

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4027620号
(P4027620)

(45) 発行日 平成19年12月26日(2007.12.26)

(24) 登録日 平成19年10月19日(2007.10.19)

(51) Int. Cl. F I
G06F 9/38 (2006.01) G06F 9/38 330B

請求項の数 14 (全 16 頁)

(21) 出願番号	特願2001-186473 (P2001-186473)	(73) 特許権者	000005223
(22) 出願日	平成13年6月20日(2001.6.20)		富士通株式会社
(65) 公開番号	特開2003-5956 (P2003-5956A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43) 公開日	平成15年1月10日(2003.1.10)	(74) 代理人	100070150
審査請求日	平成18年4月24日(2006.4.24)		弁理士 伊東 忠彦
前置審査		(72) 発明者	多湖 真一郎
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	山名 智尋
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	竹部 好正
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

最終頁に続く

(54) 【発明の名称】 分岐予測装置、プロセッサ、及び分岐予測方法

(57) 【特許請求の範囲】

【請求項1】

過去の分岐命令の履歴を保持する履歴レジスタと、

該履歴レジスタが保持する該履歴と命令アドレスとから異なる分岐命令間で同一のインデックスを共有可能な第1のインデックスを生成するインデックス生成回路と、

各第1のインデックスに対して該命令アドレスの一部であるタグと分岐のし易さを示す第1の値とを格納する履歴テーブルと、

該命令アドレスの少なくとも一部を第2のインデックスとして該命令アドレスが示す命令の分岐先アドレス又は予測分岐先アドレスと分岐のし易さを示す第2の値とを格納する分岐先バッファと、

該第1の値及び該第2の値の何れかを選択することで分岐予測を行う選択ユニットを含むことを特徴とする分岐予測装置。

【請求項2】

過去の分岐命令の履歴を保持する履歴レジスタと、

該履歴レジスタが保持する該履歴と命令アドレスとから第1のインデックスを生成するインデックス生成回路と、

各第1のインデックスに対して該命令アドレスの一部であるタグと分岐のし易さを示す第1の値とを格納する履歴テーブルと、

該命令アドレスの少なくとも一部を第2のインデックスとして該命令アドレスが示す命令の分岐先アドレス又は予測分岐先アドレスと分岐のし易さを示す第2の値とを格納する

分岐先バッファと、

該第 1 の値及び該第 2 の値の何れかを選択することで分岐予測を行う選択ユニットを含み、

該選択ユニットは、現在の命令アドレスに対して該分岐先バッファにエントリが存在し且つ該現在の命令アドレスと現在の履歴とに対して該履歴テーブルにエントリが存在する場合には該第 1 の値を選択し、該現在の命令アドレスに対して該分岐先バッファにエントリが存在し且つ該現在の命令アドレスと現在の履歴とに対して該履歴テーブルにエントリが存在しない場合には該第 2 の値を選択することを特徴とする分岐予測装置。

【請求項 3】

該選択ユニットは、該現在の命令アドレスに対して該分岐先バッファにエントリが存在しない場合には分岐しないと予測することを特徴とする請求項 2 記載の分岐予測装置。

10

【請求項 4】

該インデックス生成回路は、該履歴レジスタが保持する該履歴と該命令アドレスとの排他的論理和として該第 1 のインデックスを生成することを特徴とする請求項 1 乃至 3 のいずれか一項に記載の分岐予測装置。

【請求項 5】

該履歴テーブルは該第 1 のインデックスの各々に対して複数のエントリが登録できるように複数個設けられることを特徴とする請求項 1 乃至 4 のいずれか一項に記載の分岐予測装置。

【請求項 6】

20

過去の分岐命令の履歴を保持する履歴レジスタと、

該履歴レジスタが保持する該履歴と命令アドレスとから異なった分岐命令間で同一のインデックスを共有可能な第 1 のインデックスを生成するインデックス生成回路と、

各第 1 のインデックスに対して該命令アドレスの一部であるタグと分岐のし易さを示す第 1 の値とを格納する履歴テーブルと、

該命令アドレスの少なくとも一部を第 2 のインデックスとして該命令アドレスが示す命令の分岐先アドレスと分岐のし易さを示す第 2 の値とを格納する分岐先バッファと、

該第 1 の値及び該第 2 の値の何れかを選択することで分岐予測を行う選択ユニットと、

命令の実行を制御する実行制御ユニットと、

該命令を実行する演算実行ユニット

30

を含むことを特徴とするプロセッサ。

【請求項 7】

過去の分岐命令の履歴を保持する履歴レジスタと、

該履歴レジスタが保持する該履歴と命令アドレスとから第 1 のインデックスを生成するインデックス生成回路と、

各第 1 のインデックスに対して該命令アドレスの一部であるタグと分岐のし易さを示す第 1 の値とを格納する履歴テーブルと、

該命令アドレスの少なくとも一部を第 2 のインデックスとして該命令アドレスが示す命令の分岐先アドレスと分岐のし易さを示す第 2 の値とを格納する分岐先バッファと、

該第 1 の値及び該第 2 の値の何れかを選択することで分岐予測を行う選択ユニットと、

40

命令の実行を制御する実行制御ユニットと、

該命令を実行する演算実行ユニット

を含み、

該選択ユニットは、現在の命令アドレスに対して該分岐先バッファにエントリが存在し且つ該現在の命令アドレスと現在の履歴とに対して該履歴テーブルにエントリが存在する場合には該第 1 の値を選択し、該現在の命令アドレスに対して該分岐先バッファにエントリが存在し且つ該現在の命令アドレスと現在の履歴とに対して該履歴テーブルにエントリが存在しない場合には該第 2 の値を選択することを特徴とするプロセッサ。

【請求項 8】

該選択ユニットは、該現在の命令アドレスに対して該分岐先バッファにエントリが存在し

50

ない場合には分岐しないと予測することを特徴とする請求項 7 記載の プロセッサ。

【請求項 9】

該インデックス生成回路は、該履歴レジスタが保持する該履歴と該命令アドレスとの排他的論理和として該第 1 のインデックスを生成することを特徴とする請求項 6 乃至 8 のいずれか一項に記載の プロセッサ。

【請求項 10】

該履歴テーブルは該第 1 のインデックスの各々に対して複数のエントリが登録できるように複数個設けられることを特徴とする請求項 6 乃至 9 のいずれか一項に記載の プロセッサ。

【請求項 11】

過去の分岐命令の履歴と命令アドレスとから異なった分岐命令間で同一のインデックスを共有可能に生成する第 1 のインデックスに対して該命令アドレスの一部であるタグと分岐のし易さを示す第 1 の値とを格納する履歴テーブルと、該命令アドレスの少なくとも一部を第 2 のインデックスとして該命令アドレスが示す命令の分岐先アドレスと分岐のし易さを示す第 2 の値とを格納する分岐先バッファとを設けた構成において、

該第 1 の値及び該第 2 の値の何れかを選択し、

選択した値に応じて分岐予測を行う

各段階を含むことを特徴とする分岐予測方法。

【請求項 12】

過去の分岐命令の履歴と命令アドレスとから生成する第 1 のインデックスに対して該命令アドレスの一部であるタグと分岐のし易さを示す第 1 の値とを格納する履歴テーブルと、該命令アドレスの少なくとも一部を第 2 のインデックスとして該命令アドレスが示す命令の分岐先アドレスと分岐のし易さを示す第 2 の値とを格納する分岐先バッファとを設けた構成において、

該第 1 の値及び該第 2 の値の何れかを選択し、

選択した値に応じて分岐予測を行う

各段階を含み、

該選択する段階は、現在の命令アドレスに対して該分岐先バッファにエントリが存在し且つ該現在の命令アドレスと現在の履歴とに対して該履歴テーブルにエントリが存在する場合には該第 1 の値を選択し、該現在の命令アドレスに対して該分岐先バッファにエントリが存在し且つ該現在の命令アドレスと現在の履歴とに対して該履歴テーブルにエントリが存在しない場合には該第 2 の値を選択することを特徴とする分岐予測方法。

【請求項 13】

該現在の命令アドレスに対して該分岐先バッファにエントリが存在しない場合に該分岐先バッファに該現在の命令アドレスを登録し、

該現在の命令アドレスに対して該分岐先バッファにエントリが存在し且つ該現在の命令アドレスと現在の履歴とに対して該履歴テーブルにエントリが存在せず且つ該第 2 の値に基づく予測結果が誤りである場合に該履歴テーブルに該現在の命令アドレスに関する登録を行う

各段階を含むことを特徴とする請求項 12 記載の分岐予測方法。

【請求項 14】

該現在の命令アドレスに対して該分岐先バッファにエントリが存在し且つ該現在の命令アドレスと現在の履歴とに対して該履歴テーブルにエントリが存在せず且つ該第 2 の値に基づく予測結果が正しい場合に該履歴テーブルに該現在の命令アドレスに関する登録を行わないことを特徴とする請求項 12 又は 13 記載の分岐予測方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、一般に分岐命令の分岐予測方式及びプロセッサに関し、詳しくは、PHT (Pattern History Table) を用いた分岐予測方式及びプロセッサに関する。

10

20

30

40

50

【 0 0 0 2 】

【従来の技術】

パイプライン動作を実行するプロセッサにおいては、分岐命令の分岐結果を待ってから分岐先アドレスに分岐するのでは、命令フェッチのタイミングが遅くなりパイプライン動作に乱れが生じる。従って、分岐命令を実行する前に分岐予測をし、パイプライン動作の流れに沿った一連の命令フェッチを可能にする必要がある。

【 0 0 0 3 】

分岐命令には、その分岐命令自体が分岐し易い或いは分岐し難いといったように、ローカルな意味で分岐方向に偏りが存在する場合が多い。また最近実行された分岐命令の分岐結果に依存してある分岐命令が分岐し易い或いは分岐し難いといったように、グローバルな意味で分岐方向に偏りが存在する場合も多い。PHT (Pattern History Table) は、このような分岐傾向のローカル性及びグローバル性を考慮して、高い精度で分岐を予測する方式である。

10

【 0 0 0 4 】

図1は、従来のPHTを利用した分岐予測方式の構成図である。

【 0 0 0 5 】

図1の分岐予測装置は、XOR回路11、GHRユニット12、及びPHTユニット13を含む。GHR (Global History Register) 12は、最近実行した分岐命令について分岐したか否かに関する履歴を記録しておくレジスタである。分岐命令が分岐した場合にはレジスタ内部を左に1ビットシフトしながら最下位ビットに1を格納し、分岐命令が分岐しなかった場合にはレジスタ内部を左に1ビットシフトしながら最下位ビットに0を格納する。例えば、GHR 12が6ビット長であり、現在の内容が“011001”であるとする。この状態で、ある分岐命令を実行した結果が「分岐」であったとすると、GHR 12の内容を左に1ビットシフトし更に1を最下位ビットに挿入する。結果として、GHR 12の内容は“110011”となる。この“110011”は、6回前の分岐命令が分岐、5回前の分岐命令が分岐、4回前の分岐命令が非分岐、3回前の分岐命令が非分岐、2回前の分岐命令が分岐、最も最近の分岐命令が分岐であったことを示す。

20

【 0 0 0 6 】

XOR回路11は、プログラムカウンタ10が示す実行する分岐命令アドレスとGHRユニット12の内容との排他的論理和を計算する。計算された排他的論理和は、PHTユニット13にインデックスとして供給される。

30

【 0 0 0 7 】

PHT 13は、各インデックスに対して例えば2ビットのカウンタを格納するRAMである。ここでインデックスは、上述のようにプログラムカウンタ10の示す実行する分岐命令アドレスとGHRユニット12の内容との排他的論理和である。また、インデックスの内容である2ビットのカウンタ値は、そのインデックスをヒットした場合の予測結果である。カウンタ値が0及び1の場合には分岐しないと予測し、カウンタ値が2及び3の場合には分岐すると予測する。

【 0 0 0 8 】

例えば、GHR 12の内容が“110011”であり、分岐命令アドレスが“001000”である場合、インデックスは“111011”となる。このインデックス“111011”の内容の2ビットカウンタ値を参照して、例えばカウンタ値が2であるとする。上述のように2或いは3のカウンタ値は分岐予測を意味するので、命令アドレスが“001000”である現在実行しようとしている分岐命令は、分岐すると予測する。実際に命令を実行した結果として分岐すれば、カウンタ値に1を加算する。また実際に命令を実行した結果として分岐しない場合には、カウンタ値から1を減算する。従って、例えば実際の実行結果として分岐したとすると、カウンタ値は3に設定される。

40

【 0 0 0 9 】

その後、例えば、GHRユニット12が全く同一の内容“110011”である状態で、同じ分岐命令アドレス“001000”の分岐命令を再度実行する場合、インデックスは前回と同一の

50

“111011”となる。この場合のカウント値は3であり、分岐命令は分岐すると予測する。前述のように、GHRユニット12の内容は、最近実行した分岐命令の分岐結果の履歴である。従って、最近の分岐傾向が同一の条件で同一の分岐命令を実行した場合には、その分岐結果は同一のインデックスに蓄積されていく。次回同一のインデックスを参照したときには、そのカウント値が分岐予測に用いられる。

【0010】

最近実行した分岐命令の分岐結果の履歴が、上記の履歴と若干異なる場合として、例えばGHRユニット12の内容が“110010”であるとする。これは上記の場合“110011”と比較して、最後の分岐命令の結果が異なっていた場合である。この場合、上記と同一の分岐命令アドレス“001000”の分岐命令を実行するとすると、インデックスは“111010”となる。従って、このインデックスには、過去の分岐の履歴が“110010”である条件で分岐命令アドレス“001000”の分岐命令を実行した場合について、分岐結果が蓄積されていくことになる。

10

【0011】

従って、仮に分岐命令がプログラム中に1つしか存在しない場合には、各インデックスは、この分岐命令の分岐の結果を種々の分岐履歴に対して蓄積することになり、分岐履歴を反映した非常に高い精度での分岐予測が可能となる。しかし分岐命令がプログラム中に複数個存在する場合には、PHTユニット13内で互いの分岐命令の結果が干渉し合うことになり、予測精度が低下する。例えば、GHRユニット12の内容が“111010”の場合に分岐命令アドレス“000001”の分岐命令を実行するとすると、インデックスは“111011”となり、上記のようにGHR12の内容が“110011”であり分岐命令アドレスが“001000”である場合と同一のインデックスとなってしまふ。このようにXOR回路11でインデックスを計算する方式では、異なった分岐命令間で同一のインデックスを共有することになり、分岐結果の記録が干渉してしまい予測精度が低下する。

20

【0012】

予測精度を低下させないためには、GHRユニット12の内容とプログラムカウンタ10の内容とを繋ぎ合わせてインデックスを作成すればよい。例えば、GHR12の内容が“110011”であり分岐命令アドレスが“001000”である場合には、インデックスを“110011001000”とすればよい。しかしこの場合には、インデックスが長くなることで、PHTユニット13のRAMのエントリ数が大幅に増大してしまう。実際、この場合のエントリ数は64倍(=2⁶)に増大することになる。

30

【0013】

【発明が解決しようとする課題】

上述のように、分岐予測に使用するPHTにおいてエントリが干渉する構成においては、分岐予測の精度が低下してしまうという問題がある。しかし予測精度を向上させるために、PHTのメモリ容量を必要以上に増大させることは望ましくない。可能な限り小さなメモリ容量で可能な限り予測精度を向上させることが望ましい。

【0014】

従って本発明は、PHTを使用する分岐予測方式において、可能な限り小さなメモリ容量を使用しながらエントリ干渉を回避して分岐予測の精度を向上させた分岐予測方式を提供することを目的とする。

40

【0015】

【課題を解決するための手段】

本発明による分岐予測装置は、過去の分岐命令の履歴を保持する履歴レジスタと、該履歴レジスタが保持する該履歴と命令アドレスとから異なった分岐命令間で同一のインデックスを共有可能な第1のインデックスを生成するインデックス生成回路と、各第1のインデックスに対して該命令アドレスの一部であるタグと分岐のし易さを示す第1の値とを格納する履歴テーブルと、該命令アドレスの少なくとも一部を第2のインデックスとして該命令アドレスが示す命令の分岐先アドレスと分岐のし易さを示す第2の値とを格納する分岐先バッファと、該第1の値及び該第2の値の何れかを選択することで分岐予測を行う選択

50

ユニットを含むことを特徴とする。

【0016】

上記分岐予測装置において、該選択ユニットは、現在の命令アドレスに対して該分岐先バッファにエントリが存在し且つ該現在の命令アドレスと現在の履歴とに対して該履歴テーブルにエントリが存在する場合には該第1の値を選択し、該現在の命令アドレスに対して該分岐先バッファにエントリが存在し且つ該現在の命令アドレスと現在の履歴とに対して該履歴テーブルにエントリが存在しない場合には該第2の値を選択することを特徴とする。

【0017】

上記分岐予測装置においては、履歴テーブル(PHT)のエントリに命令の一部をタグとして設けることで、異なった分岐命令間でのPHTエントリの干渉を避けることが可能になる。しかしながらあるインデックスを参照しても、タグが現在の命令に一致しない場合には、当該履歴状況での当該命令に関する情報は登録されていないことになる。このような場合には、分岐先バッファ(BTB)を分岐予測に使用する。即ち、履歴テーブルがヒットした場合(タグが一致)には、履歴テーブルの第1の値を分岐予測に使用し、履歴テーブルがミスした場合(タグが不一致)には、分岐先バッファの第2の値を分岐予測に使用する。

【0018】

このように本発明による分岐予測方式では、エントリ干渉を回避して高精度な分岐予測が可能になるので、実際の分岐方向が確定する前に予測に基づいて命令フェッチしても、予測がはずれる可能性は小さく、パイプライン動作の乱れを避けることが出来る。

【0019】

また本発明による分岐予測方式では、分岐先バッファの第2の値を予測に併用することにより、従来のようにPHT単体で分岐予測を実行する場合と比較して、分岐結果を学習させるまでの時間が短くてすむ。従って、本発明による分岐予測方式は、コンテキストスイッチによる状況変化に短時間で対応できる。

【0020】

【発明の実施の形態】

以下に、本発明の実施例を添付の図面を用いて詳細に説明する。

【0021】

図2は、本発明によるPHTを利用した分岐予測装置の構成図である。

【0022】

本発明による分岐予測装置は、XOR回路21、GHRユニット22、タグ付きPHTユニット23、比較ユニット24、BTB(Branch Target Buffer)25、比較ユニット26、及び選択ユニット27を含む。

【0023】

GHR(Global History Register)22は、最近実行した分岐命令について分岐したか否かに関する履歴を記録しておくレジスタである。分岐命令が分岐した場合にはレジスタ内部を左に1ビットシフトしながら最下位ビットに1を格納し、分岐命令が分岐しなかった場合にはレジスタ内部を左に1ビットシフトしながら最下位ビットに0を格納する。XOR回路21は、プログラムカウンタ20が示す実行する分岐命令アドレスとGHRユニット22の内容との排他的論理和を計算する。計算された排他的論理和は、タグ付きPHTユニット23にインデックスとして供給される。

【0024】

タグ付きPHTユニット23は、XOR回路21の出力である各インデックスに対して、タグ23aとカウント値23bとを格納するRAMである。またBTBユニット25は、命令アドレスの一部をインデックスとして、タグ(命令アドレスの一部)25aと、分岐命令の分岐先アドレス25bと、バイアスビット25cとを格納するRAMである。

【0025】

BTBユニット25は、従来の分岐予測方式においても使用されるユニットであり、分岐

10

20

30

40

50

命令が分岐すると予測されると、その分岐先アドレスを直ちに提供するために設けられる。通常分岐先アドレスは、フェッチした分岐命令をデコードして命令を実行する時に算出する必要があるために、分岐先アドレスの特定には時間を要する。B T Bを設けておき各分岐命令に対して分岐先アドレスを格納しておくことで、分岐命令が分岐すると予測される時に、分岐先アドレスを直ちにB T Bから読み出すことが出来る。ここでB T Bのメモリ容量を小さくするために、命令アドレスの一部をインデックスとし、残りの部分をタグとして、当該インデックスのエントリの一部としてタグを格納しておく。

【0026】

プログラムカウンタ20から命令アドレスが入力されると、この命令アドレスに対応するインデックスからタグ25aを読み出し、読み出したタグ25aと入力命令アドレスの対応部分とを比較する。この比較は、比較ユニット26によって実行される。比較結果が一致を示す場合（ヒットした場合）には、この命令アドレスの分岐命令が登録されていることになり、当該インデックスに格納されている分岐先アドレス25bをプリフェッチアドレスとして使用する。

10

【0027】

バイアスピット25cは、その分岐命令が分岐し易いが分岐し難いかを示す。1ビットで構成される場合には、例えば“1”が分岐し易いことを示し、“0”が分岐し難いことを示す。このバイアスピット25cは、その分岐命令自体が分岐し易いか分岐し難いかを示す、ローカルな意味での分岐方向の偏りに関する情報である。

【0028】

タグ付きPHTユニット23は、XOR回路21が求めた命令アドレスの一部とGHRユニット22との排他的論理和をインデックスとして、過去に実行した分岐結果の情報をカウント値23bとして保持する。本発明においては、タグ付きPHTユニット23は、カウント値23bだけでなく、命令アドレスの一部をタグ23aとして保持している。

20

【0029】

プログラムカウンタ20の命令アドレスとGHRユニット22の内容とからXOR回路21によりインデックスが求められると、タグ付きPHTユニット23からタグ23aを読み出し、読み出したタグ23aと入力命令アドレスの対応部分とを比較する。この比較は、比較ユニット24によって実行される。比較結果が一致を示す場合（ヒットした場合）には、GHRユニット22の内容が示す履歴状態での当該命令アドレスの分岐命令の情報が登録されていることになる。この場合には、このインデックスに格納されているカウント値23bを、分岐予測に使用する。カウント値23bは、例えば2ビットカウンタのカウントであり、0及び1の場合には分岐しないと予測し、2及び3の場合には分岐すると予測する。

30

【0030】

タグ23aを設けることで、異なった分岐命令間でのPHTエントリの干渉を避けることが可能になる。しかしながら、あるインデックスを参照しても、タグ23aが現在の命令に一致しない場合（ミスした場合）には、当該履歴状況での当該命令に関する情報は登録されていないことになる。本発明では、このような場合には、B T Bユニット25のバイアスピット25cを分岐予測に使用する。

40

【0031】

即ち、タグ付きPHTユニット23がヒットした場合（タグが一致）には、タグ付きPHTユニット23のカウント値23bを分岐予測に使用し、タグ付きPHTユニット23がミスした場合（タグが不一致）には、B T Bユニット25のバイアスピット25cを分岐予測に使用する。この選択は、選択ユニット27によって実行される。選択ユニット27は、比較ユニット24からの比較結果によって、タグ付きPHTユニット23がヒットしたか否かを検出する。この検出に基づいて、比較ユニット24は、タグ付きPHTユニット23からのカウント値23b或いはB T Bユニット25からのバイアスピット25cの何れかを選択し、選択したデータに基づいて分岐予測をして分岐予測結果を出力する。なおB T Bユニット25がヒットしていない場合には、分岐先アドレス25bが利用できな

50

いので、一律に「分岐しない」と予測して予測結果を出力する。

【0032】

図3は、本発明による分岐予測のフローチャートである。

【0033】

ステップS1で、BTBユニット25がヒットしたか否かを判断する。ヒットした場合にはステップS3に進み、ミスした場合にはステップS2に進む。

【0034】

ステップS2で、BTBユニット25がヒットしておらず分岐先アドレス25bが利用できないので、一律に「分岐しない」と予測する。

【0035】

ステップS3で、タグ付きPHTユニット23がヒットしたか否かを判断する。ヒットした場合にはステップS5に進み、ミスした場合にはステップS4に進む。

【0036】

ステップS4で、BTBユニット25のバイアスピット25cに基づいて分岐予測を行う。

【0037】

ステップS5で、タグ付きPHTユニット23のカウント値23bに基づいて分岐予測を行う。

【0038】

図4は、本発明によるデータ更新処理のフローチャートである。

【0039】

ステップS1で、BTBユニット25がヒットしたか否かを判断する。ヒットした場合にはステップS3に進み、ミスした場合にはステップS2に進む。

【0040】

ステップS2で、ヒットしなかった対象命令アドレスを、その分岐先アドレスと共にBTBユニット25に登録する。この際、実際に命令を実行した分岐結果を、BTBユニット25のバイアスピット25cとして格納する。即ち、実際の命令実行の結果、分岐をしたのであれば1をバイアスピット25cに格納し、分岐しなかったのであれば0をバイアスピット25cに格納する。

【0041】

ステップS3では、BTBユニット25がヒットしているので、BTBユニット25の対象命令アドレスを更新する。

【0042】

ステップS4で、タグ付きPHTユニット23がヒットしたか否かを判断する。ヒットした場合にはステップS8に進み、ミスした場合にはステップS5に進む。

【0043】

ステップS5で、対象命令を実際に実行した結果、実際の分岐方向が分岐予測と一致したか否かを判断する。ここで判断対象の分岐予測は、タグ付きPHTユニット23がミスした場合であるので、BTBユニット25のバイアスピット25cに基づいて行われたものである。判断の結果、一致する場合にはステップS6に進み、一致しない場合にはステップS7に進む。

【0044】

ステップS6で、何もせずに処理を終了する。これは、分岐予測の結果と実際の分岐の結果とが一致する場合には、現状の分岐予測が適切であると判断できるからである。

【0045】

ステップS7で、タグ付きPHTユニット23に当該履歴状況での当該命令に関する情報を登録する。即ち、対象となるインデックスに、対象となる命令アドレスのタグ23aを格納すると共に、カウント値23bに実際の分岐結果を格納する。例えば、分岐した場合には2(“10”)を格納し、分岐しなかった場合には1(“01”)を格納する。これは、分岐予測の結果と実際の分岐の結果とが一致しない場合には、BTBユニット25のバイ

10

20

30

40

50

アスピット 2 5 c に基づく現状の分岐予測が不適切であると判断できるからである。

【 0 0 4 6 】

ステップ S 8 では、タグ付き P H T ユニット 2 3 がヒットしているので、タグ付き P H T ユニット 2 3 のカウント値 2 3 b を更新する。具体的には、実際の分岐結果が分岐の場合にカウント値 2 3 b を 1 増加させ、分岐結果が分岐でない場合にカウント値 2 3 b を 1 減少させる。なおカウント値 2 3 b が既に最大値（例えば 2 ビットカウンタなら 3）の場合には、実際の分岐結果が分岐であってもカウント値 2 3 b はそのままである。またカウント値 2 3 b が既に最小値（例えば 0）の場合には、実際の分岐結果が非分岐であってもカウント値 2 3 b はそのままである。

【 0 0 4 7 】

ステップ S 9 で、B T B ユニット 2 5 のバイアスピット 2 5 c とタグ付き P H T ユニット 2 3 のカウント値 2 3 b とについて、夫々の値をチェックする。バイアスピット 2 5 c とカウント値 2 3 b とが、夫々 0 と 0（“ 00 ”）であるか或いは夫々 1 と 3（“ 11 ”）である場合に、ステップ S 1 1 に進む。それ以外の場合には、ステップ S 1 0 に進む。

【 0 0 4 8 】

ステップ S 1 0 で、L R U ビット等を更新する。ここで L R U（Least Recently Used）ビットとは、タグ付き P H T ユニット 2 3 の各エントリに付加され、参照されてから最も使われなかったエントリを特定するためのビットである。タグ付き P H T ユニット 2 3 を分岐予測に使用する毎に、この L R U ビットを更新する。L R U ビットの意味については後述する。

【 0 0 4 9 】

ステップ S 1 1 で、タグ付き P H T ユニット 2 3 の対象エントリを無効にする。即ち、タグ付き P H T ユニット 2 3 の当該対象エントリを実質的に削除する。これは、タグ付き P H T ユニット 2 3 のカウント値 2 3 b が特定の分岐方向を強く示唆しており且つ B T B ユニット 2 5 のバイアスピット 2 5 c がそれと同一の分岐方向を示している場合には、B T B ユニット 2 5 のバイアスピット 2 5 c で分岐を予測しても同一の予測結果が得られるので、タグ付き P H T ユニット 2 3 から登録を抹消することで、タグ付き P H T ユニット 2 3 の R A M のメモリ空間を有効に使用するためである。

【 0 0 5 0 】

以上が、本発明によるデータ更新処理である。

【 0 0 5 1 】

以下に、本発明におけるセットアソシアティブ方式について説明する。

【 0 0 5 2 】

本発明においては、図 2 に示されるように、タグ付き P H T ユニット 2 3 が複数個設けられていてもよい。このように複数のタグ付き P H T ユニット 2 3 を設けることで、同一インデックスに対して複数のエントリを格納可能になる。例えば、4 セットの P H T が設けられる 4 - ウェイセットアソシアティブ方式においては、同一インデックスに対して、4 つまでのエントリを格納可能である。

【 0 0 5 3 】

このようにセットアソシアティブ方式を使用した場合には、データ更新時にどのエントリを削除するかを決定する必要がある。例えば 4 - ウェイセットアソシアティブにおいて既に 4 つのエントリを使用している状況で、5 つめのエントリを格納する必要があるとすると、既存の 4 つのエントリの何れかを選択して削除する必要がある。例えば、図 4 のステップ S 7 では、タグ付き P H T ユニット 2 3 に当該履歴状況での当該命令に関する情報を登録するが、既に同一のインデックスに対するエントリが満杯である場合には、既存の登録の何れかを削除して新規登録に置き換える必要がある。

【 0 0 5 4 】

図 4 のステップ S 1 0 に関連して説明した L R U ビットは、既存エントリを新規エントリで置き換える際に、置き換え対象のエントリを特定するために使用される。L R U ビットは、各エントリに対して保持され、各エントリの参照順序を示すビットである。この L R

10

20

30

40

50

Uビットを調べることで、参照されてから最も長期間使われなかったエントリを特定して、新規エントリと置き換えることができる。

【0055】

L R U方式は置き換え対象のエントリを特定する方式の1つに過ぎず、他の方式として、最も使用されなかった最低使用頻度のエントリを置換するL F U (Least Frequently Used)方式、最も過去に登録されたエントリを置換するF I F O (First-In First-Out)方式、任意に選択したエントリを置換するランダム方式などがある。

【0056】

なおセットアソシアティブ方式を使用せずタグ付きP H Tユニット23が一つだけ設けられる場合には、各インデックスに対して1つのエントリしか格納することは出来ない。この場合、タグ付きP H Tユニット23に新規の情報を登録する際に、既に同一のインデックスに対して別の命令アドレスの登録がなされている場合には、この既存の登録を削除して、新規登録に置き換えることになる。

10

【0057】

また図2に示されるように、B T Bユニット25に対してもセットアソシアティブ方式を用いても良い。

【0058】

以下に、本発明による分岐予測を具体的なプログラムの例を用いて説明する。

【0059】

図5は、分岐命令を含むプログラムの一例である。

20

【0060】

G H Rユニット22の長さは6であり、その初期値を000000とする。

【0061】

命令アドレス000001において、B T Bユニット25に0001をインデックスとしてアクセスすると、タグミスが検出され「分岐しない」と予測する。これは図3のステップS2に対応する。実際の命令実行の結果、分岐命令が分岐するので予測がはずれる。B T Bユニット25のインデックス0001のエントリにタグ00、バイアス1、分岐先アドレス000011を登録する。これは図4のステップS2に対応する。

【0062】

G H Rユニット22は、左に1ビットシフトしながら最下位ビットに分岐を表す1を格納し、結果として000001となる。この状態で、予測結果と各レジスタ/メモリの内容は、PC=000001, 予測失敗, BTB[0001]=00-1-000011, GHR:000001となる。ここでPCはプログラムカウンタを示し、B T Bの内容は、タグ00、バイアス1、及び分岐先アドレス000011の順に示される。

30

【0063】

次に命令アドレス000100において、B T Bユニット25に0100をインデックスとしてアクセスすると、タグミスが検出され「分岐しない」と予測する。実際の命令実行の結果、分岐命令が分岐しないので予測があたる。B T Bユニット25のインデックス0100のエントリにタグ00、バイアス0を登録する。

【0064】

G H Rユニット22は、左に1ビットシフトしながら最下位ビットに非分岐を表す0を格納し、結果として000010となる。この状態で、予測結果と各レジスタ/メモリの内容は、PC=000001, 予測成功, BTB[0100]=00-0-*****, GHR:000010となる。

40

【0065】

更に命令アドレス0005、0006、0007、及び0008の分岐命令が分岐しない。結果として、上記と同様に、

PC=000101, 予測成功, BTB[0101]=00-0-*****, GHR=000100

PC=000110, 予測成功, BTB[0110]=00-0-*****, GHR:001000

PC=000111, 予測成功, BTB[0111]=00-0-*****, GHR=010000

50

PC=001000, 予測成功, BTB[1000]=00-0- ***** , GHR=100000
となる。

【 0 0 6 6 】

更に命令アドレス001100において、BTBユニット25に1100をインデックスとしてアクセスすると、タグミスが検出され「分岐しない」と予測する。実際の命令実行の結果、分岐命令が分岐するので予測がはずれる。BTBユニット25のインデックス1100のエントリに、タグ00、バイアス1、分岐先アドレス001001を登録する。

【 0 0 6 7 】

GHRユニット22は、左に1ビットシフトしながら最下位ビットに分岐を表す1を格納し、結果として000001となる。この状態で、予測結果と各レジスタ/メモリの内容は、
PC=001100, 予測失敗, BTB[1100]=00-1-001001, GHR=000001
となる。

10

【 0 0 6 8 】

ここで命令アドレス001001に分岐したので、再び命令アドレス001100において、BTBユニット25に1100をインデックスとしてアクセスを行う。今回はタグヒットが検出され、分岐先アドレス001001を得る。BTBユニット25がタグヒットしたので、タグ付きPHTユニット23がタグヒットするか否かをチェックする。これは図3のステップS3に対応する。

【 0 0 6 9 】

具体的には、アドレス001100とGHRの内容000001との排他的論理和001101をインデックスとして、タグ付きPHTユニット23にアクセスする。タグ判定結果はミスなので、BTBユニット25のバイアスビットに基づいて、「アドレス001001に分岐する」と予測する。これは図3のステップS4に対応する。実際の命令実行の結果、分岐命令が001001に分岐し予測が的中する。従って、図4のステップS6にあるように、BTBユニット25及びタグ付きPHTユニット23の更新はしない。この状態で、予測結果と各レジスタ/メモリの内容は、

20

PC=001100, 予測成功, BTB[1100]=00-1-001001(更新無し), GHR=000011

である。命令アドレス001100の分岐命令によって、ループが更に3回実行されると、

PC=001100, 予測成功, BTB[1100]=00-1-001001(更新無し), GHR=000111

PC=001100, 予測成功, BTB[1100]=00-1-001001(更新無し), GHR=001111

30

PC=001100, 予測成功, BTB[1100]=00-1-001001(更新無し), GHR=011111

となる。

【 0 0 7 0 】

ループが6回実行(5回分岐)された後に、命令アドレス001100において、上記と同様に「アドレス001001に分岐する」と予測する。この場合、ループは6回目で終了するので、分岐命令は分岐せず予測がはずれる。従って、図4のステップS7に示されるように、命令アドレス001100とGHRの内容011111との排他的論理和010011をタグ付きPHTユニット23のインデックスとし、タグ1100とカウント値0を登録する。この状態で、予測結果と各レジスタ/メモリの内容は、

PC=001100, 予測失敗, PHT[010011]=1100-0, GHR=111110

40

となる。

【 0 0 7 1 】

その後、命令アドレス001111において命令アドレス000010に分岐すると、

PC=001111, 予測失敗, BTB[1111]=00-1-000010, GHR=111101

となる。

【 0 0 7 2 】

命令アドレス000010以降の命令を再度実行すると、

PC=000100, 予測成功(非分岐), BTB及びPHT更新なし, GHR=111010

PC=000101, 予測成功(非分岐), BTB及びPHT更新なし, GHR=110100

PC=000110, 予測成功(非分岐), BTB及びPHT更新なし, GHR=101000

50

PC=000111, 予測成功 (非分岐), BTB及びPHT更新なし, GHR=010000

PC=001000, 予測成功 (非分岐), BTB及びPHT更新なし, GHR=100000

PC=001100, 予測成功 (分岐), BTB及びPHT更新なし, GHR=000001

PC=001100, 予測成功 (分岐), BTB及びPHT更新なし, GHR=000011

PC=001100, 予測成功 (分岐), BTB及びPHT更新なし, GHR=000111

PC=001100, 予測成功 (分岐), BTB及びPHT更新なし, GHR=001111

PC=001100, 予測成功 (分岐), BTB及びPHT更新なし, GHR=011111

PC=001100, 予測成功 (非分岐), PHT[010011]=1100-0, GHR=111110

となる。ここで、ループの最後の繰り返しにおける命令アドレス001100の実行においては、BTBユニット25及びタグ付きPHTユニット23が双方共にヒットし、タグ付きPHTユニット23のカウント値23bは0なので、非分岐を予測する。最後に、

PC=001111, 予測成功 (分岐), BTB及びPHT更新なし, GHR=111101

となる。従って、今回は分岐予測が全体的中することになる。

【0073】

このプログラムの実行において、本発明による分岐予測方式では、タグ付きPHTユニット23では1つのエントリしか使用していない。このように、本発明の分岐予測方式は、RAMの使用容量が少なくても、高い予測精度を実現することが出来る方式である。

【0074】

なお図1に示される従来技術の分岐予測方式では、上記プログラムを実行した場合に、命令アドレス000010以降の命令を2度目に実行した際であっても、PHTエントリの干渉により全ての予測を的中させることは出来ない。

【0075】

図6は、本発明による分岐予測装置を採用したプロセッサの構成例を示す。

【0076】

図6のプロセッサ100は、命令キャッシュ101、データキャッシュ102、命令フェッチユニット103、命令実行制御部104、レジスタ105、レジスタ106、演算部107乃至110を含む。

【0077】

命令キャッシュ101及びデータキャッシュ102は、それぞれ命令及びデータを一時的に格納する。命令フェッチユニット103は、プログラムカウンタの示すアドレスの命令を、命令キャッシュ101から順次フェッチする。命令実行制御部104は、命令フェッチユニット103がフェッチした命令を順次デコードして、デコード結果に基づいて命令実行動作を制御する。レジスタ105、レジスタ106、及び演算部107乃至110は、演算実行ユニットを構成する。この演算実行ユニットは、命令実行制御部104の制御の下で動作して、命令に基づいた演算を実行する。ここで演算部107乃至110は、夫々命令0乃至3を独立に実行する形となっており、パイプライン動作を高速に実行可能な構成となっている。

【0078】

図2に示される本発明による分岐予測装置は、命令フェッチユニット103に設けられ、分岐命令があるときにその分岐方向を予測して、予測分岐方向に対応したアドレスの命令をフェッチする。本発明による分岐予測方式ではエントリ干渉を回避して高精度な分岐予測が可能になるので、実際の分岐方向が確定する前に予測に基づいて命令フェッチしても、予測がはずれる可能性は小さく、パイプライン動作の乱れを避けることが出来る。

【0079】

また本発明による分岐予測方式では、BTBユニット25のバイアスピット25cを使用することにより、従来のようにPHT単体で分岐予測を実行する場合と比較して、分岐結果を学習させるまでの時間が短くてすむ。従って、本発明による分岐予測方式は、コンテキストスイッチによる状況変化に短時間で対応できる。8KBのRAMを使用したベンチマークjpeg, jbig, mpeg4, ghostscriptにおいて、本発明による分岐予測方式は、平均で96%の予測精度を実現した。

10

20

30

40

50

【 0 0 8 0 】

以上、本発明を実施例に基づいて説明したが、本発明は上記実施例に限定されるものではなく、特許請求の範囲に記載の範囲内で様々な変形が可能である。

【 0 0 8 1 】

【 発明の効果 】

本発明による分岐予測方式では、エントリ干渉を回避して高精度な分岐予測が可能になるので、実際の分岐方向が確定する前に予測に基づいて命令フェッチしても、予測がはずれる可能性は小さく、パイプライン動作の乱れを避けることが出来る。

【 0 0 8 2 】

また本発明による分岐予測方式では、B T Bのバイアスピットを予測に併用することにより、従来のようにP H T単体で分岐予測を実行する場合と比較して、分岐結果を学習させるまでの時間が短くてすむ。従って、本発明による分岐予測方式は、コンテキストスイッチによる状況変化に短時間で対応できる。

10

【 図面の簡単な説明 】

【 図 1 】 従来のP H Tを利用した分岐予測方式の構成図である。

【 図 2 】 本発明によるP H Tを利用した分岐予測装置の構成図である。

【 図 3 】 本発明による分岐予測のフローチャートである。

【 図 4 】 本発明によるデータ更新処理のフローチャートである。

【 図 5 】 分岐命令を含むプログラムの一例である。

【 図 6 】 本発明による分岐予測装置を採用したプロセッサの構成図である。

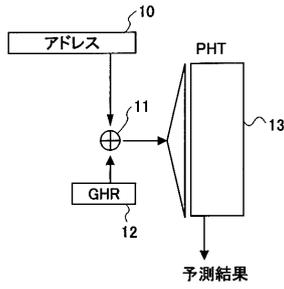
20

【 符号の説明 】

- 2 1 X O R回路
- 2 2 G H Rユニット
- 2 3 タグ付きP H Tユニット
- 2 4 比較ユニット
- 2 5 B T B (Branch Target Buffer)
- 2 6 比較ユニット
- 2 7 選択ユニット 2 7

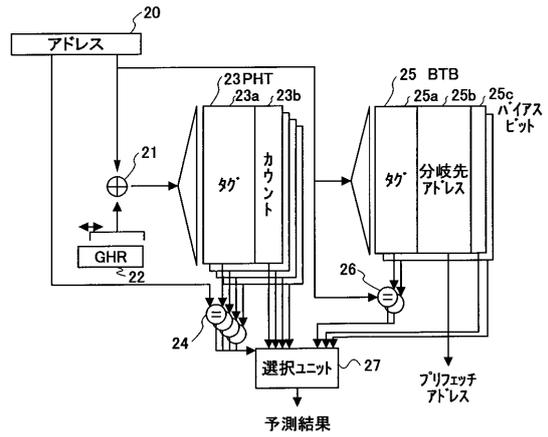
【 図 1 】

従来のPHTを利用した分岐予測方式の構成図



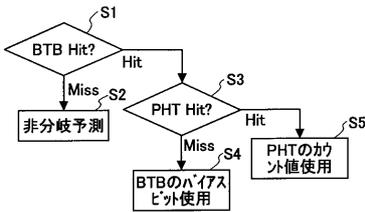
【 図 2 】

本発明によるPHTを利用した分岐予測装置の構成図



【 図 3 】

本発明による分岐予測のフローチャート



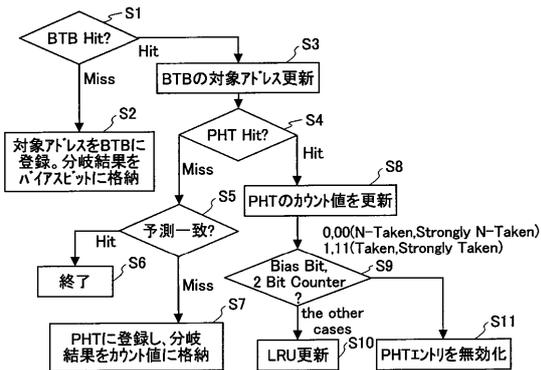
【 図 5 】

分岐命令を含むプログラムの一例

アドレス	命令
000000	演算命令
000001	000011への分岐命令
000010	演算命令
000011	演算命令
000100	分岐命令
000101	分岐命令
000110	分岐命令
000111	分岐命令
001000	分岐命令
001001	演算命令
001010	演算命令
001011	演算命令
001100	001001への分岐命令(5回分岐して1回分岐しない)
001101	演算命令
001110	演算命令
001111	000010への分岐命令
010000	演算命令

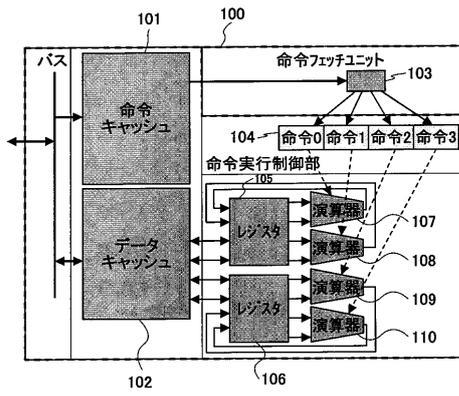
【 図 4 】

本発明によるデータ更新処理のフローチャート



【 図 6 】

本発明による分岐予測装置を採用したプロセッサの構成図



フロントページの続き

審査官 間野 裕一

(56)参考文献 特開平10 - 105401 (JP, A)

特開昭63 - 189943 (JP, A)

Tse-Yu Yeh他1名, "A Comparison of Dynamic Branch Predictors that use Two Levels of Branch History", Proceedings of the 20th Annual International Symposium on Computer Architecture, IEEE, 1993年 5月16日, pp.257-266

(58)調査した分野(Int.Cl., DB名)

G06F 9/38