

1C-5

マイクロ・プロセッサ向き
分岐命令先取り方式についての検討

堀川 隆*、難波 信治*、加藤 哲**

*日本電気(株) C & Cシステム研究所、**日本電気技術情報システム開発(株)

1. はじめに

パイプライン型のプロセッサでは、分岐命令実行時においてパイプラインの各ステージに空き時間の生じることが、実行時間を低下させる一因となっている。この問題を解決するため、ディレイド・ブランチ、スカッシュド・ブランチ¹⁾などが提案されている。しかし、これらの方法を既存のマイクロ・プロセッサに適用しようとする、分岐命令の機能を変更することになり、ソフトウェア資産の継承を考慮すると容易に採用することはできない。本稿では、このような変更を行うことなく分岐のコストを下げる分岐予測方式について効果とハードウェア量について評価を行った結果を報告する。

2. 分岐命令実行時に生じるロスの分類

ここでは、分岐命令実行時に生じるパイプラインの空き時間をロスと呼び、これを次の2つに分類する。(図1)

- 1)分岐先アドレスの計算が終了するまで命令フェッチを行うことができない。(ロス1)
- 2)条件分岐命令の場合、命令①の実行が終了し、条件コードが設定されるまで分岐命令の次に実行すべき命令が定まらない。(ロス2)

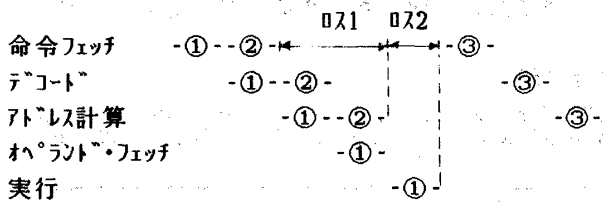


図1 簡単なパイプライン型計算機の実行モデル

ロス1は、総ての分岐命令で生じる。これに対処するためには、分岐先アドレスを予測する必要がある。(予測1)

ロス2は、条件分岐命令についてのみ生じる。これに対処するためには、条件分岐における分岐の成立、不成立を予測する必要がある。(予測2)

一般には、図1に示すようにロス1とロス2を比べると、ロス1の影響が大きいと考えられる。

3. プロセッサ・モデルと分岐予測の効果

- 1)条件分岐命令の条件成立、不成立が確定するまで次の命令のデコードやオペランド・フェッチを行わない方式のプロセッサ(プロセッサ1)では、条件分岐命令について予測1を行って分岐先の命令をフェッチしても、デコード段階で待たされることになるため、分岐予測の効果は予測2と大差なく、ロス2の分と考えられる。無条件分岐命令に関しては、デコードを待たせることなく先行させる。
- 2)条件分岐命令の条件成立、不成立が確定するより前に次の命令のデコード、実効アドレス計算、オペランド・フェッチまで行い、実行のみを保留するタイプのプロセッサ(プロセッサ2)で予測1を行うと、ロス1+ロス2の分をなくすることができ、効果が大きくなる。

	無条件分岐	条件分岐成立	条件分岐不成立
予測なし	0	0	0
予測1	0x1	0x2/0x1+0x2	0/0x1
予測2	0	0x2	0/0x1

表1 分岐予測により削減できるロス (°:0x1/°:0x2)

4. 分岐予測方式とハードウェア量

分岐予測方式として、次の3種類を考える。これらについてのハードウェア量を比較するため、予測に必要なメモリ量を示す。

I. 連想メモリ方式

分岐命令の存在するアドレスを連想メモリ、対応する分岐先アドレスをメモリに記録することにより予測1を行う。ここでは、条件分岐命令のみを対象にする方式、無条件に分岐する命令のみを対象にする方式、両方を対象にする方式の3種類、エントリ数を1~8を考えた。

アドレスを32ビットとし、連想メモリのコストを通常のメモリの2倍と考えると、連想メモリの1エントリ当り、12バイトのメモリが必要である。

II. キャッシュ・メモリ方式

キャッシュ・メモリに分岐予測のためのビット

を持たせて、前回の分岐が成立したかどうかを記録し、これをもとに予測2を行う。容量が256バイト~2Kバイトの命令キャッシュを対象とし、4バイトを単位として分岐予測のビットを2ビット(前回の分岐の成否を記録するビットが有効であるかどうかを示すビットも含める)を持たせる方式を考える。

この方式では、キャッシュの容量が256バイト当たり、16バイトのメモリが必要である。

III. スタティック予測方式

命令コードで定まる分岐条件と分岐距離をキーとする分岐の成立、不成立の予測をあらかじめ記録しておき、これをもとに予測2を行う。

{分岐条件の種類×分岐距離を区分した数}のメモリが必要である。VAXの場合、分岐条件の種類を14種類に分類し、分岐距離を8に区分したため、14バイトの読み出し専用メモリが必要である。

以上より、容量が256バイトのキャッシュ・メモリ方式とスタティック予測方式と連想メモリ方式の1.5エントリ分のハードウェア量がほぼ等しいと考えられる。

6. シミュレーション

VAXのトレース・データ9種類(Cで記述されたプログラム)について、シミュレーションを行った。全命令数は約600万であり、このうち分岐命令は以下の通りである。

- 無条件に分岐する命令 約66万
- 分岐の成立する条件分岐命令 約63万
- 分岐の成立しない条件分岐命令 約67万

これらについて、分岐予測が成功した命令、失敗した命令、分岐予測が設定されていないために予測できなかった命令の割合を調べた。この結果

の一例を表2に示す。この表における効果の欄は、ロス1がロス2の2倍であると仮定して導いた相対値である。

7. 結果および考察

- 1) キャッシュ・メモリ方式では、分岐予測が設定されているときの的中率は9割程度と高いが、数Kバイト程度のキャッシュ・メモリでは、分岐予測が設定されていないことが多く、全体としての効果はスタティック予測方式と余り変わらない。同様なことが、連想メモリ方式についてもいえる。
- 2) 同程度のハードウェアを用いた分岐命令の予測を行うのであれば、連想メモリ方式の方が効果は大きい。プロセッサ1については、連想メモリ方式でエントリ数が8以下の場合、無条件分岐のみを対象とする方が全分岐命令を対象とするより効果は大きい。プロセッサ2では、エントリ数が4以上になると全分岐命令を対象とするほうが効果が大きくなる。
- 3) スタティック予測方式は、少ないハードウェア量で高い的中率を得ることができるが、的中率はプログラムの種類に大きく依存するので、分岐履歴を記録する方式との併用が効果的である。

8. 今後の課題

- 1) プロセッサのタイミングを評価して、コストを正確に評価する。
- 2) 分岐予測に失敗したときのコストを評価する。

参考文献

1) "Reducing the Cost of Branches", Proc. 13th. Annual Sym. on Comp. Arch., 1986, pp.396-403.

	無条件分岐			条件分岐 成立			条件分岐 不成立			効果		
	予測成功	予測失敗	予測不可	予測成功	予測失敗	予測不可	予測成功	予測失敗	予測不可	°0tツツ1	°0tツツ2	
連想メモリ分岐方式 エントリ数	1	0.0%	0.0%	100.0%	5.9%	0.3%	93.8%	0.0%	0.4%	99.6%	372	14506
	2	24.2%	0.0%	75.8%	8.6%	1.8%	89.6%	25.4%	0.6%	74.0%	3748	18195
	4	37.4%	0.0%	62.6%	22.4%	3.3%	74.3%	41.6%	3.2%	55.1%	6366	22188
	8	48.0%	0.0%	52.0%	34.8%	5.0%	60.2%	54.5%	4.5%	41.0%	8551	25774
スタティック方式	79.4%	20.6%	82.5%	17.5%	5000	16092	

表2 シミュレーション結果および効果の一例