間が少なくなってしまい効果は低下する。このため Load/Store キューのエントリ数を考えた上での定量的評価が重要である。

## 4 まとめ

次世代プロセッサとして研究されている複数パス実行プロセッサの概要を述べ、キャッシュシステムの多ポート化の重要性を述べた。その後、複製法を用いたキャッシュに対する Lazy Store の提案を行った。今後は Super Scalar プロセッサと複数パス実行プロセッサのシミュレータを用いて本手法の定量的な評価を行う予定である。

本研究の一部は、文部省科学研究費補助金 (基盤研究 (B) 課題番号 11480066) および、(株)半導体理工学研究センターとの共同研究によるものである。

## 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," *International Symposium on Microarchitecture*, Pages 46-56, December 1-3, 1997, Research Triangle Park United States.
- [2] Semiconductor Industry Association, *International Technology Roadmap for Semiconductors 1999 Edition*. <http://www.itrs.net/ntrs/publntrs.nsf>