

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

**特許第3890910号**  
**(P3890910)**

(45) 発行日 平成19年3月7日(2007.3.7)

(24) 登録日 平成18年12月15日(2006.12.15)

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

請求項の数 5 (全 12 頁)

(21) 出願番号	特願2001-79716 (P2001-79716)	(73) 特許権者	000005108 株式会社日立製作所 東京都千代田区丸の内一丁目6番6号
(22) 出願日	平成13年3月21日(2001.3.21)	(74) 代理人	100100310 弁理士 井上 学
(65) 公開番号	特開2002-278752 (P2002-278752A)	(72) 発明者	高村 明裕 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所中央研究所内
(43) 公開日	平成14年9月27日(2002.9.27)	審査官	間野 裕一
審査請求日	平成18年3月20日(2006.3.20)	(56) 参考文献	特開平11-312086 (JP, A) 特開平07-210383 (JP, A)
		(58) 調査した分野(Int.Cl., DB名)	G06F 9/38

(54) 【発明の名称】 命令の実行結果予測装置

(57) 【特許請求の範囲】

【請求項1】

命令を実行するための1つまたは複数の命令実行ユニットと、過去に実行された命令の実行結果から、現在実行中の命令の実行結果の予測値を生成する予測手段を含むプロセッサであって、

前記命令実行ユニットは、前記予測手段の出力する命令の実行結果の予測値を用いて命令を実行し、

前記予測手段は、過去に実行された命令の実行結果を、2つ以上の格納場所から選択して格納可能な実行結果の履歴テーブルと、前記履歴テーブルに格納された過去に実行された命令の実行結果の間の、有用性の順位を格納する有用性順位テーブルと、前記履歴テーブルから読み出された過去に実行した命令の実行結果から、現在実行中の命令の実行結果を予測するための予測値生成手段と、実行結果を予測した命令を実際に実行した結果、前記予測手段の出力した予測値が正しかったか否かを判定する予測判定手段と、前記履歴テーブルに格納する場所を決定する格納場所決定手段と、前記有用性順位テーブルに書き込まれる有用性順位を生成する有用性順位生成手段とを含み、

前記予測手段は、前記履歴テーブルの内容を用いて前記予測手段が現在実行中の命令の実行結果の予測値を生成し、

前記予測判定手段が予測値と、現在実行中の命令の実行結果とが同じであることを示す時に、有用性順位生成手段は、前記予測値生成手段の入力となった実行結果の履歴テーブルの情報が有用であることを示す様に、前記有用性順位テーブル内の有用性順位を上げ、

10

20

前記予測判定手段が予測値と、現在実行中の命令の実行結果とが異なることを示す時に、前記有用性順位テーブル内の有用性順位を変更しないことを特徴とするプロセッサ。

【請求項2】

命令を実行するための1つまたは複数の命令実行ユニットと、過去に実行され命令の実行結果から、現在実行中の命令の実行結果の予測値を生成する予測手段を含むプロセッサであって、

前記命令実行ユニットは、前記予測手段の出力する命令の実行結果の予測値を用いて命令を実行し、

前記予測手段は、過去に実行された命令の実行結果を、2つ以上の格納場所から選択して格納可能な実行結果の履歴テーブルと、前記履歴テーブルに格納された過去に実行された命令の実行結果の間の、有用性の順位を格納する有用性順位テーブルと、前記履歴テーブルから読み出された過去に実行した命令の実行結果から、現在実行中の命令の実行結果を予測するための予測値生成手段と、実行結果を予測した命令を実際に実行した結果、前記予測手段の出力した予測値が正しかったか否かを判定する予測判定手段と、前記履歴テーブルに格納する場所を決定する格納場所決定手段と、前記有用性順位テーブルに書き込まれる有用性順位を生成する有用性順位生成手段とを含み、

前記予測手段は、前記履歴テーブルの内容を用いて前記予測手段が現在実行中の命令の実行結果の予測値を生成し、

前記予測判定手段が予測値と、現在実行中の命令の実行結果とが異なることを示す時に、有用性順位生成手段は、前記予測値生成手段の入力となった実行結果の履歴テーブルの情報が有用ではないことを示す様に、前記有用性順位テーブル内の有用性順位を下げ、前記予測判定手段が予測値と、現在実行中の命令の実行結果とが同じであることを示す時に、前記有用性順位テーブル内の有用性順位を変更しないことを特徴とするプロセッサ。

【請求項3】

命令を実行するための1つまたは複数の命令実行ユニットと、過去に実行された命令の実行結果から、現在実行中の命令の実行結果の予測値を生成する予測手段を含むプロセッサであって、

前記命令実行ユニットは、前記予測手段の出力する命令の実行結果の予測値を用いて命令を実行し、

前記予測手段は、過去に実行された命令の実行結果を、2つ以上の格納場所から選択して格納可能な実行結果の履歴テーブルと、前記履歴テーブルに格納された過去に実行された命令の実行結果の間の、有用性の順位を格納する有用性順位テーブルと、前記履歴テーブルから読み出された過去に実行した命令の実行結果から、現在実行中の命令の実行結果を予測するための予測値生成手段と、実行結果を予測した命令を実際に実行した結果、前記予測手段の出力した予測値が正しかったか否かを判定する予測判定手段と、前記履歴テーブルに格納する場所を決定する格納場所決定手段と、前記有用性順位テーブルに書き込まれる有用性順位を生成する有用性順位生成手段とを含み、

前記予測手段は、前記履歴テーブルの内容を用いて前記予測手段が現在実行中の命令の実行結果の予測値を生成し、

前記予測判定手段が予測値と、現在実行中の命令の実行結果とが同じであることを示す時に、有用性順位生成手段は、前記予測値生成手段の入力となった実行結果の履歴テーブルの情報が有用であることを示す様に、前記有用性順位テーブル内の有用性順位を上げ、前記予測判定手段が予測値と、現在実行中の命令の実行結果とが異なることを示す時に、有用性順位生成手段は、前記予測値生成手段の入力となった実行結果の履歴テーブルの情報が不要であることを示す様に、前記有用性順位テーブル内の有用性順位を下げることを特徴とするプロセッサ。

【請求項4】

命令を実行するための1つまたは複数の命令実行ユニットと、過去に実行された命令の実行結果から、現在実行中の命令の実行結果の予測値を生成する予測手段を含むプロセッサであって、

10

20

30

40

50

前記命令実行ユニットは、前記予測手段の出力する命令の実行結果の予測値を用いて命令を実行し、

予測値を用いて命令を実行する際に、予測を行ない、かつ、予測が正しかった場合に減少するサイクル数と、予測を行ない、かつ、予測が正しくなかった場合に増加するサイクル数を判定する予測成果判定手段を含み、

前記予測手段は、過去に実行された命令の実行結果を、2つ以上の格納場所から選択して格納可能な実行結果の履歴テーブルと、前記履歴テーブルに格納された過去に実行された命令の実行結果の間の、有用性の順位を格納する有用性順位テーブルと、前記履歴テーブルから読み出された過去に実行した命令の実行結果から、現在実行中の命令の実行結果を予測するための予測値生成手段と、実行結果を予測した命令を実際に実行した結果、前記予測手段の出力した予測値が正しかったか否かを判定する予測判定手段と、前記履歴テーブルに格納する場所を決定する格納場所決定手段と、前記有用性順位テーブルに書き込まれる有用性順位を生成する有用性順位生成手段とを含み、

前記予測手段は、前記履歴テーブルの内容を用いて前記予測手段が現在実行中の命令の実行結果の予測値を生成し、

前記予測判定手段が予測値と、現在実行中の命令の実行結果とが同じであることを示す時に、有用性順位生成手段は、前記予測値生成手段の入力となった実行結果の履歴テーブルの情報が有用であることを示す様に、予測成果判定手段の示す減少するサイクル数が大きい程、前記有用性順位テーブル内の有用性順位を大きく上げ、

前記予測判定手段が予測値と、現在実行中の命令の実行結果とが異なることを示す時に、前記有用性順位テーブル内の有用性順位を変更しないことを特徴とするプロセッサ。

#### 【請求項5】

実行結果を予測する命令は分岐命令であることを特徴とする請求項1に記載のプロセッサ

#### 【発明の詳細な説明】

#### 【0001】

#### 【発明の属する技術分野】

本発明は、過去に実行した命令の実行結果から現在実行中の命令の実行結果を予測する方式を有するプロセッサにおける過去に実行した命令の実行結果の管理法に関する。

#### 【0002】

#### 【従来の技術】

近年のプロセッサは、高速化のため、プロセッサ内で実行される処理を命令フェッチ、命令デコード、オペランドの読み出し、実行、実行結果の書き戻しといった処理に細分化し、前後の命令とオーバーラップして実行するパイプライン方式を用いている。

分岐予測機構は、条件つき分岐命令を実行する際に起きるパイプラインの停止が起こらないようにすることで、パイプライン方式を用いているプロセッサの性能を向上させるためのものである。図7に示す様に、分岐予測機構がない場合、条件つき分岐命令の実行が終了するまで、分岐する/しないが決定しないため、条件つき分岐命令の後続の命令に対する命令フェッチができず、パイプラインを停止させなくてはならない。これに対して、分岐予測機構がある場合、条件つき分岐命令の実行結果(分岐する/しない)を前もって予測することにより、分岐命令の完了を待たずに命令フェッチが可能になり、予測が正しければ分岐命令を実行する際、パイプラインを停止させないようにすることができる。

同様に、値予測機構は、現在実行中の命令の実行結果を予測することで、パイプラインを停止させないようにして、プロセッサの性能を向上させる。値予測がない場合、ある命令を実行するにあたって、先行する命令の実行結果を必要としているが、その時点で先行する命令の実行結果が得られていない場合(例えば、多数のサイクルを必要とする除算命令の実行結果が得られていない場合)には、先行する命令の実行結果が得られるまで、パイプラインを停止させなくてはならない。これに対して、値予測機構がある場合、先行する命令の実行結果を予測することにより、予測した実行結果が正しければ、パイプラインを停止させないようにすることができる。

分岐/値予測を実現する一つの方法は、命令を実行した際に、命令の実行結果を履歴テ

10

20

30

40

50

ブルに記憶させておき、命令の実行結果を予測する際に、履歴テーブルを参照し、過去の命令の実行結果から予測を行うことである。予測テーブルの添字には、プログラムカウンタ等が用いられる。

分岐/値予測を行なった時、プロセッサ内の命令実行ユニットは、実行結果を予測した命令の正しい実行結果を求める。正しい実行結果と予測値の比較を行い、予測値が正しければ、予測値を用いて実行した命令の実行結果が有効となる。予測値が誤りの時は、予測値を用いて実行した命令を再実行する。

予測値を用いて実行された命令の再実行が少ない程、高いプロセッサ性能が得られる。したがって、分岐/値予測には高い予測正答率が求められる。

分岐予測装置において、高い予測正答率を得るための方法の一つは、ScottMcFarlingのCombining Branch Predictions. WRL Technical Notes TN-36, Digital Equipment Western Research Laboratory, June 1993. に示されている様に、複数の性質の異なる分岐予測アルゴリズムと、分岐予測アルゴリズムの出力を選択する装置を用いることである。

分岐予測機構は、予測アルゴリズムによって、得意とする分岐命令の振舞いが異なる。そこで、2つ以上の性質の異なる分岐予測アルゴリズムをもつ分岐予測機構と、それぞれの分岐予測アルゴリズムの最近の予測正答/不正答を記憶するテーブルを用いて、どちらの分岐予測アルゴリズムの予測が正しいかを予測することによって、一つの分岐予測アルゴリズムだけを用いるより高い予測正答率を得ることができる。

しかしながら、この方法は、2つ以上の予測アルゴリズムと、予測アルゴリズムを選択する装置を実装しなくてはならない為、回路量が大きくなる欠点がある。分岐予測装置において、高い予測正答率を得るための方法の一つは、特開H08-110857に示されている様に、分岐する確率の低いエントリを無効化することである。エントリを無効化することにより、無効化された分岐命令は分岐しないと判断される。このため、分岐する確率の低いエントリを無効化することによる予測正答率の低下は僅かである一方、分岐命令の実行結果を履歴テーブルに新規に登録する際に分岐する確率が高いエントリを消してしまう確率が低くなり、予測正答率が高くなると考えられる。

しかしながら、この方法は、ある一定の基準を満たすエントリを消去してしまう為に、過剰にエントリを無効化してしまったり、無効化すべきエントリが少なすぎるために、予測正答率を高めることができないことがある。

分岐/値予測において、予測正答率を高くする方法の一つは、命令の実行結果を格納可能なエントリを増やすことである。命令の実行結果を格納可能なエントリが2つ以上あり、そのうちのいずれか一つのエントリに実行結果を格納する構成方式を一般にセットアソシアティブ方式と呼ぶ。セットアソシアティブ方式は、キャッシュメモリによく使用される方式である。セットアソシアティブ方式は、通常、選択可能なエントリ数を $k$ 、エントリ総数を $n$ 、実行結果を格納するテーブルの添字を $i$ とした時、格納するエントリは、 $(i \bmod n) - (i \bmod k)$ から  $(i \bmod n) - (i \bmod k) + (k-1)$ のいずれかとする。

セットアソシアティブ方式のキャッシュメモリにおいて、2つ以上の格納可能なエントリから格納するエントリを選択する方法のうち、最も性能が良いとされる方式は、Least-Recently-Used (LRU)方式と呼ばれる方式である。LRU方式は、選択可能なエントリの中で最も長時間使用されていないエントリに格納する。

これを実現するために、LRU方式は、総エントリ数を $n$ 、選択可能なエントリ数を $k$ とした時、 $n/k$ エントリのLRUテーブルを使用する。LRUテーブルには、エントリが参照された時間を選択可能なエントリ間の相対的な順位で記録する。以下、選択可能なエントリ間の相対的な順位をここでは有用性順位と呼ぶ。

例えば、総エントリ数が16、選択可能なエントリ数を4とした時、8~15番目のエントリが、8 9 10 11の順に参照されたとする。この時、LRUテーブルには、8 9 10 11の順に参照されたことが記録されている。次に9が参照されたとすると、LRUテーブルには、8 10 11 9の順に参照されたことが記録される。そして、新しく格納すべき実行結果があり、その実行結果を8~11エントリのいずれかに格納しなくてはならない時、選択可能なエントリの中で有用性順位の最も低いエントリ、すなわち、最も長時間参照されていな

10

20

30

40

50

い8番目のエントリに実行結果を格納する。

分岐/値予測におけるLRU方式の問題点は、予測失敗となる確率が高くても、参照頻度が高いエントリは、有用性順位が高くなり、予測用テーブルに残るため、予測正答であるエントリが履歴テーブルから消去されてしまうことである。予測正答率を高めるためには、予測失敗となる確率が高いエントリよりも、参照頻度が低くても、予測正答となるエントリが予測用テーブルにある方がよい。すなわち、LRU方式では、参照頻度は高いが予測失敗となる確率も高いエントリのために、プロセッサ性能が低下しているという問題がある。

【0003】

【発明が解決しようとする課題】

本発明の目的は、セットアソシアティブ方式で構成された分岐/値予測機構の予測正答率を高めることである。

10

【0004】

【課題を解決するための手段】

本発明は、上記目的を達成するために、LRUテーブルと同様に選択可能なエントリ間での有用性順位を記録する有用性順位テーブルを用いる。LRUテーブルは、参照の度に参照されたエントリの有用性順位が高くなる様に設定されるが、有用性順位テーブルは、予測のために参照され、予測が正しかった時にだけ、参照されたエントリの有用性順位が高くなる様に設定される。

これによって、分岐/値予測機構において、参照頻度が低いエントリ、及び、予測正答率が低いエントリの有用性順位が低くなる。新しく格納すべき実行結果がある時は、選択可能なエントリのうち最も有用性順位が低いエントリに格納する。この時、履歴テーブルから、性能向上への寄与が小さいエントリが消え、相対的に性能向上への寄与が大きいエントリが履歴テーブルに残るようになるため、プロセッサの性能を向上させることができる。

20

さらに、本発明は、分岐/値予測機構において、履歴テーブルが参照され、予測が失敗の時に、参照されたエントリの有用性順位を低くすることにより、予測テーブルから、性能向上への寄与が小さいエントリが消え、相対的に性能向上への寄与が大きいエントリが履歴テーブルに残るようになるため、プロセッサの性能を向上させることができる。

さらに、本発明は、分岐/値予測機構において、参照され、予測が正しかった時に、参照されたエントリの有用性順位を高め、予測失敗の時に、参照されたエントリの有用性順位を低くすることにより、履歴テーブルから、性能向上への寄与が小さいエントリが消え、相対的に性能向上への寄与が大きいエントリが予測テーブルに残るようになるため、プロセッサの性能を向上させることができる。

30

さらに、本発明は、分岐/値予測機構において、予測が正しかった時は、予測を行なわなかった時に比べて減少するサイクル数に応じて有用性順位を高める。これにより、履歴テーブルから、性能向上への寄与が大きいエントリは残りやすくなり、プロセッサの性能を向上させることができる。

【0005】

【発明の実施の形態】

図1は本発明による第一から第三の実施例を示す分岐/値予測回路のブロック図である。説明の便宜上、図1は4-wayセットアソシアティブ方式、履歴テーブルは1 wayにつき256エントリ(総エントリ数は1024)、参照/書き込みアドレス(ADR)は32ビット(ADR[31:0])、履歴テーブルの参照には参照/書き込みアドレスのLSBから8ビット(ADR[7:0])を使用する構成での実施例であるが、本発明の原理と動作を十分に示している。

40

本発明の実施例は、命令実行ユニット 100、有用性順位テーブル 110、実行結果の履歴テーブルa~d 311~314、タグ一致検出回路a~d 321~324、実行結果選択回路 115、予測値生成回路 116、予測判定回路 117、格納場所決定回路 118、有用性順位生成回路 119、参照/書き込みアドレスレジスタ 122からなる。

【0006】

図1の実施例は、4-wayセットアソシアティブ方式であるので、ここでは、有用性順位をp

50

rio[5:0]の6ビットで表わしている。1番有用性順位が高いデータが格納されているway番号(0~3)をprio[1:0]、2番目に有用性順位が高いデータが格納されているway番号(0~3)をprio[3:2]、3番目に有用性順位が高いデータが格納されているway番号(0~3)をprio[5:4]と表わす。有用性の最も低いデータが格納されているway番号(0~3)はprio[1:0] xor prio[3:2] xor prio[5:4]で求めることができる。例えば、8~15番目のエントリがそれぞれway0~way3に対応していて、8 10 11 9の順にエントリが参照された時、LRUでは9 11 10 8の順に有用性が高いと評価され、有用性順位を示すprioの各ビットは、prio[1:0] = 1、prio[3:2] = 3、prio[5:4] = 2となる。

全体の動作は以下の通りである。まず、命令実行ユニット 100が、実行結果を予測したい命令のアドレスを参照/書き込みアドレスレジスタ 122に格納する。

10

履歴テーブルa~d 311~314から、タグ情報と過去の命令の実行結果がそれぞれ読み出される。タグ一致検出回路a 321は履歴テーブルaから読み出されたタグ(TAGa[31:8])と履歴参照/書き込みアドレスのうち、履歴テーブルの参照に使用しない上位24ビット(ADR[31:8])との比較を行ない、一致するなら履歴a選択ビット 331を1、一致しないなら履歴a選択ビット 331を0とする。同様に、タグ一致検出回路b~d 322~324でも比較を行ない、履歴b~d選択ビット 332~334を出力する。

タグヒット検出回路 111は履歴a~d選択ビット321~324のいずれかが1であればタグヒット信号208に1を、すべてが0なら0であれば0を出力する。

タグヒットデコード回路 120は、1-outof-4符号で出力される履歴a~d選択ビット321~324を入力とし、履歴選択ビットが1となっているway番号(0~3)を出力する。

20

実行履歴選択回路は、タグヒットデコード回路 120が出力するway番号に応じて履歴テーブルから出力された過去の命令の実行結果a~d 301~304のいずれかを過去の命令の実行結果 205に出力する。

予測値生成回路 116は、過去の命令の実行結果 205を入力として、実行結果の予測値206を生成する。

命令実行ユニット 100は、実行結果の予測値 206を用いて命令の実行を行なう。命令実行ユニット 100は、予測を行なった命令の実行結果 221を出力する。予測判定回路は、実行結果の予測値 206と命令の実行結果 221とを比較し、一致する場合は1を、一致しない場合は0を予測正答信号 207に出力する。

有用性順位生成回路と格納場所決定回路は次に示す4つのうち、いずれかの動作を行なう

30

(1a) 履歴テーブルを参照し、履歴テーブルa~dから読み出されたタグのいずれかと参照アドレスの上位24ビット 220が一致し、さらに、実行履歴から生成された予測値と命令の実行結果が一致した場合(予測正答信号 207 が 1の場合)は、履歴テーブル内の予測に使用したエントリの有用性順位を高くする。履歴テーブルの内容は予測アルゴリズムにもとづいて更新する。

(1b) 履歴テーブルを参照し、履歴テーブルa~dから読み出されたタグのいずれかと参照アドレスの上位24ビット 220が一致したが、実行履歴から生成された予測値と命令の実行結果が一致しなかった場合(予測正答信号 207 が 0の場合)は、履歴テーブル内の予測に使用したエントリの有用性順位を変更しない。履歴テーブルの内容は予測アルゴリズムにもとづいて更新する。

40

(1c) 履歴テーブルを参照したが、履歴テーブルa~dから読み出されたタグのいずれとも参照アドレスの上位24ビット 220が一致せず、かつ、予測と命令の実行結果が一致した場合(予測正答信号 207 が 1の場合)履歴テーブルへの登録は不要であるため、何もしない。

(1d) 履歴テーブルを参照したが、履歴テーブルa~dから読み出されたタグのいずれとも参照アドレスの上位24ビット 220が一致せず、かつ、予測と命令の実行結果が一致しなかった場合(予測正答信号 207 が 0の場合)、予測アルゴリズムにもとづいて、実行ユニットの出力する実行結果を履歴テーブルに格納する。以後、この動作を新規登録と呼ぶ。

新規登録の際、格納場所決定回路は、履歴テーブルa~d 311~314の格納可能なエントリ

50

のバリッドビットが「無効」であるかどうかを調べる。バリッドビットが「無効」であるエントリがあれば、そのエントリに書き込みが行なわれる様に、履歴テーブル書き込み制御信号 340をセットする。

バリッドビットが「無効」であるエントリがない場合、履歴テーブルを参照した際に用いたアドレスを用いて有用性順位テーブル110から有用性順位を読み出し、有用性順位の最も低いエントリの履歴テーブル側へ書き込みが行なわれる様に履歴テーブル書き込み制御信号 340をセットする。

#### 【 0 0 0 7 】

図 2 に way 数が 4 である時の有用性順位生成回路の第一の実施例を示す。有用性順位生成回路は、新規登録時の way 番号 212、有用性順位テーブルから読み出された有用性順位 210、予測正答信号 207、タグヒット信号 208、予測のため参照した way 209 を入力とし、有用性順位テーブル 110 に記録すべき有用性順位 266 を出力する。

(1a) の場合、予測正答信号 207 が 1、タグヒット信号 208 が 1 となる。この時、予測のために参照した way を最も有用性の高い way として有用性順位テーブルに登録するため、予測のために参照した way 209 が有用性順位テーブル 110 に記録すべき有用性順位 266 の P0[1:0] に出力される。これにあわせて、予測のために参照した way 以外の way の有用性順位は、予測のために参照した際の有用性順位より高ければ、1 段階低くなる。

(1b) の場合、予測正答信号 207 が 0、タグヒット信号 208 が 1 となる。この時、マルチプレクサ 401 ~ 404 のセレクト信号が 1 となり、有用性順位は変化しない。

(1c) の場合、予測正答信号 207 が 1、タグヒット信号 208 が 0 となる。この時、マルチプレクサ 401 ~ 404 のセレクト信号が 1 となり、有用性順位は変化しない。

(1d) の場合、予測正答信号 207 が 0、タグヒット信号 208 が 0 となる。この時、格納場所決定回路が出力する way を有用性順位が高い way として有用性順位テーブルに登録するため、新規登録する way 212 が有用性順位テーブル 110 に記録すべき有用性順位 266 の P0[1:0] に出力される。これにあわせて、新規登録する way 以外の way の有用性順位は、新規登録する way の有用性順位より高ければ、1 段階低くなる。

以上に示した様に動作することで、分岐/値予測機構において、参照頻度が高く、かつ、予測正答率が高いエントリの有用性順位が高くなる。すなわち、性能向上への寄与が大きいエントリの有用性順位が高くなる。したがって、履歴テーブルから、性能向上への寄与が小さいエントリが消え、相対的に性能向上への寄与が大きいエントリが履歴テーブルに残り、予測正答率を高くすることができる。図 3 に way 数が 4 である時の有用性順位生成回路の第二の実施例を示す。この回路は次に示す 4 つのうち、いずれかの動作を行なう。

(2a) 履歴テーブルを参照し、履歴テーブル a ~ d から読み出されたタグのいずれかと参照アドレスの上位 24 ビット 220 が一致し、さらに、実行履歴から生成された予測値と命令の実行結果が一致した場合(予測正答信号 207 が 1 の場合)は、履歴テーブル内の予測に使用したエントリの有用性順位を変更しない。履歴テーブルの内容は予測アルゴリズムにもとづいて更新する。

(2b) 履歴テーブルを参照し、履歴テーブル a ~ d から読み出されたタグのいずれかと参照アドレスの上位 24 ビット 220 が一致したが、実行履歴から生成された予測値と命令の実行結果が一致しなかった場合(予測正答信号 207 が 0 の場合)は、履歴テーブル内の予測に使用したエントリの有用性順位を低くする。履歴テーブルの内容は予測アルゴリズムにもとづいて更新する。

(2c) 履歴テーブルを参照したが、履歴テーブル a ~ d から読み出されたタグのいずれとも参照アドレスの上位 24 ビット 220 が一致せず、かつ、予測と命令の実行結果が一致した場合(予測正答信号 207 が 1 の場合)履歴テーブルへの登録は不要であるため、何もしない。

(2d) 履歴テーブルを参照したが、履歴テーブル a ~ d から読み出されたタグのいずれとも参照アドレスの上位 24 ビット 220 が一致せず、かつ、予測と命令の実行結果が一致しなかった場合(予測正答信号 207 が 0 の場合)、予測アルゴリズムにもとづいて、実行ユニットの出力する実行結果を履歴テーブルに格納する(新規登録)。

10

20

30

40

50

以上に示した様に動作することで、分岐/値予測機構において、参照頻度が高く、かつ、予測正答率が低いエントリの有用性順位が低くなる。すなわち、性能向上への寄与が小さいエントリの有用性順位が低くなる。したがって、履歴テーブルから、性能向上への寄与が小さいエントリが消え、相対的に性能向上への寄与が大きいエントリが履歴テーブルに残り、予測正答率を高くすることができる。図4に示す有用性順位生成回路は、本発明の第三の実施例である。この回路は次に示す4つのうち、いずれかの動作を行なう。

(3a) 履歴テーブルを参照し、履歴テーブルa~dから読み出されたタグのいずれかと参照アドレスの上位24ビット 220が一致し、さらに、実行履歴から生成された予測値と命令の実行結果が一致した場合(予測正答信号 207 が 1の場合)は、履歴テーブル内の予測に使用したエントリの有用性順位を高くする。履歴テーブルの内容は予測アルゴリズムにもとづいて更新する。 10

(3b) 履歴テーブルを参照し、履歴テーブルa~dから読み出されたタグのいずれかと参照アドレスの上位24ビット 220が一致したが、実行履歴から生成された予測値と命令の実行結果が一致しなかった場合(予測正答信号 207 が 0の場合)は、履歴テーブル内の予測に使用したエントリの有用性順位を低くする。履歴テーブルの内容は予測アルゴリズムにもとづいて更新する。

(3c) 履歴テーブルを参照したが、履歴テーブルa~dから読み出されたタグのいずれとも参照アドレスの上位24ビット 220が一致せず、かつ、予測と命令の実行結果が一致した場合(予測正答信号 207 が 1の場合)履歴テーブルへの登録は不要であるため、何もしない。 20

(3d) 履歴テーブルを参照したが、履歴テーブルa~dから読み出されたタグのいずれとも参照アドレスの上位24ビット 220が一致せず、かつ、予測と命令の実行結果が一致しなかった場合(予測正答信号 207 が 0の場合)、予測アルゴリズムにもとづいて、実行ユニットの出力する実行結果を履歴テーブルに格納する(新規登録)。

以上に示した様に動作することで、分岐/値予測機構において、参照頻度が高く、かつ、予測正答率が高いエントリの有用性順位が高くなる。また、参照頻度が高く、予測正答率が低いエントリの有用性順位が低くなる。すなわち、性能向上への寄与が小さいエントリの有用性順位が低くなる。したがって、履歴テーブルから、性能向上への寄与が小さいエントリが消え、相対的に性能向上への寄与が大きいエントリが履歴テーブルに残り、予測正答率を高くすることができる。 30

図6にway数が4である時の有用性順位生成回路の第四の実施例を示す。図5に第四の実施例における分岐/値予測回路のブロック図を示す。

#### 【0008】

この有用性順位生成回路は次に示す4つのうち、いずれかの動作を行なう。

(4a) 履歴テーブルを参照し、履歴テーブルa~dから読み出されたタグのいずれかと参照アドレスの上位24ビット 220が一致し、さらに、実行履歴から生成された予測値と命令の実行結果が一致した場合(予測正答信号 207 が 1の場合)は、履歴テーブル内の予測に使用したエントリの有用性順位を高くする。この時、実行ユニット100が出力する予測を行なったことによって減少するサイクル数が大きい時は信号222が1となり、予測を行なったことによって減少するサイクル数が小さい時は信号222が0となる。有用性順位生成回路は、予測を行なったことによって減少するサイクル数が大きい時(信号222が1の時)は、履歴テーブル内の予測に使用したエントリの有用性順位を最も高くする。これに対して、予測を行なったことによって減少するサイクル数が小さい時(信号222が0の時)は、履歴テーブル内の予測に使用したエントリの有用性順位を2番目に高くする。ただし、予測に使用したエントリの有用性順位が元々1番目である時は、有用性順位を変更しない。履歴テーブルの内容は予測アルゴリズムにもとづいて更新する。 40

(4b) 履歴テーブルを参照し、履歴テーブルa~dから読み出されたタグのいずれかと参照アドレスの上位24ビット 220が一致したが、実行履歴から生成された予測値と命令の実行結果が一致しなかった場合(予測正答信号 207 が 0の場合)は、有用性順位を変更しない。履歴テーブルの内容は予測アルゴリズムにもとづいて更新する。 50



(4c) 履歴テーブルを参照したが、履歴テーブルa~dから読み出されたタグのいずれとも参照アドレスの上位24ビット 220が一致せず、かつ、予測と命令の実行結果が一致した場合(予測正答信号 207 が 1 の場合)履歴テーブルへの登録は不要であるため、何もしない。

(4d) 履歴テーブルを参照したが、履歴テーブルa~dから読み出されたタグのいずれとも参照アドレスの上位24ビット 220が一致せず、かつ、予測と命令の実行結果が一致しなかった場合(予測正答信号 207 が 0 の場合)、予測アルゴリズムにもとづいて、実行ユニットの出力する実行結果を履歴テーブルに格納する(新規登録)。

以上に示した様に動作することで、分岐/値予測機構において、参照頻度が高く、予測正答率が高いエントリであって、予測が正しかった時に減少するサイクル数の大きいものほど有用性順位が高くなる。これにより、履歴テーブルから、性能向上への寄与が大きいエントリほど残りやすく、性能向上の小さいエントリは消えやすくなる。このため、プロセッサの性能を高めることができる。

【0009】

【発明の効果】

本発明は以上のように構成されているので、以下に示すように分岐/値予測機構の予測正答率を高める効果がある。

まず第1の効果は、(a)予測の為に参照され、かつ、予測が正解である時のみ、予測に使用したエントリの有用性順位を高くする (b)履歴テーブルに命令の実行結果を新規登録する際、有用性順位が最も低いエントリに登録を行なう ことにより、予測的中率を向上させることができることである。

第2の効果は、(a)予測の為に参照され、かつ、予測が失敗である時のみ、予測に使用したエントリの有用性順位を低くする (b)履歴テーブルに命令の実行結果を新規登録する際、有用性順位が最も低いエントリに登録を行なう ことにより、予測的中率を向上させることができることである。

第3の効果は、(a)新規登録時と予測の為に参照され、かつ、予測が正解である時は、予測に使用したエントリの有用性順位を高くする (b)新規登録時と予測の為に参照され、かつ、予測が失敗である時に、予測に使用したエントリの有用性順位を低くする (c)履歴テーブルに命令の実行結果を新規登録する際、有用性順位が最も低いエントリに登録を行なう ことにより、予測的中率を向上させることができることである。

第4の効果は、(a)新規登録時と予測の為に参照され、予測が正解であり、予測が正しかった時に減少するサイクル数の大きいほど予測に使用したエントリの有用性順位を高くする (b)履歴テーブルに命令の実行結果を新規登録する際、有用性順位が最も低いエントリに登録を行なう ことにより、プロセッサの性能を高めることができることである。

【図面の簡単な説明】

【図1】本発明の実施例を示すブロック図。

【図2】4-wayセットアソシアティブ方式の有用性順位生成回路の第1の実施例。

【図3】4-wayセットアソシアティブ方式の有用性順位生成回路の第2の実施例。

【図4】4-wayセットアソシアティブ方式の有用性順位生成回路の第3の実施例。

【図5】本発明の第4の実施例を示すブロック図。

【図6】4-wayセットアソシアティブ方式の有用性順位生成回路の第4の実施例。

【図7】パイプライン方式のプロセッサにおける分岐予測機構による高速化の例。

【符号の説明】

100 命令実行ユニット

110 有用性順位テーブル

111 タグヒット検出回路

115 実行結果選択回路

116 予測値生成回路

117 予測判定回路

118 格納場所決定回路

10

20

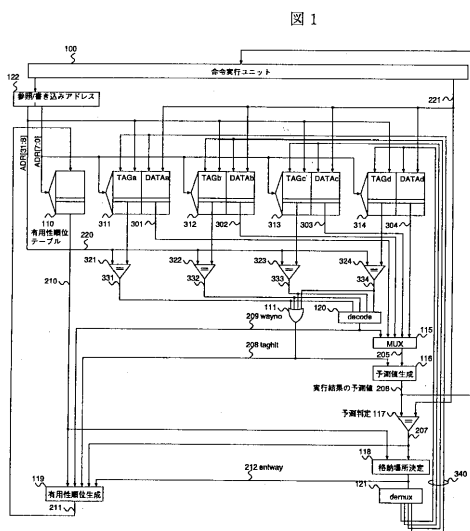
30

40

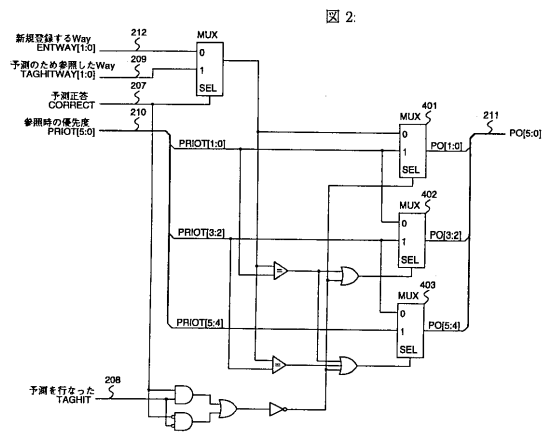
50

- 119 有用性順位生成回路
- 120 タグヒットデコード回路
- 122 参照/書き込みアドレスレジスタ
- 311 実行結果の履歴テーブルa
- 312 実行結果の履歴テーブルb
- 313 実行結果の履歴テーブルc
- 314 実行結果の履歴テーブルd
- 321 タグ一致検出回路a
- 322 タグ一致検出回路b
- 323 タグ一致検出回路c
- 324 タグ一致検出回路d。

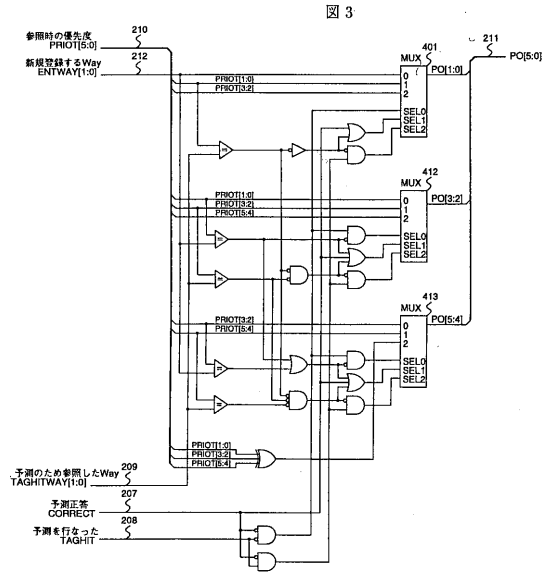
【 図 1 】



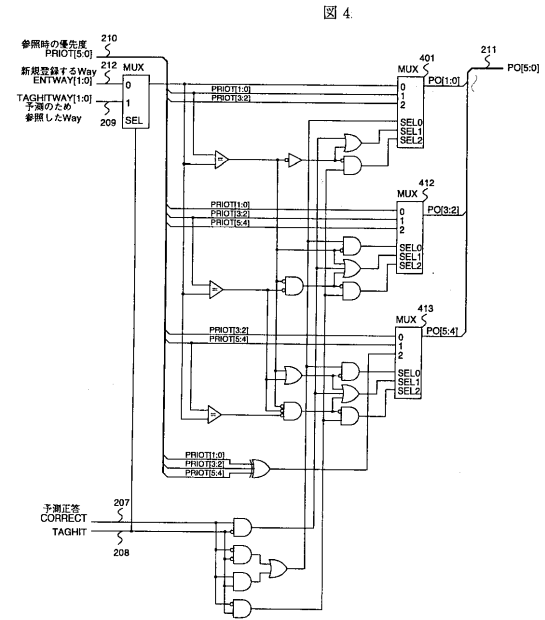
【 図 2 】



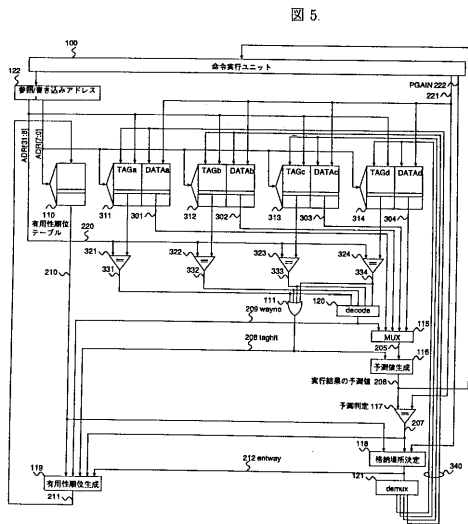
【 図 3 】



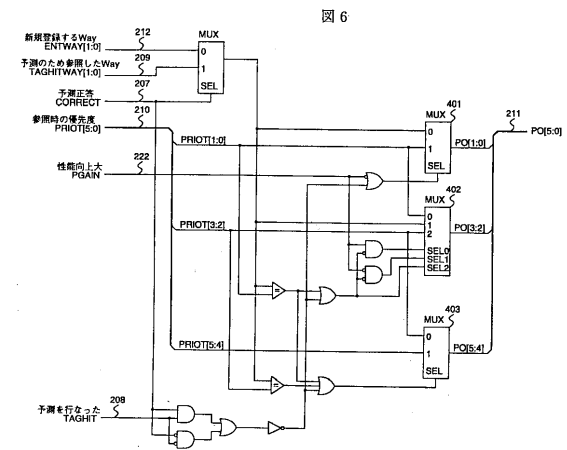
【 図 4 】



【 図 5 】



【 図 6 】



【 図 7 】

