

分岐先バッファにおける分岐先情報の削減

1L-2

原 哲也 安藤 秀樹 中西 知嘉子 中屋 雅夫
三菱電機株式会社 システムLSI開発研究所

1.はじめに

パイプライン方式のプロセッサにおいて、分岐によるパイプラインの乱れは性能向上を阻む大きな原因となっている。この分岐ペナルティを低減させるために通常のパイプライン・プロセッサでは分岐予測を導入している。静的に予測を行う方式ではコンパイラへの負担が大きく、また、静的な予測結果を命令を介してプロセッサに伝えるためオブジェクト・コードを変える必要がある。これに対して、分岐先バッファを用いた分岐予測方式は、命令の実行時に動的に予測を行うので、オブジェクト・コードの互換性を保つことができるという利点がある。しかしその欠点は、予測に必要な情報を保持するための分岐先バッファ(BTB:Branch Target Buffer)と呼ばれるハードウェアが必要であることである。

本稿では、従来BTB方式の問題点であるハードウェア量を削減するための方式を提案し、その構成、動作、および、性能について述べる。

2.従来のBTB方式^[1]

従来のBTBの1エントリは命令キャッシュとタグを共有する場合、エントリの有効ビット(1bit)、タグ(数bit)、履歴情報(1-2bit)、分岐先情報(分岐先アドレス:30bit)で構成される。このBTBを用いて、以下のような手順で分岐予測を行う。

①登録・更新:分岐命令が実行されると分岐先アドレス、分岐方向などの実行結果をその分岐命令の命令アドレスで指定されるBTBエントリに登録・更新する。

②予測:命令フェッチ・ステージにおいて、命令キャッシュをアクセスすると同時に、同じアドレスでBTBのアクセスを行う。分岐命令が登録されていればその履歴情報に基づいて分岐方向の予測を行い、taken予測であれば分岐先アドレスをPCに設定する(図1-a)。

③予測のチェック:分岐命令が実行されると、その実行と分岐予測を照らし合わせて予測のチェックを行う。予測が間違っていた場合、誤って投入された命令を無効化すると共に、正しい命令流を再フェッチする。

BTBを用いた分岐予測方式では、BTBに登録されている分岐命令を対象に予測を行うので、高い予測ヒット率を得るためには、BTBのエントリがヒットすることが条件であり、BTBには多くのエントリ数が必要となる。一般に、80%程度の予測ヒット率を得るには256-1KのBTBエントリが要求される^{[2][3]}。

BTBの1エントリは30数bitと大きいので、512エン

トリでは8KB命令キャッシュの約40%の大きさとなる(表1)。このように、BTB方式では、予測ヒット率を上げるためにBTBエントリ数を増やすとハードウェア量が大きくなるという欠点がある。

ハードウェア量を小さくするためには、エントリ数を減らすか大きさを小さくするかのどちらかであるが、BTBのヒット率を落とさないためにエントリ数を減らすことを避けると、エントリの大きさを小さくするしかない。BTBエントリはそのほとんどが分岐先情報で占められているので、エントリの大きさを抑えるにはこの分岐先情報を削減する必要がある。しかしながら、従来のBTB分岐予測では、BTBから読み出した分岐先情報をtaken予測の場合は次サイクルの命令フェッチに予測分岐先アドレスとして用いる必要がある(図1-a)、BTBからの読み出しも含めて1サイクルで予測先を決定しなければならず、予測先を生成するために加算などを行う時間を得られない。したがって、分岐先のアドレスをそのまま(30bit)保持する必要があり分岐先情報を削減することはできなかった。

3.変位保持BTB方式

3.1.構成および処理手順

我々が提案する変位保持BTB方式は、プログラム・カウンタ(PC)から分岐先までの変位を分岐先情報として保持させることによりBTBエントリの大きさを削減する。以下にその処理手順を示す。

①登録・更新:分岐命令が実行されると、その分岐先アドレスから分岐命令のアドレスを減算した値、つまり、PCから分岐先までの変位を求め、その値を分岐命令の1つ前の命令アドレスで指定されるBTBエントリに登録・更新する。

②予測:命令フェッチ・ステージにおいて、命令キャッシュと同じアドレスでBTBのアクセスを行う。BTB登録を分岐命令の1つ前のアドレスを用いて行っているため、読み出した情報は次サイクルでフェッチする命令の予測情報である。つまり、予測を反映させるのは次のサイクルの終わりではなく、1サイクルの余裕があるので、この時間を利用して、変位+PCの計算を行い予測分岐先を生成し、taken予測であれば生成した予測分岐先アドレスをPCに設定する(図1-b)。

③予測のチェック:予測が間違っていた場合、命令の無効化と命令再フェッチを行う。

3.2.保持する変位のbit数

保持する変位のbit数が少ないとBTBエントリの大きさを小さくできるが、変位で表現可能な分岐距離も限定され予測ヒット率が低下する可能性がある。そこで、分岐距離の分布を3つのSPEC INTを含む8つの整数系プログラムについて調べた。図2はその算術平均であり、PC相対で分岐するBranch、絶対ア

Reducing the branch target information on a Branch Target Buffer

Tetsuya HARA, Hideki ANDO, Chikako NAKANISHI and Masao NAKAYA

System LSI Laboratory, Mitsubishi Electric Corporation

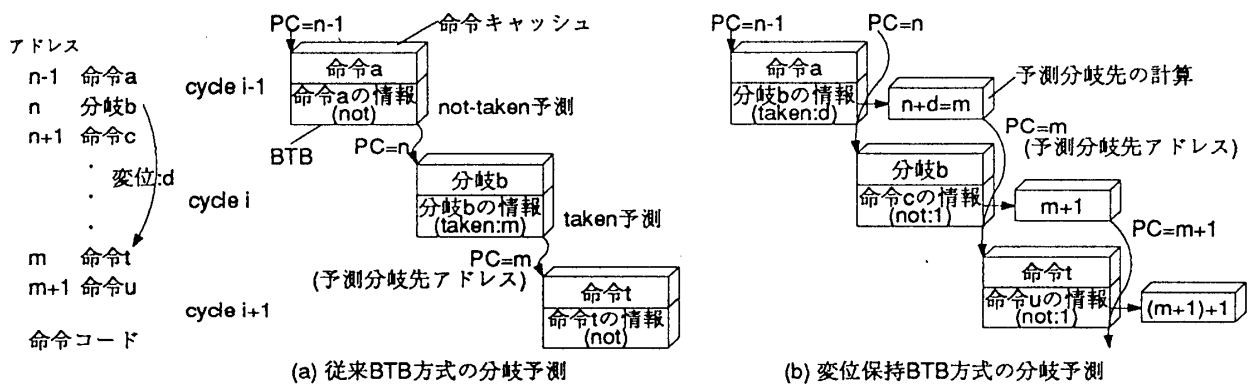


図1 BTB方式の分岐予測方法と予測分岐先の生成

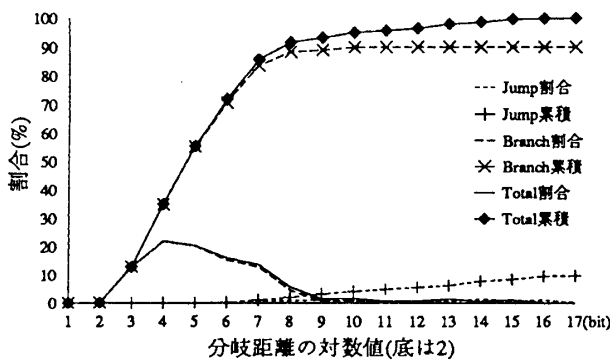


図2 分岐距離の分布

ドレスで分岐するJump、BranchとJumpを合計した分岐全体(Total)について分布と累積の割合を示している。これによると、変位が10bitで95%の分岐命令を、17bitで100%をカバーすることができる。

3.3.別情報読み出しによる予測阻害

図3のように、アドレスがjの分岐命令1はその分岐情報をアドレスj-1のBTBに登録する。分岐命令0が実行されその分岐先が分岐命令1であるとすると、読み出していたのは命令i+1の予測情報であり、分岐先の分岐命令1のものでないで予測を行うことができない。この影響は次章で述べる。

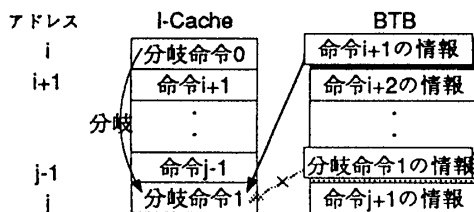


図3 別情報読み出しが生じる場合

4.評価

変位保持BTB方式の有効性を評価するために、面積と分岐予測ヒット率の見積を行った。

表1 BTBの面積

	I-Cache	BTB	
容量	8KB	35bit/entry	17bit/entry
面積	8.89(mm ²)	3.46	2.14

表2 分岐予測ヒット率

	Intmm	bubble	perm	quick	queens	towers	tree	average
従来方式	96.0%	81.6	70.6	63.4	78.9	90.9	84.6	80.86%
変位保持方式	95.4%	80.7	70.6	63.4	78.8	88.8	79.5	79.60%

4.1.面積

本方式によるBTB面積の減少を確かめるために、命令キャッシュおよびBTBの面積の見積を行った。想定プロセスは0.5μmCMOSである。命令キャッシュの容量は8KB、BTBはエントリ数が512でエントリの大きさは従来方式が35bit、変位保持方式が17bit(変位は12bit)とする。表1にその面積を示す。

本方式によって分岐先情報を12bitに削減しエントリの大きさを35bit→17bitと半減した場合、その面積も従来方式の6割程度に小さくすることができる。

4.2.分岐予測ヒット率

別情報読み出しによる予測阻害の影響を調べるために、ソフトウェア・シミュレーションによる予測ヒット率の評価を行った。MIPS R3000の命令コードを実行するシミュレータを作成し、従来のBTB方式と変位保持BTB方式に対して、整数系であるスタンフォード・ベンチマークの実行を行った。

表2に、それぞれの方式による分岐予測ヒット率を示す。変位保持方式の別情報読み出しによる予測阻害は、平均で1.26%と僅かであることが分かる。

5.おわりに

以上、変位保持BTB方式の構成、動作、および、その有効性について述べた。

本方式は、BTBの読み出しタイミングを1サイクル前にずらすことによって生じる時間を分岐先の復元に用いることにより、BTBに保持する分岐先情報を変位の10数bitに削減する。これにより、分岐予測ヒット率をほとんど低下させることなくBTBの面積を6割程度に抑えられることが分かった。

参考文献

- [1] J.L. Hennessy, and D.A.Patterson, Computer Architecture: A Quantitative Approach, Morgan Kauffmann Pub., Inc., 1990.
- [2] Mike Johnson, "Superscalar Microprocessor Design," Prentice-Hall, Inc. 1991.
- [3] MICROPROCESSOR REPORT, MicroDesign Resources, Mar. 24, 1993.