

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5167489号  
(P5167489)

(45) 発行日 平成25年3月21日(2013.3.21)

(24) 登録日 平成25年1月11日(2013.1.11)

(51) Int. Cl. F I  
**G06F 9/38 (2006.01)** G O 6 F 9/38 3 3 0 B  
**G06N 3/08 (2006.01)** G O 6 N 3/08 Q

請求項の数 10 (全 37 頁)

(21) 出願番号	特願2008-526684 (P2008-526684)	(73) 特許権者	504133110 国立大学法人電気通信大学 東京都調布市調布ヶ丘一丁目5番地1
(86) (22) 出願日	平成19年2月6日(2007.2.6)	(74) 代理人	100082131 弁理士 稲本 義雄
(86) 国際出願番号	PCT/JP2007/051965	(72) 発明者	二ノ宮 康之 東京都調布市調布ヶ丘1丁目5番地1 電 気通信大学内
(87) 国際公開番号	W02008/012957	(72) 発明者	阿部 公輝 東京都調布市調布ヶ丘1丁目5番地1 電 気通信大学内
(87) 国際公開日	平成20年1月31日(2008.1.31)	審査官	中野 裕二
審査請求日	平成21年12月4日(2009.12.4)		
(31) 優先権主張番号	特願2006-203106 (P2006-203106)		
(32) 優先日	平成18年7月26日(2006.7.26)		
(33) 優先権主張国	日本国(JP)		

最終頁に続く

(54) 【発明の名称】 情報処理装置および方法、プログラム、並びに記録媒体

(57) 【特許請求の範囲】

【請求項1】

過去における分岐命令の分岐結果が時系列に並べられたグローバル履歴を用いて分岐命令の分岐方向を予測する情報処理装置であって、

前記グローバル履歴に含まれる各分岐結果を履歴順に分割してステージ毎に順次処理を行うパイプライン構造を有し、

前記分岐命令のアドレスに前記分岐命令毎の分岐結果を並べたローカル履歴を反映させた多重化アドレスを生成する多重化アドレス生成手段と、

前記各分岐結果に対して事前になされた分岐予測の正否に応じて付与された重みが前記多重化アドレス生成手段により生成された前記多重化アドレスと対応付けられ前記ステージ毎に割り当てられた重みテーブルより、重みを選択する重み選択手段と、

前記重み選択手段により選択された前記各重みにより、前記ステージ毎に割り当てられた前記グローバル履歴の各分岐結果に対して重み付けを行うことによって重み付き分岐結果値を取得し、前記重み付き分岐結果値を順次累算して重み付き分岐結果累算値を算出する重み付き分岐結果値累算手段と、

前記多重化アドレス生成手段により生成された前記多重化アドレスをインデックスとして最終段のステージに割り当てられた前記重みテーブルより閾値を選択する閾値算出手段と、

前記最終段のステージにおいて、前記重み付き分岐結果値累算手段により累算された前記重み付き分岐結果値に、前記閾値算出手段により算出された前記閾値を加算することに

10

20

より、予測値を算出する予測値算出手段と  
を備える情報処理装置。

【請求項 2】

前記最終段のステージを除いて、前記グローバル履歴に含まれる各分岐結果を履歴順に所定長を有する複数の鎖に分割し、前記各鎖を前記ステージ毎に順次処理することにより、前記鎖の本数分の処理を同時並列的に行うパイプライン構造を有し、

前記重み選択手段は、いずれかの前記鎖のいずれかの前記ステージにおいては、前記多重化アドレスを用いて、割り当てられた前記重みテーブルより重みを選択し、

前記重み付き分岐結果値累算手段は、前記鎖の前記ステージ毎に前記重み付き分岐結果値を取得し、前記鎖毎に前記重み付き分岐結果値を順次累算して重み付き分岐結果累算値を算出し、

前記予測値算出手段は、前記重み付き分岐結果値累算手段により前記鎖毎に累算された重み付き分岐結果累算値の総和を算出し、算出した重み付き分岐結果累算値の総和を用いて予測値を算出する

請求項 1 に記載の情報処理装置。

【請求項 3】

前記最終段のステージを除いて、前記グローバル履歴に含まれる各分岐結果を履歴順に同一ステージで同時並列的に処理する複数のグループに分割することにより、前記グループの数分の長さを有し前記ステージ毎に順次処理を行う鎖が複数本形成されるパイプライン構造を有し、

前記重み選択手段は、いずれかの前記鎖のいずれかの前記ステージにおいては、前記多重化アドレスを用いて、割り当てられた前記重みテーブルより重みを選択し、

前記重み付き分岐結果値累算手段は、前記鎖の前記ステージ毎に前記重み付き分岐結果値を取得し、前記鎖毎に前記重み付き分岐結果値を順次累算して重み付き分岐結果累算値を算出し、

前記予測値算出手段は、前記重み付き分岐結果値累算手段により前記鎖毎に累算された重み付き分岐結果累算値の総和を算出し、算出した重み付き分岐結果累算値の総和を用いて予測値を算出する

請求項 1 に記載の情報処理装置。

【請求項 4】

前記重み選択手段は、

予測を行う分岐命令のアドレスであるターゲットアドレスに最も近い分岐命令のアドレスに対する分岐結果が割り当てられた前記鎖の前記ステージにおいては、その時点で取得可能な最新の分岐結果に対応する分岐命令のアドレスである最新のアドレスのみをインデックスとして前記重みテーブルより前記重みを選択し、

前記ターゲットアドレスに最も近い分岐命令のアドレスに対する分岐結果以外の分岐結果が割り当てられた前記鎖の前記ステージにおいては、前記最新のアドレスと前記多重化アドレスとの排他的論理和をインデックスとして前記重みテーブルより前記重みを選択する

請求項 2 または 3 に記載の情報処理装置。

【請求項 5】

前記閾値算出手段は、前記多重化アドレス生成手段により生成された前記多重化アドレスをインデックスとして最終段のステージに割り当てられた前記重みテーブルより閾値を選択する代わりに、前記最終段のステージに割り当てられた前記重みテーブルから、その時点で取得可能な最新の分岐結果に対応する分岐命令のアドレスである最新のアドレスに対応する重みを閾値の候補として全て特定し、前記ローカル履歴を用いて全候補の中から前記閾値を選択し、

前記重み付き分岐結果値累算手段は、最終段の 1 つ前のステージの 1 本目の鎖において、前記重み選択手段により選択された前記重みにより、1 つ前の分岐命令に対する予測値に対して重み付けを行うことにより重み付き分岐結果値を取得する

10

20

30

40

50

請求項 1 乃至 4 のいずれかに記載の情報処理装置。

【請求項 6】

前記閾値算出手段は、前記多重化アドレス生成手段により生成された前記多重化アドレスをインデックスとして最終段のステージに割り当てられた前記重みテーブルより閾値を選択する代わりに、前記最終段のステージに割り当てられた前記重みテーブルから、その時点で取得可能な最新の分岐結果に対応する分岐命令のアドレスである最新のアドレスに対応する重みを閾値の候補として全て特定し、前記候補の特定と並行して前記最新のアドレスに対応する前記ローカル履歴を特定し、前記ローカル履歴を用いて全候補の中から前記閾値を選択する

請求項 1 乃至 5 のいずれかに記載の情報処理装置。

10

【請求項 7】

前記予測値算出手段により算出された前記予測値が分岐命令の実行結果と一致するか否か、並びに、前記予測値の信頼性に応じて、前記重みテーブルを更新する学習処理を行う学習処理部をさらに備える

請求項 1 乃至 6 のいずれかに記載の情報処理装置。

【請求項 8】

過去における分岐命令の分岐結果が時系列に並べられたグローバル履歴を用いて分岐命令の分岐方向を予測する情報処理方法であって、

前記グローバル履歴に含まれる各分岐結果を履歴順に分割してステージ毎に順次パイプライン処理を行い、

20

前記分岐命令のアドレスに前記分岐命令毎の分岐結果を並べたローカル履歴を反映させた多重化アドレスを生成し、

前記各分岐結果に対して事前になされた分岐予測の正否に応じて付与された重みが前記多重化アドレスと対応付けられ前記ステージ毎に割り当てられた重みテーブルより、重みを選択し、

選択された前記各重みにより、前記ステージ毎に割り当てられた前記グローバル履歴の各分岐結果に対して重み付けを行うことによって重み付き分岐結果値を取得し、前記重み付き分岐結果値を順次累算して重み付き分岐結果累算値を算出し、

前記多重化アドレスをインデックスとして最終段のステージに割り当てられた前記重みテーブルより閾値を選択し、

30

前記最終段のステージにおいて、累算された前記重み付き分岐結果値に、算出された前記閾値を加算することにより、予測値を算出する

ステップを含む情報処理方法。

【請求項 9】

過去における分岐命令の分岐結果が時系列に並べられたグローバル履歴を用いて分岐命令の分岐方向を予測する処理を行うプログラムであって、

前記グローバル履歴に含まれる各分岐結果を履歴順に分割してステージ毎に順次パイプライン処理を行い、

前記分岐命令のアドレスに前記分岐命令毎の分岐結果を並べたローカル履歴を反映させた多重化アドレスを生成し、

40

前記各分岐結果に対して事前になされた分岐予測の正否に応じて付与された重みが前記多重化アドレスと対応付けられ前記ステージ毎に割り当てられた重みテーブルより、重みを選択し、

選択された前記各重みにより、前記ステージ毎に割り当てられた前記グローバル履歴の各分岐結果に対して重み付けを行うことによって重み付き分岐結果値を取得し、前記重み付き分岐結果値を順次累算して重み付き分岐結果累算値を算出し、

前記多重化アドレスをインデックスとして最終段のステージに割り当てられた前記重みテーブルより閾値を選択し、

前記最終段のステージにおいて、累算された前記重み付き分岐結果値に、前記閾値算出手段により算出された前記閾値を加算することにより、予測値を算出する

50

ステップをコンピュータに実行させるプログラム。

【請求項 10】

請求項 9 に記載のプログラムが記録されている記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理装置および方法、プログラム、並びに記録媒体に関し、特に、予測に必要な時間を抑制するとともに分岐予測の精度を向上させ、プロセッサのプログラム実行速度を向上させることができるようにした情報処理装置および方法、プログラム、並びに記録媒体に関する。

10

【背景技術】

【0002】

近年、プロセッサのパイプラインはさらなる深化の傾向を見せており、パイプラインが深くなるにつれ制御ハザードに起因する性能低下はより深刻なものとなってきている。制御ハザードはプログラム内に存在するif文やWhile文に代表される分岐命令によって引き起こされる。これを回避する1つの手段として用いられるのが分岐命令の分岐方向を予測する分岐予測である。1990年代に入りgshare分岐予測器が考えられたが、わずかな予測精度向上でプロセッサのパフォーマンスは大幅に向上するため、分岐予測器には、さらなる予測精度の向上が要求された。

【0003】

20

そこで、ニューラルネットワークの一種であるパーセプトロンを応用したパーセプトロン分岐予測器が注目された。

【0004】

例えば、図1に示されるように、予測すべき分岐命令のアドレスであるターゲットアドレスとプログラムの分岐結果を時系列に並べたグローバル履歴(GH)を入力として用いるグローバルパーセプトロン分岐予測器(Global Perceptron)があった(例えば、特許文献1参照)。

【0005】

この分岐予測器の場合、図1に示されるように、まず、ターゲットアドレス(TA)の下位ビットをインデックスとして重みテーブルより重みベクトルを読み出す。この重みベクトルは、分岐結果に対して事前になされた分岐予測の正否に応じて付与され、各要素がグローバル履歴(GH)の特定の場所と1対1に対応付けされている。グローバル履歴(GH)の各要素は、その値が「1」であれば分岐成立であり、値が「-1」であれば分岐不成立であることを表している。読み出された重みベクトル(重み群)の各要素とグローバル履歴(GH)を1つずつ乗算することにより重み付き分岐結果値を求め、この重み付き分岐結果値の総和の値が正であれば分岐成立(Taken)と予測され、値が負であれば分岐不成立(Not Taken)と予測される。このグローバルパーセプトロン分岐予測器を式で表すと、式(1)のように表すことができる。

30

【0006】

【数 1】

40

$$y = W_0 + \sum_{i=1}^N W_i X_i \quad \dots (1)$$

【0007】

なお、式(1)において、 $W_i$ は読み出された重みを示し、 $X_i$ ( $1 \leq i \leq n$ )は、各重みの要素に対応するグローバル履歴(GH)の要素を示す。また、 $W_0$ は、分岐成立、不成立を区別するための閾値であり、重みベクトルとともに重みテーブルより読み出される。

【0008】

求められた  $y$  の値が正ならば分岐成立として値「1」が出力され、 $y$  の値が負ならば分岐不成立として値「-1(または0)」が出力される。

50

【 0 0 0 9 】

この重み  $W_i$  の更新規則は分岐結果を  $t \in \{1, -1\}$  とすると基本的に次の式(2)によって表される。

【 0 0 1 0 】

【数2】

$$W_i = W_i + \alpha t X_i \quad \dots (2)$$

【 0 0 1 1 】

なお、式(2)において、係数  $\alpha$  の値は通常の場合「1」である。

【 0 0 1 2 】

このようなグローバルパーセプトロン分岐予測器は、gshare分岐予測器などの予測器と比べて予測精度は高いが、予測に必要となる計算量が多いため、ターゲットアドレスが入力されてから分岐予測結果が得られるまでのレイテンシが非常に大きく、CPU(Central Processing Unit)等への実装は事実上不可能であった。

【 0 0 1 3 】

また、図1に示されるグローバルパーセプトロン分岐予測器では、グローバル履歴(GH)を用いるように説明したが、グローバル履歴の代わりに分岐命令毎の分岐結果を並べたローカル履歴を用いたものも知られている。図4は、その場合のローカルパーセプトロン分岐予測器の例を示す図である。このようにローカル履歴(LHT(Local History Table))より読み出した分岐結果(LH)を用いることにより、ローカルパーセプトロン分岐予測器は、グローバルパーセプトロン分岐予測器よりも予測精度を向上させることができる。ただし、このローカルパーセプトロン分岐予測器の場合も、グローバルパーセプトロン分岐予測器の場合と同様に、予測に必要となる計算量が多いため、ターゲットアドレスが入力されてから分岐予測結果が得られるまでのレイテンシが非常に大きく、CPU等への実装は事実上不可能であった。

【 0 0 1 4 】

レイテンシを隠蔽する工夫がなされたパーセプトロン分岐予測器としては、グローバル履歴を履歴順に分割して履歴が古い方から順次処理を行うパイプライン構造を有するパスベースパーセプトロン分岐予測器(Path-Based Perceptron)があった。パスベースパーセプトロン分岐予測器では、図2に示されるように、パイプラインの最終段ステージ以外に割り当てられた重みテーブル(テーブル1乃至テーブルN)より重みを読み出すためのインデックスにターゲットアドレスではなく、過去に実行された分岐命令のアドレスを用い、最終段ステージにおいて重みテーブル(テーブル0)より閾値を読み出すためにのみターゲットアドレスを用いるようになされている。つまり、インデックスに使用するターゲットアドレスを必要最低限に抑えることで、ターゲットアドレスが入力されるより前から、パイプライン化による複数の予測計算を同時に行うことが可能である。

【 0 0 1 5 】

しかしながら、この場合、過去のアドレスを用いて予測を行うため、プログラムが辿るアドレスが変化した場合には、無関係な値によって予測計算がなされてしまうことになる。このため、履歴長を伸ばしても予測精度の向上度はグローバルパーセプトロン分岐予測器に及ばない恐れがあった。

【 0 0 1 6 】

また、パーセプトロン分岐予測器として他に、パイプライン化されたパストレース分岐予測器(Pipelined PTBP(Path Trace Branch Predictor))が考えられた。パイプラインPTBP(Pipelined PTBP)は、前記最終段のステージを除いて、前記パイプラインを履歴順に所定長を有する複数の鎖に分割し、前記各鎖を前記ステージ毎に順次処理することにより、前記鎖の本数分の処理を同時並列的に行うパイプライン構造を有しており、鎖の長さ(図3において横方向の段数)をNとすると1鎖目(図3中、上の鎖)のn(n = N)番目のグローバル履歴に対応する重みは、過去に実行された分岐命令のアドレスを並べた実行パス履歴のn番目のアドレス(読み出し時はその時点で取得可能な最新のアドレス(A[0]))を重

10

20

30

40

50

み選択のインデックスとし、2鎖目の $n + N$ 番目のグローバル履歴に対応する重みは $n$ 番目のアドレス( $A[0]$ )と $n + N$ 番目のアドレス( $A[N]$ )をローテートして排他的論理和を取った値をインデックスとし、3鎖目の $n + 2N$ 番目のグローバル履歴に対応する重みは $n$ 番目のアドレス( $A[0]$ )と $n + N$ 番目のアドレス( $A[N]$ )と $n + 2N$ 番目のアドレス( $A[2N]$ )をローテートして排他的論理和を取った値をインデックスとして重みを読み出す。ここで、ターゲットアドレスに代えて最新のアドレス $A[0]$ を用いているのは、プログラムは局所的なループ構造を持つことが多いため、ターゲットアドレスの数個程度前に処理された分岐命令アドレスであればターゲットアドレスの情報を含んでいる可能性が高いためである。パイプラインPTBPは、このように $N$ 個毎にアドレス履歴の排他的論理和を取り、それをインデックスとして読み出した重みにより、各鎖のステージ毎に割り当てられた各分岐結果に対して重み付けをした重み付き分岐結果値を鎖毎に各ステージにおいて順次累算する。そして、最終段ステージにおいて、鎖毎に累算された重み付き分岐結果値の総和を算出し、その総和とターゲットアドレスをインデックスとしてテーブル0より読み出された閾値としての重みとを加算して分岐予測を行う。このようにすることにより、パイプラインPTBPは、ターゲットの分岐命令までに辿ったパス情報を予測に反映することができ、予測精度を向上させることができた。

【0017】

しかしながら、プログラム中にはグローバルな分岐履歴の一部が異なっても同じふるまいをする分岐命令が多数存在する。このような場合、パイプラインPTBPのパイプライン構造では、予測に関係の無いアドレス履歴によってテーブルの読み出し位置の一部が変化し予測に悪影響を与える恐れがあった。パーセプトロン分岐予測器では、この予測に関係の無いアドレス履歴によって生成されるインデックスで読み出される重みは予測に相関のある重みと異なるものとなってしまう。PTBPはその構造上、1つのアドレス履歴が従来のパーセプトロンに比べて多くの重みのインデックスに関わることになり、この問題が及ぼすPTBPへの影響は他の分岐予測器に比較して大きくなってしまふ。その結果として予測精度の向上を妨げてしまう恐れがあった。

【先行技術文献】

【特許文献】

【0018】

【特許文献1】

特開平10 - 171653号公報

【発明の概要】

【発明が解決しようとする課題】

【0019】

以上のように、パーセプトロン分岐予測器において、様々な工夫がなされるものの、レイテンシを増大させずに予測精度を向上させることができないという課題があった。また、CPUの性能向上のために、分岐予測器の更なる予測精度の向上が求められているという課題もあった。

【0020】

本発明は、このような状況に鑑みてなされたものであり、予測に必要な時間を抑制するとともに分岐予測の精度を向上させることができるようにするものである。

【課題を解決するための手段】

【0021】

本発明の一側面は、過去における分岐命令の分岐結果が時系列に並べられたグローバル履歴を用いて分岐命令の分岐方向を予測する情報処理装置であって、グローバル履歴に含まれる各分岐結果を履歴順に分割してステージ毎に順次処理を行うパイプライン構造を有し、分岐命令のアドレスに分岐命令毎の分岐結果を並べたローカル履歴を反映させた多重化アドレスを生成する多重化アドレス生成手段と、各分岐結果に対して事前になされた分岐予測の正否に応じて付与された重みが多重化アドレス生成手段により生成された多重化アドレスと対応付けられステージ毎に割り当てられた重みテーブルより、重みを選択する

重み選択手段と、重み選択手段により選択された各重みにより、ステージ毎に割り当てられたグローバル履歴の各分岐結果に対して重み付けを行うことによって重み付き分岐結果値を取得し、重み付き分岐結果値を順次累算して重み付き分岐結果累算値を算出する重み付き分岐結果値累算手段と、多重化アドレス生成手段により生成された多重化アドレスをインデックスとして最終段のステージに割り当てられた重みテーブルより閾値を選択する閾値算出手段と、最終段のステージにおいて、重み付き分岐結果値累算手段により累算された重み付き分岐結果値に、閾値算出手段により算出された閾値を加算することにより、予測値を算出する予測値算出手段とを備える情報処理装置である。

【0022】

前記最終段のステージを除いて、グローバル履歴に含まれる各分岐結果を履歴順に所定長を有する複数の鎖に分割し、各鎖をステージ毎に順次処理することにより、鎖の本数分の処理を同時並列的に行うパイプライン構造を有し、重み選択手段は、いずれかの鎖のいずれかのステージにおいては、多重化アドレスを用いて、割り当てられた重みテーブルより重みを選択し、重み付き分岐結果値累算手段は、鎖のステージ毎に重み付き分岐結果値を取得し、鎖毎に重み付き分岐結果値を順次累算して重み付き分岐結果累算値を算出し、予測値算出手段は、重み付き分岐結果値累算手段により鎖毎に累算された重み付き分岐結果累算値の総和を算出し、算出した重み付き分岐結果累算値の総和を用いて予測値を算出することができる。

10

【0023】

前記最終段のステージを除いて、グローバル履歴に含まれる各分岐結果を履歴順に同一ステージで同時並列的に処理する複数のグループに分割することにより、グループの数分の長さを有しステージ毎に順次処理を行う鎖が複数本形成されるパイプライン構造を有し、重み選択手段は、いずれかの鎖のいずれかのステージにおいては、多重化アドレスを用いて、割り当てられた重みテーブルより重みを選択し、重み付き分岐結果値累算手段は、鎖のステージ毎に重み付き分岐結果値を取得し、鎖毎に重み付き分岐結果値を順次累算して重み付き分岐結果累算値を算出し、予測値算出手段は、重み付き分岐結果値累算手段により鎖毎に累算された重み付き分岐結果累算値の総和を算出し、算出した重み付き分岐結果累算値の総和を用いて予測値を算出することができる。

20

【0024】

前記重み選択手段は、予測を行う分岐命令のアドレスであるターゲットアドレスに最も近い分岐命令のアドレスに対する分岐結果が割り当てられた鎖のステージにおいては、その時点で取得可能な最新の分岐結果に対応する分岐命令のアドレスである最新のアドレスのみをインデックスとして重みテーブルより重みを選択し、ターゲットアドレスに最も近い分岐命令のアドレスに対する分岐結果以外の分岐結果が割り当てられた鎖のステージにおいては、最新のアドレスと多重化アドレスとの排他的論理和をインデックスとして重みテーブルより重みを選択することができる。

30

【0025】

前記閾値算出手段は、多重化アドレス生成手段により生成された多重化アドレスをインデックスとして最終段のステージに割り当てられた重みテーブルより閾値を選択する代わりに、最終段のステージに割り当てられた重みテーブルから、その時点で取得可能な最新の分岐結果に対応する分岐命令のアドレスである最新のアドレスに対応する重みを閾値の候補として全て特定し、ローカル履歴を用いて全候補の中から閾値を選択し、重み付き分岐結果値累算手段は、最終段の1つ前のステージの1本目の鎖において、重み選択手段により選択された重みにより、1つ前の分岐命令に対する予測値に対して重み付けを行うことによって重み付き分岐結果値を取得することができる。

40

前記閾値算出手段は、多重化アドレス生成手段により生成された多重化アドレスをインデックスとして最終段のステージに割り当てられた重みテーブルより閾値を選択する代わりに、最終段のステージに割り当てられた重みテーブルから、その時点で取得可能な最新の分岐結果に対応する分岐命令のアドレスである最新のアドレスに対応する重みを閾値の候補として全て特定し、候補の特定と並行して最新のアドレスに対応するローカル履歴を

50

特定し、ローカル履歴を用いて全候補の中から閾値を選択することができる。

前記予測値算出手段により算出された前記予測値が分岐命令の実行結果と一致するか否か、並びに、前記予測値の信頼性に応じて、前記重みテーブルを更新する学習処理を行う学習処理部をさらに備えることができる。

【0026】

本発明の一側面は、過去における分岐命令の分岐結果が時系列に並べられたグローバル履歴を用いて分岐命令の分岐方向を予測する情報処理方法であって、グローバル履歴に含まれる各分岐結果を履歴順に分割してステージ毎に順次パイプライン処理を行い、分岐命令のアドレスに分岐命令毎の分岐結果を並べたローカル履歴を反映させた多重化アドレスを生成し、各分岐結果に対して事前になされた分岐予測の正否に応じて付与された重みが多重化アドレスと対応付けられステージ毎に割り当てられた重みテーブルより、重みを選択し、選択された各重みにより、ステージ毎に割り当てられたグローバル履歴の各分岐結果に対して重み付けを行うことによって重み付き分岐結果値を取得し、重み付き分岐結果値を順次累算して重み付き分岐結果累算値を算出し、多重化アドレスをインデックスとして最終段のステージに割り当てられた重みテーブルより閾値を選択し、最終段のステージにおいて、累算された重み付き分岐結果値に、算出された閾値を加算することにより、予測値を算出するステップを実行する情報処理方法である。

10

【0027】

本発明の一側面は、過去における分岐命令の分岐結果が時系列に並べられたグローバル履歴を用いて分岐命令の分岐方向を予測する処理を行うプログラムであって、グローバル履歴に含まれる各分岐結果を履歴順に分割してステージ毎に順次パイプライン処理を行い、分岐命令のアドレスに分岐命令毎の分岐結果を並べたローカル履歴を反映させた多重化アドレスを生成し、各分岐結果に対して事前になされた分岐予測の正否に応じて付与された重みが多重化アドレスと対応付けられステージ毎に割り当てられた重みテーブルより、重みを選択し、選択された各重みにより、ステージ毎に割り当てられたグローバル履歴の各分岐結果に対して重み付けを行うことによって重み付き分岐結果値を取得し、重み付き分岐結果値を順次累算して重み付き分岐結果累算値を算出し、多重化アドレスをインデックスとして最終段のステージに割り当てられた重みテーブルより閾値を選択し、最終段のステージにおいて、累算された重み付き分岐結果値に、算出された閾値を加算することにより、予測値を算出するステップをコンピュータに実行させることができる。

20

30

【0028】

請求項9に記載のプログラムが記録されている記録媒体とすることができる。

【0029】

本発明の一側面においては、グローバル履歴に含まれる各分岐結果を履歴順に分割してステージ毎に順次処理を行うパイプライン構造が有され、分岐命令のアドレスに分岐命令毎の分岐結果を並べたローカル履歴を反映させた多重化アドレスが生成され、各分岐結果に対して事前になされた分岐予測の正否に応じて付与された重みが多重化アドレスと対応付けられステージ毎に割り当てられた重みテーブルより、重みが選択され、その選択された各重みにより、ステージ毎に割り当てられたグローバル履歴の各分岐結果に対して重み付けが行われることによって重み付き分岐結果値が取得され、重み付き分岐結果値が順次累算されて重み付き分岐結果累算値が算出され、多重化アドレスをインデックスとして最終段のステージに割り当てられた重みテーブルより閾値が選択され、最終段のステージにおいて、累算された重み付き分岐結果値に、算出された閾値が加算されることにより、予測値が算出される。

40

【発明の効果】

【0034】

本発明の側面によれば、分岐命令の予測を行うことができる。特に、分岐予測に必要な時間を抑制するとともに分岐予測の精度を向上させることができる。

【図面の簡単な説明】

【0035】

50



- 【図1】従来の分岐予測器の例を説明する図である。
- 【図2】従来の分岐予測器の他の例を説明する図である。
- 【図3】従来の分岐予測器の、さらに例を説明する図である。
- 【図4】従来の分岐予測器の、さらに例を説明する図である。
- 【図5】本発明を適用した情報処理システムの構成例を示すブロック図である。
- 【図6】図5の分岐予測部の詳細な構成例を示すブロック図である。
- 【図7】パラメータの例を説明する図である。
- 【図8】分岐予測パイプラインの構成例を模式的に示す図である。
- 【図9】グローバル履歴長およびパイプライン段数と、予測ミス率との関係を示すグラフである。
- 【図10】多重化アドレスの生成手順を説明する模式図である。
- 【図11】多重化アドレスを用いた重み読み出しの例を説明する図である。
- 【図12】アドレスに付加するローカル履歴の分岐結果の長さとの関係を示すグラフである。
- 【図13】重み付き分岐結果累算値の算出の例を説明するフローチャートである。
- 【図14】パイプライン処理の流れの例を説明する図である。
- 【図15】ステージ0処理の流れの例を説明するフローチャートである。
- 【図16】ステージnからステージ1の処理の流れの例を説明するフローチャートである。

10

- 【図17】第1重み付き分岐結果累算処理の流れの例を説明するフローチャートである。
- 【図18】第2重み付き分岐結果累算処理の流れの例を説明するフローチャートである。
- 【図19】第3重み付き分岐結果累算処理の流れの例を説明するフローチャートである。
- 【図20】学習処理の流れの例を説明するフローチャートである。
- 【図21】シミュレーションに用いた履歴長の例を示す表である。
- 【図22】予測器毎の記憶容量と予測ミス率の関係を示すグラフである。
- 【図23】分岐予測パイプラインの他の構成例を模式的に示す図である。
- 【図24】分岐予測パイプラインの、さらに他の構成例を模式的に示す図である。
- 【図25】予測器毎の記憶容量と予測ミス率の関係を示すグラフである。
- 【図26】ステージ0の閾値を読み出す部分の、他の構成例を示す図である。
- 【図27】ステージ0処理の流れの、他の例を説明するフローチャートである。
- 【図28】分岐予測パイプラインの、さらに他の構成例を模式的に示す図である。
- 【図29】本発明を適用したパーソナルコンピュータの構成例を示すブロック図である。
- 【発明を実施するための形態】

20

30

【0036】

次に、本発明を適用した実施の形態について、図面を参照して説明する。

【0037】

図5は、本発明を適用した情報処理装置の構成例を示すブロック図である。

【0038】

図5の情報処理装置に搭載されるCPU(Central Processing Unit)11は、命令実行を並列化し、前の命令の実行終了を待たずに次の命令を実行するパイプライン処理を行うプロセッサである。CPU11は、メモリ12に記憶されている命令を読み出して実行し、必要に応じて処理結果をメモリ12に返す。

40

【0039】

CPU11は、分岐予測部21、フェッチ部22、デコード部23、演算部24、ライトバック部25、命令キャッシュ31、およびデータキャッシュ32を有している。

【0040】

1つの命令に対する実行処理は複数のステージ(工程)により構成される。つまり、命令キャッシュ31に保持された命令を読み出すフェッチ(IF)、フェッチされた命令をデコードし、制御信号を生成したり、レジスタ・ファイルレジスタ指定子で参照したり、必要に応じてデータキャッシュ32よりデータを読み出したりするデコード(ID)、数値の計算やロ

50

ード・ストアのデータやアドレス・分岐先の計算を行う命令実行(EX)、並びに、演算結果を、データキャッシュを介してメモリ12に戻すライトバック(WB)などのステージがある。

【0041】

命令キャッシュ31は、メモリ12より順次命令を読み出し保持する。フェッチ部22は、IFステージにおいて、命令キャッシュ31より命令をフェッチし、それをデコード部23に供給する。デコード部23は、IDステージにおいて、フェッチ部22より供給された命令をデコードし、必要に応じてデータキャッシュ32よりデータを取得し、それらを演算部24に供給する。

【0042】

演算部24は、EXステージにおいて、デコード部23より供給される情報に基づいて演算を行う。その演算結果はライトバック部25に供給される。また、演算部24は、演算結果によって次の命令(ジャンプ先のアドレス)が異なる分岐命令のライトバック部25は、WBステージにおいて、演算部24より供給された演算結果をデータキャッシュ32に保持させる。データキャッシュ32は、必要に応じて、その演算結果をデコード部23に供給したり、メモリ12に書き込ませたりする。

10

【0043】

分岐予測部21は、IFステージにおいて、演算部24より供給される演算結果等に基づいて、過去に実行された分岐命令の予測結果に応じて付与された重みによって重み付けされたグローバル履歴の分岐結果を用いて、複数の鎖および複数のステージを有するパイプライン構造の予測処理を行い、分岐命令の実行結果、すなわち分岐先(アドレス)を予測する(分岐命令に対して分岐予測を行う)。フェッチ部22は、その分岐予測部21の予測に従って次の命令をフェッチする。

20

【0044】

このとき、フェッチ部22、デコード部23、演算部24、およびライトバック部25は、パイプライン処理を行い、複数の命令を並列に実行する。すなわち、フェッチ部22は、演算部24の演算結果を待たずに、分岐予測部21の予測結果に従って、命令キャッシュ31より命令を次々とフェッチする。従って、分岐予測部21の予測精度を向上させることにより、CPU11において行われるパイプライン処理において制御ハザードの発生が減少するので、CPU11の処理性能を向上させることができる。

【0045】

ここで、上述のIFステージにおけるパイプライン構造についてより詳細に説明する。本パイプライン構造は、グローバル履歴84に含まれる各分岐結果を履歴順に分割してステージ毎に順次処理を行うパイプラインを、ステージの最終段を除いて、M個ずつ最新の履歴から順に同一ステージで同時並列的に処理するn個のグループに分割することにより、n個の長さを有しステージ毎に順次処理を行う鎖がM本形成されてなる(図8参照)。各鎖の各ステージにおいては、割り当てられたグローバル履歴84の分岐結果に重み付けをした重み付き分岐結果値を順次算出して鎖毎に累算していく処理が行われるようになっている。そして、最終段ステージ(ステージ0)では、各鎖において累算された重み付き分岐結果値の総和を求め、閾値を用いて予測値を算出する処理が行われるようになっている(図8参照)。

30

40

【0046】

次に、その分岐予測部21の構成について説明する。図6は、図5の分岐予測部21の詳細な構成例を示すブロック図である。

【0047】

図6において、分岐予測部21は、上述したパイプラインを用いて分岐予測を行う予測処理部41、分岐命令のアドレスに分岐命令ごとの分岐結果の履歴であるローカル履歴を反映させた多重化アドレスを生成する多重化アドレス生成部74、分岐予測に使用される重みテーブルの各重みについて演算結果に基づいて学習を行う学習処理部42、分岐予測に使用される各パラメータの更新を行う更新処理部43、および分岐予測に使用される各パラメータやデータを保持するレジスタ44を有している。

50

## 【 0 0 4 8 】

レジスタ44は、予測処理部41の分岐予測に使用されるパラメータやデータ等を必要に応じて保持する。レジスタ44には、例えば、重み付き分岐結果累算値81、重みテーブル82、多重化アドレス83、グローバル履歴84、実行パス履歴85、およびローカル履歴86が保持される。

## 【 0 0 4 9 】

グローバル履歴(GHR(p))84は、プログラム全体における分岐結果を時系列に並べた分岐結果の履歴(分岐履歴)である。GHR(p)は、グローバル履歴84における最新の分岐結果から(p+1)番目の分岐結果を示す(0 ≤ p < n(M-1))。各分岐結果は、分岐成立(Taken)を示す値「1」、若しくは、分岐不成立(Not Taken)を示す値「-1」により構成される。つまり、各分岐結果は分岐方向を示す。ただし、「-1」は実際には「0」で表現されている。

10

## 【 0 0 5 0 】

実行パス履歴(A[p])85は、そのグローバル履歴84の各分岐結果に対応する分岐命令のアドレスの履歴である。A[p]は、実行パス履歴85における最新のアドレスから(p+1)番目のアドレスを示す。つまり、グローバル履歴84の分岐結果GHR(p)は、実行パス履歴85のアドレスA[p]で示される分岐命令の分岐結果を示す。また、A[0]は、その時点での最新の分岐命令アドレスを示す。なお、最新の分岐結果GHR(0)によりジャンプする先のアドレスをターゲットアドレス(TA)とも称する。つまり、換言すると、古い(最新でない)ターゲットアドレス(TA)が実行パス履歴85に時系列順に蓄積される。

20

## 【 0 0 5 1 】

ローカル履歴(LHT(Local History Table)(q))86は、グローバル履歴84の各分岐結果を、分岐命令ごとにまとめたものである。例えば、LHT(q)は、ある分岐命令の、最新の分岐結果からq(0 ≤ q < p)番目の分岐結果を示す。つまり、ローカル履歴86においても、各分岐結果は分岐方向を示し、その値は、分岐成立(Taken)を示す値「1」、若しくは、分岐不成立(Not Taken)を示す値「-1」により構成される。

## 【 0 0 5 2 】

図7を参照して、グローバル履歴84乃至ローカル履歴86についてさらに詳細に説明する。

## 【 0 0 5 3 】

図7の左に示されるプログラム(sample.c)91は、グローバル履歴84乃至ローカル履歴86の説明のための、分岐命令を含むプログラムの例である。プログラム91においては、分岐命令として、200行目にfor文が記述され、208行目、216行目、および224行目にif文が記述されている。

30

## 【 0 0 5 4 】

図7の右側には、このプログラム91について、グローバル履歴84、実行パス履歴85、およびローカル履歴86の例が示されている。各履歴は、図7において、左から右に向かって時系列順に並んでいる。つまり、最も左側の履歴が一番古く、最も右側の履歴が一番新しい。実行パス履歴85の、最も古い(一番左の)アドレスは「200」であり、200行目のfor文が実行されたことが示されている。

40

## 【 0 0 5 5 】

このアドレスに対応するグローバル履歴84の分岐結果(GH)は「0」であり、200行目のfor文の条件(i<2)が真であったことが示されている。従って、このときのターゲットアドレス、すなわち次(実行パス履歴85の左から2番目)のアドレスは、for文のループ内の次の分岐命令が記述された「208」となる。このアドレスに対応するグローバル履歴84の分岐結果(GH)は「0」であり(すなわち、if文の条件(a==b)が偽であり)、変数xのインクリメント(x++)は行われずに、次の分岐処理の216行目に処理が移行する。従って、このときのターゲットアドレス、すなわち次(実行パス履歴85の左から3番目)のアドレスは、次の分岐命令が記述された「216」である。このアドレスに対応するグローバル履歴84の分岐結果(GH)は「1」であり(すなわち、if文の条件(a==b)が真であり)、変数yがインクリメント(

50

y++)される。その後、処理は200行目に戻る(実行パス履歴85においては左から4番目に進む)。

【 0 0 5 6 】

以降も同様にして各命令が実行され、200行目のfor文の条件( $i < 2$ )が偽となるまでfor文のループ内の命令(208行目および216行目の分岐命令)が繰り返し実行される。

【 0 0 5 7 】

そして、実行パス履歴85の最新の(一番右の)アドレスで指定された200行目のfor文の予測において、分岐結果が「1」(グローバル履歴84の一番右の分岐結果(GH))であること、すなわち、200行目のfor文のループが終了されたことと予測されたことが示されている。この場合、224行目のif文にジャンプすることになるので、このときのターゲットアドレス(TA)87の値は、図7に示されるように「224」となる。

10

【 0 0 5 8 】

なお、ローカル履歴86の各分岐結果(LH)は、グローバル履歴に示される分岐結果が、分岐命令毎に(行番号毎に)整理されている。

【 0 0 5 9 】

図6及び図10を参照して、多重化アドレス(MA(Multiplex Address)(p))83は、実行パス履歴85のアドレス(A[p])の下位ビットをそのビット長さ分のローカル履歴86(LHT(q))に差し替えて生成される。MA[p]は、実行パス履歴85における最新のアドレスから(p+1)番目のアドレスA[p]の下位ビットをローカル履歴86LHT(q)に差し替えたものを示す。なお、本実施形態においては、多重化アドレス(MA(p))83は、実行パス履歴85のアドレス(A[p])の下位ビットをそのビット長さ分のローカル履歴86(LHT(q))に差し替えて生成されるようにしたが、アドレス(A[p])の下位ビットを削除することなく、アドレス(A[p])にローカル履歴86(LHT(q))を付加させることにより、アドレス(A[p])にローカル履歴86(LHT(q))を反映させるようにしてもよい。

20

【 0 0 6 0 】

ここで、予測精度の向上のために、ローカル履歴を利用することについて説明する。ローカル履歴は、例えば、図4を参照して説明したようなグローバル/ローカルパーセプトロン分岐予測器等で使用されており、このローカル履歴の分岐結果を利用することにより、グローバル履歴の分岐結果のみを用いた場合よりも予測精度が向上している。

【 0 0 6 1 】

しかしながら、ローカル履歴の分岐結果に重み付けを行って分岐予測をする場合、ターゲットアドレス以外を用いて重みを読み出すとすると、分岐結果と無関係な重み付けがなされてしまうため、重みはターゲットアドレスを用いて読み出さなくてはならず、パイプライン化ができなくなる。

30

【 0 0 6 2 】

そこで、本実施形態においては、図10に示すように、実行パス履歴85のアドレス131の下位2ビットを削除してアドレス(ショート)132を生成し、そのアドレス(ショート)132にローカル履歴86(LHT)から読み出された2ビットの分岐結果(LH)133を付加したものを多重化アドレス(MA)83とし、この多重化アドレス83をレジスタ44に記憶しておき、ステージnからステージ1における重みテーブル82からの重みの読み出し及びステージ0における閾値の読み出しに利用している。

40

【 0 0 6 3 】

このように、本実施形態においては、ローカル履歴86の分岐結果は直接的に予測演算に使用されることはないが、インデックスに加えることで1つのアドレスからローカル履歴長のべき乗通りのインデックスが作成可能になる。一般的に1つの分岐命令が、その時々に応じて分岐方向を変える(分岐方向の両方に分岐する)ことは多いが、実行パス履歴85のアドレスのみをインデックスとして重みを選択する従来の手法の場合、その分岐命令がいずれの方向に分岐する場合であっても同じ重みを選択されることになっていた(重みの負の衝突が発生した)。

【 0 0 6 4 】

50

これに対し、図11に示されるように、ローカル履歴86の分岐結果を実行パス履歴85のアドレスの下位ビットと差し替えた多重化アドレス(MA)をインデックスとして使用することにより、選択する重みを、その分岐方向によって変更することができ、重みの負の衝突を回避することができる。これにより、ハードウェアコストの増加を最小限に抑えつつ予測精度の向上を期待することができる。

【0065】

図6を参照して、重みテーブル(WT(i, j))82は、各分岐命令について、過去に当該分岐命令まで辿ったグローバル履歴の各分岐結果に対して、事前になされた分岐予測の正否を反映した重み(重み群)が、当該分岐命令のアドレスの下位ビットをローカル履歴に差し替えた多重化アドレスと対応付けられて、保持されたものである。多重化アドレス毎に重みに対応付けられているため、1つの分岐命令の両分岐方向について、それぞれ分岐予測の正否に応じて付与された重みが保持されている。重みテーブル82は、多重化アドレス毎に、その分岐命令までのグローバル履歴に含まれる各分岐結果に対する重みの全てが重み群として保持されたものであるが、WT(i, j)は、j(1 ≤ j ≤ M)番目の鎖のi(1 ≤ i ≤ n)番目のステージに割り当てられたグローバル履歴がGHR(p)とすると、全多重化アドレスについて最新の分岐結果から(p+1)番目の分岐結果に対応する重みが保持された部分を重みテーブルとして示すものである(i, jは整数)。

10

【0066】

重み付き分岐結果累算値(REG(i, j))81は、各鎖の各ステージに割り当てられたグローバル履歴の各分岐結果に重み付けを行った重み付き分岐結果値を、各ステージにおいて鎖毎に順次累算した値である。REG(i, j)は、j(1 ≤ j ≤ M)番目の鎖のi(1 ≤ i ≤ n)番目のステージにおいて算出される累算値であり、その鎖のステージnからステージi(1 ≤ i ≤ n)までの各ステージにおいて算出された、重み付き分岐結果値の累算値を示す(i, jは整数)。この累算値は、各ステージで算出される度にレジスタ44に一旦保持された後、次のステージにおいて読み出され、新たな累算値の算出に利用される。

20

【0067】

図6のレジスタ44には、このような各種のパラメータが必要に応じて保持される。

【0068】

図6の予測処理部41は、レジスタ44に保持される、このようなデータやパラメータ等を用いて、複数(n+1段)のステージおよび複数(M個)の鎖を有するパイプライン構造の予測処理を行う。なお、ステージ0は、各鎖共通である。予測処理部41は、このようなパイプライン構造の予測処理のうち、ステージnからステージ1の処理を行うステージ1-n処理部51、および、ステージ0の処理を行うステージ0処理部52を有している。

30

【0069】

ステージ1-n処理部51は、各鎖のステージn乃至ステージ1の各ステージにおいて、割り当てられた重みテーブル82(WT(i, j))から重みを選択する重み選択部61、及び、その選択された重みによりグローバル履歴の分岐結果に重み付けをした重み付き分岐結果値を算出し、鎖毎に累算処理を行う重み付き分岐結果値累算部62を有する。

【0070】

重み選択部61は、1本目の鎖(図8において一番上の段の鎖)のステージ1においては、割り当てられた重みテーブル82(WT(1, 1))より、実行パス履歴85の最新のアドレスA[0]をインデックスとして重みを読み出す。1本目の鎖のステージ1以外のステージ(n~2)及び他の鎖(2~M本目の鎖)の各ステージ(n~1)においては、それぞれ割り当てられた重みテーブル82より、実行パス履歴85の最新のアドレスA[0]と、割り当てられたグローバル履歴84に対応する多重化アドレス(MA)との排他的論理和をインデックスとして、それぞれ重みを選択して読み出す。

40

【0071】

重み付き分岐結果値累算部62は、各鎖の各ステージにおいて、割り当てられたグローバル履歴84の分岐結果に重み付けをした重み付き分岐結果値を算出するようになっている。実際には、分岐不成立(Not Taken)を示す値「-1」が「0」で表現されており、「0」に重

50

みを掛け合わせると「0」になってしまうため、分岐結果が「0」である場合は重みの符号を反転させた値を、分岐結果が「1」である場合には重みそのままの値を出力させることにより、重み付き分岐結果値を求めるようになっている。そのため、重み付き分岐結果値累算部62は、例えば、重みの値(ビットパターン)を反転させるための論理否定回路(図示しない)と、分岐結果の値を反転させるための論理否定回路(図示しない)と、読み出した重みの値とこれを論理否定回路により反転させた値とを入力して分岐結果の反転値を用いてどちらかを選択して出力するセレクタ(図示しない)と、セレクタから出力された値と分岐結果の反転値とを加算する加算器を備え、セレクタに分岐結果「0」を反転した「1」が入力されたときは、重みを反転した値(重みの値の1の補数)を選択して出力し、重みを反転した値と分岐結果の反転値「1」とを加算して2の補数(負の表現)として出力し、セレクタに分岐結果「1」を反転した「0」が入力されたときは、重みそのままの値を選択して出力し、その値と分岐結果の反転値「0」とを加算して重みそのままの値を出力するようになっている。

10

## 【0072】

また、重み付き分岐結果値累算部62は、加算器102(図8参照)を有しており、各ステージにおいて、求めた重み付き分岐結果値に、レジスタ44に保持されている直前のステージまでの重み付き分岐結果累算値81を読み出して加算することによって、そのステージまでの重み付き分岐結果累算値81を算出し、算出した新たな重み付き分岐結果累算値81をレジスタ44に保持させるようになっている。なお、加算器102は、上記セレクタ出力値と分岐結果の反転値とを加算する加算器としても用いるようにしても良い。

20

## 【0073】

ステージ0処理部52は、ステージ0において、分岐成立または不成立を識別するための閾値を算出する閾値算出部72と、各鎖の重み付き分岐結果累算値の総和を算出し、算出した総和と閾値とに基づいて予測値を算出する予測値算出部73とを有する。

## 【0074】

閾値算出部72は、ステージ0に割り当てられた重みテーブル82から、最新のアドレスA[0](ステージ0においてはターゲットアドレス)における多重化アドレスを生成する際にローカル履歴に差し替えらる下位ビットを除く上位ビットをインデックスとして、特定される複数の閾値を候補として全て選定する。閾値の候補数は、多重化アドレスを生成する際にローカル履歴に差し替えらる下位ビットの数のべき乗であり、例えば、ローカル履歴に差し替えらるビット数が2である場合は、閾値の候補数は4となる。また、閾値算出部72は、セレクタ112を有しており(図8参照)、ローカル履歴86のテーブルより分岐結果を読み出して、複数の閾値候補の重みの中から1つを選択することにより、閾値を算出する。なお、このように最新のアドレスA[0]の上位ビットで特定した閾値の候補からローカル履歴で1つを選択することによって、結果として多重化アドレスを用いて閾値を算出したことになる。

30

## 【0075】

予測値算出部73は、加算器111を有しており(図8参照)、予測処理がパイプライン構造のステージnからステージ1まで進む間に各鎖において累算された重み付き分岐結果累算値81をそれぞれレジスタ44より取得し、それらの総和を算出する。また、予測値算出部73は、加算器113を有しており(図8参照)、重み付き分岐結果累算値の総和と閾値算出部72により算出された閾値とを加算することによって予測値を算出する。例えば、予測値算出部73は、重み付き分岐結果累算値の総和と閾値の加算結果の最上位ビット(符号ビット)の値が「1(-を意味)」の場合は、過去において分岐不成立の傾向が強いことを示しているため、分岐不成立を予測して予測値「0」を出力し、最上位ビットの値が「0(+を意味)」の場合は、過去において分岐成立の傾向が強いことを示しているため、分岐成立を予測して予測値「1」を出力する。この場合、予測値算出部73は、論理否定回路(図示しない)を備え、重み付き分岐結果累算値の総和と閾値の加算結果の最上位ビットを反転させ、その値を予測値として出力するようになっている。なお、重み付き分岐結果累算値の総和と閾値の加算結果の絶対値は、その予測値の確かさ(信頼度)とみなすことができる。

40

50

## 【 0 0 7 6 】

多重化アドレス生成部74は、実行パス履歴85のアドレス(処理時はターゲットアドレス)の下位2ビットを、ローカル履歴86に含まれる最新の分岐結果(ターゲットアドレスの1つ前の分岐命令に対する分岐結果)及びその前の分岐結果に差し替えて多重化アドレス(MA)を生成する。

## 【 0 0 7 7 】

学習処理部42は、予測値算出部73において算出された予測値と、演算部24における実際の演算結果(分岐結果)に基づいて、重みテーブル82の各重みについての学習処理を行う。更新処理部43は、新たなターゲットアドレス、演算結果(分岐結果)が確定した後、実行パス履歴85、グローバル履歴84、ローカル履歴86を随時更新する処理を行う。なお、更新処理部43は、予測値が算出される毎にグローバル履歴84を暫定的に更新し、演算結果が算出された後に、予測値を演算結果が異なる場合は、演算結果に書き換えるようになっている。また、更新処理部43は、新たなターゲットアドレスとローカル履歴86とによりアドレス83が生成された後、多重化アドレス83を随時更新する処理を行う。

## 【 0 0 7 8 】

図8は、予測処理部41により実行される予測処理のパイプライン構造(Advanced Anti-Aliasing Perceptron Branch Predictor(A<sup>3</sup>PBPまたはAAAPBP)と称するパーセプトロン分岐予測器の実行する予測処理の構造)の構成例を模式的に示す図である。

## 【 0 0 7 9 】

本パイプライン構造は、図8に示すように、 $nM - n + 1$ の長さを有するグローバル履歴84に含まれる各分岐結果を履歴順に分割してステージ毎に順次処理を行うパイプラインを、ステージの最終段(ステージ0)を除いて、 $GHR[0] \sim GHR[M-1]$ のM個の分岐結果に関する処理を、同時並列的にステージ1で行う1つのグループとし、同様に $GHR[M-1] \sim GHR[2M-2]$ のM個の分岐結果に関する処理を、同時並列的にステージ2で行う1つのグループとするというように、M個ずつ分割されたグループがn個あり、n個の長さを有しステージnからステージ1に向かって順次処理を行う鎖がM本形成されてなる。

## 【 0 0 8 0 】

また、各鎖の各ステージには、割り当てられたグローバル履歴に対応する重みテーブル82が割り当てられている。具体的には、1本目の鎖のステージ1には、最新の分岐結果から0番目のグローバル履歴 $GHR[0]$ と、グローバル履歴 $GHR[0]$ の分岐結果に対する重みが保持された重みテーブルとが割り当てられており、2本目の鎖のステージ1には、最新の分岐結果から1個前のグローバル履歴 $GHR[1]$ と、グローバル履歴 $GHR[1]$ の分岐結果に対する重みが保持された重みテーブル82とが割り当てられ、M本目の鎖のステージ1には、最新の分岐結果からM-1個前のグローバル履歴 $GHR[M-1]$ と、グローバル履歴 $GHR[M-1]$ の分岐結果に対する重みが保持された重みテーブル82とが割り当てられている。同様に他のステージにおいても1本目の鎖から履歴が新しい順に1つずつグローバル履歴 $[p]$ とグローバル履歴 $GHR[p]$ の分岐結果に対する重みが保持された重みテーブル82とが割り当てられている。

## 【 0 0 8 1 】

このように本実施形態においては、最新の分岐結果に近いグローバル履歴84 $GHR[p]$ 及びターゲットアドレスに近い実行パス履歴85のアドレスほど後のステージで使用されるパイプライン構造としたため、それぞれのステージにおいて処理を行う際に、演算結果に基づいて更新されたグローバル履歴84 $GHR[p]$ 及びよりターゲットアドレスに近いアドレスの更新された実行パス履歴85A $[0]$ を用いることができるため、予測精度が向上する。すなわち、図3に示すように、パイプラインを履歴順に所定長を有する複数の鎖に分割したパイプライン構造では、最上流のステージにおいては次の段の鎖の最下流のステージよりも最新の分岐結果に近いグローバル履歴84 $GHR[p]$ 及びターゲットアドレスに近い実行パス履歴85のアドレスが割り当てられていることから、上方段の鎖の上流側のステージでは、その処理時に、割り当てられたグローバル履歴84 $GHR[p]$ の演算結

10

20

30

40

50

果(分岐結果)が算出されていない場合があり、演算結果に書き換えられる前に暫定的に格納されている分岐予測結果を用いて投機的に予測を行うことになるため予測精度が低下するが、本実施形態によるパイプライン構造では、分岐予測結果を用いて投機的に予測を行わずにすむため予測精度が向上する。

【 0 0 8 2 】

そして、本パイプライン構造による処理は、次のように行われる。すなわち、ステージ  $n$  の上から1本目の鎖において、重み選択部61により、実行パス履歴85のこのステージ  $n$  の処理時に取得可能な最新のアドレス( $A[0]$ )と多重化アドレス( $MA[(M-1)(n-1)]$ )とが読み出され、両者の排他的論理和をインデックスとして、重みテーブル82 -  $\{(n-1)M+1\}$  より重みを選択される。そして、重み付き分岐結果値累算部62により、選択された重みを用いて、グローバル履歴84GHR  $[(M-1)(n-1)]$  に重み付けがなされ、その結果が重み付き分岐結果累算値81としてレジスタ44 -  $n$  に一時的に保持される。

10

【 0 0 8 3 】

また、ステージ  $n$  の上から2本目の鎖において、重み選択部61により、実行パス履歴85のこのステージ  $n$  の処理時に取得可能な最新のアドレス( $A[0]$ )と多重化アドレス( $MA[nM-n-M+2]$ )とが読み出され、両者の排他的論理和をインデックスとして、重みテーブル82 -  $\{(n-1)M+2\}$  より重みを選択されされる。そして、重み付き分岐結果値累算部62により、選択された重みを用いて、グローバル履歴84GHR  $[nM-n-M+2]$  に重み付けがなされ、その重み付き分岐結果値が重み付き分岐結果累算値81としてレジスタ44 -  $n$  に一時的に保持される。

20

【 0 0 8 4 】

同様にステージ  $n$  の各鎖において、実行パス履歴85の最新のアドレス( $A[0]$ )と多重化アドレス( $MA[p]$ )の排他的論理和に基づいて重みテーブル82より重みを選択され、グローバル履歴84に重み付けがなされ、その重み付き分岐結果値が重み付き分岐結果累算値81としてレジスタ44 -  $n$  に一時的に保持される。このステージ  $n$  の  $M$  個の各鎖における処理は、同時並列的に行われる。

【 0 0 8 5 】

ステージ  $n-1$  乃至ステージ1の各鎖においては、重み付き分岐結果値累算部62により、図8において左隣(時系列的に上流側)の、1段前のステージ(例えば、ステージ  $(n-1)$  の1段前のステージはステージ  $n$ 、ステージ1の1段前のステージはステージ2)においてレジスタ44(レジスタ44 -  $n$  乃至レジスタ44 - 2)に保持された重み付き分岐結果累算値81が読み出され、ステージ  $n$  の場合と同様に算出した重み付き分岐結果値に加算され、新たな重み付き分岐結果累算値81としてレジスタ44(レジスタ44 -  $(n-1)$  乃至レジスタ44 - 1)に一時的に保持される。

30

【 0 0 8 6 】

例えば、ステージ2の上から1本目の鎖において、重み選択部61により、実行パス履歴85のこのステージ2の処理時に取得可能な最新のアドレス( $A[0]$ )と多重化アドレス( $MA[M-1]$ )とが読み出され、両者の排他的論理和をインデックスとして、重みテーブル82 -  $(M+1)$  より重みを選択される。そして、重み付き分岐結果値累算部62により、選択された重みを用いて、グローバル履歴84GHR  $[M-1]$  に重み付けがなされ、その重み付き分岐結果値が、レジスタ44より読み出され、同じ鎖のステージ3までの重み付き分岐結果累算値81( $REG(3,1)$ )と加算される。そして、重み付き分岐結果値累算部62により、その加算結果が新たな重み付き分岐結果累算値81として、レジスタ44 - 2に一時的に保持される( $REG(2,1)$ )。ステージ2の他の鎖においても、同様の処理が同時並列的に行われる。

40

【 0 0 8 7 】

ステージ1においても同様の処理がなされるが、ステージ1の上から1本目の鎖の場合、実行パス履歴85の最新のアドレス( $A[0]$ )のみによって重みを選択される。これは、1本目の鎖において多重化アドレス( $MA[0]$ )を用いようとする、この多重化アドレス( $MA[0]$ )は最新のアドレス( $A[0]$ )が確定した後に生成されるため、最新のアドレス( $A[0]$ )が取得可能であるにもかかわらず、多重化アドレス( $MA[0]$ )の生成を待つ両者の排他的論

50



理和を算出しなければならなくなり、不要にレイテンシを増やすことになるためである。

【0088】

本パイプライン処理においては、 $x$ を任意の自然数とし、あるステージのある鎖に割り当てられたグローバル履歴が $GHR[x]$ とすると、 $GHR[x]$ の分岐結果に対応する実行パス履歴のアドレスを $A[x]$ として、重み選択に用いるインデックスを生成するための多重化アドレス( $MA[p]$ )は、 $A[x]$ から生成されたものを用いるようにしたが、任意の多重化アドレスを用いるようにしてもよい。例えば、 $GHR[0]$ が割り当てられたステージ1の1本目の鎖では $MA[1]$ を用い、 $GHR[1]$ が割り当てられたステージ1の2本目の鎖では $MA[2]$ を用いるというように、順次対応付けるようにしてもよい。ただし、この対応関係は予測処理の開始前に予め決められている。

10

【0089】

このように、本実施形態では、予測に関係のない分岐命令のアドレスの悪影響を最小限に抑えるために、パイプラインPTBP(図3参照)のように、鎖の長さ $N$ 毎に連鎖的にアドレス履歴の排他的論理和を取って重み読み出しのインデックスにすることはせず、最新のアドレス( $A[0]$ )のみ、あるいは最新のアドレス( $A[0]$ )と多重化アドレス( $MA[p]$ )とのみの排他的論理和をインデックスとするようにしている。

【0090】

本パイプライン処理においては、新たな分岐命令が命令キャッシュ31に入力される毎に、ステージが1つずつ進行するようになっている。

【0091】

なお、以上において、レジスタ44-1乃至レジスタ44-nは、いずれも図6のレジスタ44の記憶領域の一部を示しており、それぞれ、各ステージにおいて算出される重み付き分岐結果累算値を一時的に保持する領域である。

20

【0092】

ここで、図8に示される $A^3PBP$ の各ステージでレジスタ44から読み出すことのできる重み付き分岐結果累算値の最大数すなわち鎖の本数について説明する。

【0093】

$A^3PBP$ のレイテンシは、閾値が読み出されるレイテンシと、ステージ1までの鎖毎の重み付き分岐結果累算値と閾値の加算に必要なレイテンシを足した値である。実装するプロセッサのアーキテクチャに依存することではあるが、閾値の読み出し時間で隠蔽することが十分可能である加算器の段数を3段とすれば、重み付き分岐結果累算値の総和を求めるのを4段の加算器で行うのであれば、閾値の読み出し時間と同等で処理できる。つまり、同時に読み出し可能な重み付き分岐結果累算値の最大数は16となる。

30

【0094】

このことを踏まえて、このパーセプトロン分岐予測器のパイプライン段数の最適化実験を行った。実験はSPEC CINT2000の12種類のベンチマークをSimCoreシミュレータで1億命令程度実行した際にトレースしたデータを、この予測器が実装された場合の動作をシミュレートしたプログラムで処理し、その予測精度を求めた。

【0095】

図9はステージ数( $h+1$ )による予測ミス率の変化を示している。グローバル履歴(bit数)が短い場合には $h=1$ (ステージ数最小)が最も良い結果を出しているのに対し、グローバル履歴が長くなるにつれてステージ数(つまり $h$ )の深さによる予測ミス率の差は小さくなっている。例えば、グローバル履歴長が64ビットの場合、 $h=1$ の予測ミス率よりも、 $h=2$ の予測ミス率の方が低くなっている。

40

【0096】

これは、グローバル履歴が長くなるにしたがい、PTBPのようなパスの詳細な把握による予測精度の向上効果が、ステージの深化によるターゲットアドレスに関する情報の劣化による予測精度低下の影響を上回るためだと思われる。

【0097】

図9に示されるように、予測ミス率は、グローバル履歴長が16ビットの場合、 $h=1$ (す

50

なわち同時重み付き分岐結果累算値読み出し数が16)のときが一番低く、 $h = 2$ (すなわち同時重み付き分岐結果累算値読み出し数が8)のときがその次に低い。また、グローバル履歴長が32ビットの場合、予測ミス率は、 $h = 1$ (すなわち同時重み付き分岐結果累算値読み出し数が32)のときと $h = 2$ (すなわち同時重み付き分岐結果累算値読み出し数が16)のときが略同程度で最も低く、 $h = 3$ (すなわち同時重み付き分岐結果累算値読み出し数が8)のときがその次に低い。さらに、グローバル履歴長が48ビットの場合、予測ミス率は、 $h = 1$ (すなわち同時重み付き分岐結果累算値読み出し数が48)のときと $h = 2$ (すなわち同時重み付き分岐結果累算値読み出し数が24)のときが略同程度で最も低く、 $h = 3$ (すなわち同時重み付き分岐結果累算値読み出し数が12)のときがその次に低い。また、グローバル履歴長が64ビットの場合、予測ミス率は、 $h = 2$ (すなわち同時重み付き分岐結果累算値読み出し数が32)のときが最も低く、 $h = 1$ (すなわち同時重み付き分岐結果累算値読み出し数が64)のときと $h = 3$ (すなわち同時重み付き分岐結果累算値読み出し数が16)のときが略同程度でその次に低い。

【0098】

以上の結果より、最適な同時重み付き分岐結果累算値読み出し数は16乃至32とすることができる。上述したように、重み付き分岐結果累算値の許容最大数は16であるので、最大の予測精度を得るためには、同時に読み出す重み付き分岐結果累算値の数すなわち鎖の本数を16とするのが最適であるといえる。

【0099】

次に、ステージ0について説明する。ステージ0では、閾値算出部72により、重みテーブル82から、最新のアドレスA[0]の上位ビットをインデックスとして、特定される複数の重みが閾値候補として全て選定されるとともに、最新のアドレスA[0]をインデックスとしてローカル履歴86のテーブルより最新の分岐結果とその1つ前の分岐結果とが読み出され、読み出した分岐結果を用いて、閾値算出部72を構成するセクタ112により、閾値候補の中から1つが選択されることにより閾値が算出される。なお、重みの選択とローカル履歴からの分岐結果の読み出しとは並行して行われる。また、予測値算出部73により、ステージnからステージ1において鎖ごとに算出されたM個の重み付き分岐結果累算値81(REG(1, 1)乃至REG(1, M))がそれぞれレジスタ44より読み出され、予測値算出部73を構成する加算器111により、それらの総和が算出される。そして、予測値算出部73を構成する加算器113により、重み付き分岐結果累算値の総和と閾値とを加算することによって予測値が算出される。

【0100】

なお、ステージ0ではM個の重み付き分岐結果累算値81の総和を算出する必要があるが、閾値の算出と同時並行的に計算させることにより、その計算時間を隠蔽することができる。

【0101】

ここで、多重化アドレスを生成する際に差し替えられるローカル履歴86の分岐結果のビット長を示すローカル履歴長と予測精度の関係について説明する。SPEG CINT2000の12種類のベンチマークをSimCoreシミュレータで1億命令程度実行した際にトレースしたデータを、この予測器が実装された場合の動作をシミュレートしたプログラムで処理し、その予測精度を求めた。

【0102】

図12は、その実験結果を示す、実行パス履歴85のアドレスに付加する分岐結果の長さ(LHビット長(bit))と予測ミス率(%)の関係を示すグラフである。図12のグラフにおいて、曲線141は、グローバル履歴の長さが32ビットの場合の、LHビット長と予測ミス率との関係を示しており、曲線142は、グローバル履歴の長さが48ビットの場合の、LHビット長と予測ミス率との関係を示している。図12のグラフより、ローカル履歴長が3ビットの場合に最も予測ミス率が低いことが分かる。ただし、2ビットの場合とほとんど差はなく、逆に、ビット長が増えるとステージ0において閾値候補として一括読み出しが必要な重みの数が増え、処理の負荷が大きくなるデメリットが大きくなる。2ビットの場合、一括読み

10

20

30

40

50

出しの必要がある閾値の個数は4個であり、実装上特に障害とはならない。なお、もちろん、この分岐結果の長さは任意であり、処理の負荷が、実装上問題ない程度であれば、例えば3ビット以上にすることも可能である。

【0103】

次に、分岐予測処理の全体の流れについて、図13のフローチャートを参照して説明する。

【0104】

予測処理部41のステージ1 - n 処理部51(図6)は、ステップS1において、分岐命令が命令キャッシュ31(図5)に入力されたか否かを判定し、入力されていないと判定した場合、ステップS2に処理を進め、分岐予測パイプラインとしての処理をストールさせ、処理をステップS1に戻す。すなわち、ステージ1 - n 処理部51は、ステップS1において、分岐命令が入力されたと判定するまで、ステップS2の処理を繰り返しながら待機する。

10

【0105】

ステップS1において、分岐命令が入力されたと判定した場合、ステージ1 - n 処理部51は、処理をステップS3に進める。ステップS3において、ステージ1 - n 処理部51は、図8に示されるA<sup>3</sup>PBPのパイプラインのステージnについて重み付き分岐結果累算処理を行い、算出した重み付き分岐結果累算値81(REG(n, j))をレジスタ44 - nに保持させる。ステップS3の処理を終了すると、ステージ1 - n 処理部51は、処理をステップS4に進める。

【0106】

ステップS4において、ステージ1 - n 処理部51は、次の分岐命令が命令キャッシュ31に入力されたか否かを判定し、次の分岐命令が入力されたと判定するまで、ステップS5において分岐予測パイプラインとしての処理をストールさせながら待機する。

20

【0107】

ステップS4において、次の分岐命令が入力されたと判定した場合、ステージ1 - n 処理部51は、処理をステップS6に進め、図8に示されるA<sup>3</sup>PBPのパイプラインのステージ(n - 1)について重み付き分岐結果累算処理を行い、算出した重み付き分岐結果累算値81(REG((n - 1), j))をレジスタ44 - (n - 1)に保持させる。ステップS6の処理を終了すると、ステージ1 - n 処理部51は、処理を次のステップに進める。

【0108】

以上のように、ステージ1 - n 処理部51は、分岐命令が命令キャッシュ31に入力される度に、各ステージについて重み付き分岐結果累算処理を繰り返し、ステージ1まで重み付き分岐結果累算値を算出する。

30

【0109】

そしてステージ1について、ステップS(3n - 2)において、ステージ1 - n 処理部51は、分岐命令が命令キャッシュ31に入力されたか否かを判定し、次の分岐命令が入力されたと判定するまで、ステップS(3n - 1)において分岐予測パイプラインとしての処理をストールさせながら待機する。

【0110】

ステップS(3n - 2)において、次の分岐命令が入力されたと判定した場合、ステージ1 - n 処理部51は、処理をステップS3nに進め、図8に示されるA<sup>3</sup>PBPのパイプラインのステージ1について重み付き分岐結果累算処理を行い、算出した重み付き分岐結果累算値81(REG(1, j))をレジスタ44 - 1に保持させる。ステップS3nの処理を終了すると、ステージ1 - n 処理部51は、処理をステップS(3n + 1)に進める。

40

【0111】

ステージ0処理部52は、ステップS(3n + 1)乃至ステップS(3n + 3)において、図8に示されるA<sup>3</sup>PBPのパイプラインのステージ0についての処理を行う。つまり、ステップS(3n + 1)において、ステージ0処理部52は、新たな分岐命令が命令キャッシュ31に入力されたか否かを判定し、次の分岐命令が入力されたと判定するまで、ステップS(3n + 2)において分岐予測パイプラインとしての処理をストールさせながら待機する。

50

## 【 0 1 1 2 】

ステップ S (3n + 1)において、新たな分岐命令が入力されたと判定した場合、ステージ0処理部52は、処理をステップ S (3n + 3)に進め、図8に示されるA<sup>3</sup>PBPのパイプラインのステージ0についてステージ0処理を行い、重み付き分岐結果累算値の総和を算出し、閾値を選択し、その総和と閾値を加算し、予測値を算出して出力する。また、ステップ S (3n + 3)と並行して、ステップ (3n + 5)において、多重化アドレス生成部74が、ステップ S (3n + 1)において入力された分岐命令のアドレスすなわちターゲットアドレスとローカル履歴とにより多重化アドレスを生成する。

## 【 0 1 1 3 】

その後、ステップ S (3n + 4)において、レジスタの各値が更新されて、処理が終了される。具体的には、ステージ0におけるターゲットアドレスで実行パス履歴が更新され、ステップ S (3n + 4)において算出された予測値でグローバル履歴84が暫定的に更新される。また、ステップ S (3n + 5)において生成された多重化アドレス83による更新がされる。なお、その後、この分岐予測処理とは独立して、別途演算部24により実際に演算された分岐結果に基づいて、グローバル履歴84及びローカル履歴が更新される。また、分岐結果に基づいて学習処理部42により学習処理がなされた重みにより重みテーブル82が更新される。

## 【 0 1 1 4 】

このように、A<sup>3</sup>PBPを用いた分岐予測処理においては、ステージ n において初期値が生成され、分岐命令が入力される度にステージを1つずつ進み、重み付き分岐結果累算値が算出される。ただし、実際には、各分岐命令に対して予測値を出力する必要があるので、各ステージの処理は並行して行われる。つまり、図14に示すように、複数のパイプラインが並行して実行され、その結果、n組の重み付き分岐結果累算値が並行して処理され、分岐命令が入力される度に予測値が出力される。

## 【 0 1 1 5 】

次に、ステージ0処理の流れの例を図15のフローチャートを参照して説明する。

## 【 0 1 1 6 】

ステージ0処理が開始されると、予測値算出部73は、ステップ S 121において、レジスタ44 - 1に保持されている重み付き分岐結果累算値81(REG(1, j))を全て読み出し、加算器11を用いて、それらの総和を算出する。この処理と並行して、閾値算出部72は、ステップ S 122において、実行パス履歴85の最新のアドレス A [0] をインデックスとしてローカル履歴86より分岐結果(LHT(A [0]))を読み出す。

また、閾値算出部72は、さらにステップ S 122の処理と並行して、ステップ S 123において、実行パス履歴85の最新のアドレス A [0] の上位ビット(下位2ビットを削除したもの)をインデックスとして閾値用の重みテーブル82 - 0より閾値の候補(WT [0, A [0]])を全て(4個)取得する。

## 【 0 1 1 7 】

ステップ S 124において、閾値算出部72は、ローカル履歴86より読み出した分岐結果に基づいて、セレクト112により候補の中から閾値を選択する。予測値算出部73は、ステップ S 125において、加算器113を用いて重み付き分岐結果累算値の総和と閾値を加算し、ステップ S 126において、加算結果の最上位ビットの値を反転させ、ステップ S 127において、その反転結果(最上位ビット)を予測値として出力する。

## 【 0 1 1 8 】

次に、図16のフローチャートを参照して、ステージ n からステージ1における処理の詳細な流れの例を説明する。

## 【 0 1 1 9 】

ステージ1 - n 処理部51は、ステップ S 151において、図8におけるパイプラインの鎖の上からの番号を示す変数 j の値を「1」に設定し、ステップ S 152において、ステージ数を示す変数 k の値が「1」であるか否かを判定する。つまり、ステージ1 - n 処理部51は、今回のステージ k 処理がステージ1に対する処理であるか否かを判定する。

変数 k の値が「1」であると判定した場合、ステージ1 - n 処理部51は、処理をステップ S

10

20

30

40

50

153に進め、変数  $j$  の値が「1」であるか否かを判定し、「1」であると判定した場合、処理をステップ S 154に進め、図8においてステージ1の上から1本目の鎖において重みを累算する第1重み付き分岐結果累算処理を行う。第1重み付き分岐結果累算処理が終了すると、ステージ1 -  $n$  処理部51は、処理をステップ S 156に進める。また、ステップ S 153において、変数  $j$  の値が「1」でないと判定した場合、ステージ1 -  $n$  処理部51は、処理をステップ S 155に進め、図8において上から2乃至  $n$  本目の鎖のステージ1において重みを累算するために、ステージ1の上から1本目の鎖、並びに、ステージ  $n$  の各鎖以外において重みを累算する第3重み付き分岐結果累算処理を行う。第3重み付き分岐結果累算処理が終了すると、ステージ1 -  $n$  処理部51は、処理をステップ S 156に進める。

【 0 1 2 0 】

10

ステップ S 156において、ステージ1 -  $n$  処理部51は、変数  $j$  の値を「1」インクリメントし、処理対象を次の鎖に切り返る。ステップ S 157において、ステージ1 -  $n$  処理部51は、変数  $j$  の値が任意の正の整数  $M$  より大きいか否かを判定し、変数  $j$  の値が  $M$  より大きくなく、変数  $j$  が示す鎖は存在すると判定した場合、処理をステップ S 153に戻し、それ以降の処理を繰り返す。つまり、ステージ1 -  $n$  処理部51は、ステージ1の鎖のそれぞれについて、ステップ S 153乃至ステップ S 157の処理を実行する。そして、ステップ S 157において変数  $j$  の値が  $M$  より大きいと判定した場合、ステージ1 -  $n$  処理部51は、処理を終了する。

【 0 1 2 1 】

また、図16のステップ S 152において、変数  $k$  の値が「1」でないと判定した場合、すなわち、処理対象のステージがステージ2乃至ステージ  $n$  のいずれかである場合、ステージ1 -  $n$  処理部51は、処理をステップ S 158に進める。

20

【 0 1 2 2 】

ステップ S 158において、ステージ1 -  $n$  処理部51は、変数  $k$  の値が「 $n$ 」であるか否かを判定し、変数  $k$  の値が「 $n$ 」であり、処理対象のステージが最前段のステージ  $n$  であると判定した場合、処理をステップ S 159に進め、ステージ  $n$  の各鎖において重みを累算する第2重み付き分岐結果累算処理を実行する。ステップ S 159の処理を終了すると、ステージ1 -  $n$  処理部51は、処理をステップ S 160に進め、変数  $j$  の値を「1」インクリメントし、ステップ S 161において、変数  $j$  の値が任意の正の整数  $M$  より大きいか否かを判定する。変数  $j$  の値が  $M$  より大きくなく、変数  $j$  が示す鎖は存在すると判定した場合、ステージ1 -  $n$  処理部51は、処理をステップ S 159に戻し、それ以降の処理を繰り返す。つまり、ステージ1 -  $n$  処理部51は、ステージ  $n$  の鎖のそれぞれについて、ステップ S 159乃至ステップ S 161の処理を実行する。そして、ステップ S 161において変数  $j$  の値が  $M$  より大きいと判定した場合、ステージ1 -  $n$  処理部51は処理を終了する。

30

【 0 1 2 3 】

また、図16のステップ S 158において、変数  $k$  の値が「1」でないと判定した場合、すなわち、処理対象のステージがステージ2乃至ステージ  $(n - 1)$  のいずれかである場合、ステージ1 -  $n$  処理部51は、処理をステップ S 162に進め、第3重み付き分岐結果累算処理を実行する。ステップ S 162の処理が終了すると、ステージ1 -  $n$  処理部51は、処理をステップ S 163に進め、変数  $j$  の値を「1」インクリメントし、ステップ S 164において、変数  $j$  の値が任意の正の整数  $M$  より大きいか否かを判定する。変数  $j$  の値が  $M$  より大きくなく、変数  $j$  が示す鎖は存在すると判定した場合、ステージ1 -  $n$  処理部51は、処理をステップ S 162に戻し、それ以降の処理を繰り返す。つまり、ステージ1 -  $n$  処理部51は、ステージ2乃至ステージ  $(n - 1)$  の鎖のそれぞれについて、ステップ S 162乃至ステップ S 164の処理を実行する。そして、ステップ S 164において変数  $j$  の値が  $M$  より大きいと判定した場合、ステージ1 -  $n$  処理部51は処理を終了する。

40

【 0 1 2 4 】

次に、図17のフローチャートを参照して、図8のステージ1の一番上の鎖における重み付き分岐結果累算処理である第1重み付き分岐結果累算処理の流れの例を説明する。

【 0 1 2 5 】

50

第1重み付き分岐結果累算処理が開始されると、ステップS181において、1本目の鎖について、重み付き分岐結果値累算部62は、レジスタ44-2より重み付き分岐結果累算値81( REG(2, 1))を読み出して取得する。ステップS182において、重み選択部61は、実行パス履歴85の最新のアドレスA[0]をインデックスとして、重みテーブル82-1より重みを取得する(すなわちWT(1, A[0])より重みを取得する)。

【0126】

重み付き分岐結果値累算部62は、ステップS183において、レジスタ44より、グローバル履歴84の最新の分岐結果84-1(GHR[0])を取得し、ステップS184において、その分岐結果を反転させ、ステップS185において、加算器102-1を用いて、重み付き分岐結果累算値81( REG(2, 1))を被加数とし、重みテーブルWT(1, A[0])より取得した重みを加数とし、反転された分岐結果を補数信号として加算処理を行う。すなわち、重みの値そのままと反転したもの(1の補数)を用意しておき、分岐結果84-1=0の場合は、これを反転した「1」の補数信号が送られてくるので、そのときは、重みを反転した値(1の補数)をセレクトして、重みを反転した値(1の補数)と分岐結果84-1=0を反転した「1」とを加算して2の補数(負の表現)として、重み付き分岐結果累算値81( REG(2, 1))と加算し、分岐結果84-1=1の場合は、これを反転した「0」が補数信号として送られてくるので、そのときは、重みそのままの値をセレクトして、その値と分岐結果84-1=1を反転した「0」とを加算して(結局重みそのままの値)、重み付き分岐結果累算値81( REG(2, 1))と加算する。

10

【0127】

重み付き分岐結果値累算部62は、ステップS186において、加算結果を新たな重み付き分岐結果累算値81( REG(1, 1))としてレジスタ44-1に保持させる。

20

【0128】

次に、図18のフローチャートを参照して、ステージnにおける重み付き分岐結果の累算処理である第2重み付き分岐結果累算処理の流れの例を説明する。

【0129】

第2重み付き分岐結果累算処理が開始されると、ステップS201において、重み選択部61は、多重化アドレス83と実行パス履歴85の最新のアドレスの排他的論理和をインデックスとして重みテーブルWT((n-1)×M+j, MA[(n-1)×M-n+j XOR A[0]])より重みを取得する。ステップS202において、重み付き分岐結果値累算部62は、グローバル履歴84より分岐結果84-{(n-1)×M-n+j}(GHR[(n-1)×M-n+j])を取得する。ステップS203において、重み付き分岐結果値累算部62は、グローバル履歴84より取得した分岐結果84-{(n-1)×M-n+j}の値が「1」であるか否かを判定し、「1」であると判定した場合、そのまま処理をステップS205に進め、「1」でないと判定した場合、処理をステップS204に進め、取得した重みの符号を反転させた後、ステップS205に処理を進める。なお、ステップS204における重みの符号を反転は、重みの値(ビットパターン)を反転させた値(重みの値の1の補数)と分岐結果84-{(n-1)×M-n+j}(GHR[(n-1)×M-n+j])を反転させた値(=1)を加算して2の補数とすることにより行う。

30

【0130】

次に、図19のフローチャートを参照して、図8のステージ1の上から2本目乃至M本目の各鎖、並びに、ステージ2乃至ステージ(n-1)の全ての鎖における重み付き分岐結果累算処理である第3重み付き分岐結果累算処理の流れの例を説明する。

40

【0131】

第3重み付き分岐結果累算処理が開始されると、ステップS221において、重み付き分岐結果値累算部62は、レジスタ44-(k+1)に保持されている重み付き分岐結果累算値81( REG(k+1, j))を読み出して取得する。ステップS222において、重み選択部61は、実行パス履歴85の最新のアドレス(A[0])と多重化アドレス83の排他的論理和をインデックスとして、重みテーブル82-((k-1)×M+j)(すなわちWT((k-1)×M+j, MA[(k-1)×M-k+j] XOR A[0]))より重みを取得する。

【0132】

重み付き分岐結果値累算部62は、ステップS223において、レジスタ44よりグローバル

50

履歴84の分岐結果84 -  $\{ (k-1) \times M - k + j \}$  (GHR [  $(k-1) \times M - k + j$  ])を取得し、ステップS 224において、重み付き分岐結果累算値81(REG(k+1, 1))を被加数とし、重みテーブルWT( $(k-1) \times M + j$ , MA[ $(k-1) \times M - k + j$ ] XOR A[0])より取得した重みを加数とし、グローバル履歴84より取得した分岐結果84 -  $\{ (k-1) \times M - k + j \}$  (GHR[ $(k-1) \times M - k + j$ ])を反転させて補数信号として加算処理を行う。

【 0 1 3 3 】

重み付き分岐結果累算部62は、ステップS 225において、加算結果を新たな重み付き分岐結果累算値81(REG(k, j))としてレジスタ44 - kに保持させる。

【 0 1 3 4 】

次に、図20のフローチャートを参照して、図6の学習処理部42により実行される学習処理の流れの例を説明する。

10

【 0 1 3 5 】

学習処理は、演算部24において行われた実際の命令実行結果を、重みテーブル82の内容に反映させるように、それらの学習を行う処理である。この学習処理は、学習処理部42によって、上述した分岐予測処理とは独立して、パイプライン処理が行われている間実行される。

【 0 1 3 6 】

例えば、学習処理が開始されると、学習処理部42は、ステップS 241において、演算部24より取得した情報に基づいて、分岐方向が確定したか否か、すなわち、演算部24において分岐命令が実行されその実行結果が得られたか否かを判定し、得られたと判定するまで待機する。分岐命令の実行結果は、演算部24より分岐予測部21に供給される。分岐命令の実行が終了し、分岐方向が確定したと判定した場合、学習処理部42は、ステップS 243において、A<sup>3</sup>PBPのステージ0において算出された予測値と演算部24より取得した分岐命令の実行結果である分岐結果とが等しいか否か(予測が当たったか否か)を判定する。予測値と分岐結果とが等しい(予測が当たった)と判定した場合、学習処理部42は、処理をステップS 244に進め、学習処理部42は、予測値を算出する際の重み付き分岐結果累算値の総和と閾値との加算結果の絶対値が、予め定められた所定の値である偏向値より小さいか否かを判定する。すなわち、その予測が十分に確信を持ってなされたものであるか否か(予測値の信頼性が高いか否か)を判定する。

20

【 0 1 3 7 】

ステップS 244において、予測時の計算値の絶対値が偏向値より小さい、つまり、予測値の信頼性が低いと判定した場合、学習処理部42は、処理をステップS 245に進める。また、ステップS 243において、予測値と分岐結果とが等しくない(予測が外れた)と判定した場合、学習処理部42は、処理をステップS 245に進める。

30

【 0 1 3 8 】

ステップS 245において、学習処理部42は、今回の分岐方向と値が一致するグローバル履歴84の分岐結果(GHR [ p ])に対応する重みの値に「1」を加算するように、レジスタ44に保持されている重みテーブル82を更新する。

【 0 1 3 9 】

ステップS 245の処理を終了すると、学習処理部42は、ステップS 246に処理を進め、各重みテーブルの、今回の分岐方向と値が一致しないグローバル履歴84の分岐結果(GHR [ p ])に対応する重みの値に「1」を減算するように、レジスタ44に保持されている重みテーブル82を更新する。

40

【 0 1 4 0 】

換言すると、学習処理部42は、ステップS 245およびステップS 246の処理により、図8に示されるA<sup>3</sup>PBPの各重みテーブル82の、今回の分岐予測処理において読み出される値に「1」を加算または減算することにより、分岐方向と値が一致するグローバル履歴の重みを増し、分岐方向と一致しないグローバル履歴の重みを低減させるように、グローバル履歴84の各分岐結果(GHR [ p ])の重みを再調整する(更新する)。

【 0 1 4 1 】

50

ステップS246の処理を終了すると、学習処理部42は、学習処理を終了する。また、ステップS244において、予測時の計算値の絶対値が偏向値以上である、つまり、予測値の信頼性が高いと判定した場合、学習処理部42は、学習処理を終了する。

#### 【0142】

以上より、本実施形態によれば、分岐予測部21は、重みの読み出しに、実行パス履歴85の最新のアドレスA[0]に加えてローカル履歴を反映させた多重化アドレス(MA)を利用することにより、重みの負の衝突を回避し、予測精度を向上させることができる。さらに、閾値の選択にもローカル履歴を反映させた多重化アドレス(MA)を利用するので、分岐予測部21は、閾値においても同様に重みの負の衝突を回避し、予測精度をさらに向上させることができる。また、分岐予測部21は、閾値の算出において、閾値の候補の読み出しと並行して、ローカル履歴86の分岐結果の読み出しを行うので、閾値の算出の処理時間(レイテンシ)を低減させることができる。また、最新の分岐結果に近いグローバル履歴84GHR[p]及びターゲットアドレスに近い実行パス履歴85のアドレスほど後のステージで使用されるパイプライン構造としたため、それぞれのステージにおいて処理を行う際に、演算結果に基づいて更新されたグローバル履歴84GHR[p]及びよりターゲットアドレスに近いアドレスに更新された実行パス履歴85A[0]を用いることができるため、予測精度が向上する。さらに、分岐予測部21は、パイプライン処理により重みの読み出しにかかるレイテンシを隠蔽することができる。また、分岐予測部21は、ターゲットアドレスの代わりにターゲットアドレスと相関の強い実行パス履歴85の最新のアドレスA[0]を用いて重みテーブル82より重みを読み出すことにより、事前計算を可能にし、パイプライン処理を実現することができる。さらに、分岐予測部21は、連鎖的に同じインデックスを使用して重み読み出しを行わないようなパイプライン構造(図8)を実現するので、PTBPのように予測に関係の無い実行パス履歴が及ぼす悪影響を最小限に抑えることができる。

#### 【0143】

以上のように、分岐予測部21は、予測に必要な時間を抑制するとともに分岐予測の精度を向上させることができる。

#### 【0144】

以上のような分岐予測部21(A<sup>3</sup>PBP)による分岐予測処理の予測精度をシミュレーション結果について説明する。

#### 【0145】

A<sup>3</sup>PBPとの評価対象としてgshare分岐予測器、グローバル/ローカルパーセプトロン分岐予測器(Global/Local Perceptron)、パスベースドパーセプトロン分岐予測器(Path-Based Perceptron)、アヘッドパイプラインドピースワイズリニア分岐予測器(Ahead Pipelined Piecewise Linear)、並びに、PTBPの8本鎖(PTBP Eight Chain)を用いた。

#### 【0146】

シミュレーションは、SPECINT2000の12種類のベンチマークをSimCoreシミュレータで1億命令程度実行した際にトレースしたデータを、各分岐予測器をシミュレートしたプログラムで処理し、その予測ミス率を測定することにより行った。評価する分岐予測器の記憶容量は、8KBから256KBまでの6段階で制限し、各容量での予測ミス率の違いを評価した。図21の表は、各分岐予測器が各記憶容量で使用するグローバル履歴長の値(バジェットサイズ)の一覧を示している。このシミュレーションにおいて、重みは原則8ビットを使用した。A<sup>3</sup>PBPの8KB、16KB、32KBに関しては7ビットを使用した。また、パーセプトロン分岐予測器の学習閾値は全て $2.1 \times (GHR + 1)$ とした。

#### 【0147】

以上のような条件において行ったシミュレーション結果をグラフ化したものを図22に示す。図22のグラフは、各予測器の記憶容量別の予測ミス率の関係を示すグラフである。図22に示されるように、黒丸の点と実線の曲線で示されるA<sup>3</sup>PBP(AAAPBP)の予測ミス率は全ての記憶容量において、他の、どの分岐予測器の予測ミス率よりも低い。例えば、A<sup>3</sup>PBP(AAAPBP)の予測ミス率は、8本鎖のPTBP(PTBP Eight Chain)のそれと比較すると、平均で5.7%、最高で6.4%(16KB構成時)も低減されている。以上のようにA<sup>3</sup>PBPは他の分岐予測



器よりもテーブルの使用効率が高い。つまり、分岐予測部21は、分岐予測の精度を向上させることができる。

【0148】

なお、以上においては、分岐予測部21は、図8に示されるようなA<sup>3</sup>PBPによる分岐予測パイプライン処理を実現するように説明したが、例えば、図23に示されるように、ローカル履歴86(ローカル履歴86の分岐結果を反映させた多重化アドレス83)を用いないようにしてもよい。

【0149】

図23の例の場合、ステージ0に設けられた、閾値を選択するための重みテーブル182-0と、グローバル履歴84の最新の分岐結果84-1(GHR[0])の重みを選択するための重みテーブル182-1は、ローカル履歴86(多重化アドレス83)を用いずに、実行パス履歴85の最新のアドレスA[0]のみをインデックスとして閾値を選択するテーブルである。また、それら以外の、ステージnからステージ1(図23の例の場合、n=2)の各重みテーブル182-2乃至182-nMは、それぞれ、ローカル履歴86の分岐結果(多重化アドレス83)の代わりに、重み付けの相手となるグローバル履歴84の分岐結果84-(i×j-i)と履歴順が等しい、実行パス履歴85のアドレスA[i×j-i]を用い、そのアドレスA[i×j-i]と最新のアドレスA[0]との排他的論理和をインデックスとしてグローバル履歴84の分岐結果84-(i×j-i)の重みを選択するテーブルである。

10

【0150】

つまり、図23の例の場合、重み選択部61は、ステージnからステージ1の各グローバル履歴84の分岐結果(GHR[p])について、実行パス履歴85の最新のアドレスA[0]と、グローバル履歴84の分岐結果84-p(GHR[p])と履歴順pが等しい実行パス履歴85のアドレス(A[p])の排他的論理和、または実行パス履歴85の最新のアドレスA[0]のみ、のいずれかをインデックス(少なくとも実行パス履歴85の最新のアドレスA[0]を反映したインデックス)として、分岐結果84-pに対応する重みテーブル182より重みを選択して読み出す。

20

【0151】

xを任意の自然数とすると、上述のステージnからステージ1においては、分岐結果GHR[x]に対応するアドレスとして、このアドレスA[x]を用いている。もちろん、A[x]以外の任意のアドレスを分岐結果GHR[x]に対応させるようにしてもよい。ただし、この対応関係は、予測処理の開始前に予め決められている。

30

【0152】

この場合、図8を参照して説明したA<sup>3</sup>PBPによる予測処理よりも、予測精度は低下するが、最新の分岐結果に近いグローバル履歴84GHR[p]及びターゲットアドレスに近い実行パス履歴85のアドレスほど後のステージで使用されるパイプライン構造であるため、それぞれのステージにおいて処理を行う際に、演算結果に基づいて更新されたグローバル履歴84GHR[p]及びよりターゲットアドレスに近いアドレスに更新された実行パス履歴85A[0]を用いることによって予測精度の向上を実現することができる。

【0153】

また、図24に示されるように、ローカル履歴86(ローカル履歴86の分岐結果を反映させた多重化アドレス83)を用いて閾値を選択する方法を、従来のPTBPに適用することも可能である。

40

【0154】

図24の例の場合、ステージ0の構成は、図8に示されるA<sup>3</sup>PBPのステージ0の構成と同様である。ステージnからステージ1の構成は、図3を参照して説明したパイプラインPTBPの構成と同様であり、重みテーブル201-1乃至重みテーブル201-2Nは、それぞれ、実行パス履歴85の最新のアドレスA[0]と実行パス履歴85のアドレスA[j]の排他的論理和をインデックスとして重みが読み出される。各テーブルより読み出された重みは、それぞれ、乗算器202-1乃至乗算器202-2Nにおいて、グローバル履歴84の最新の分岐結果GHR[0]と乗算され、累算される。この重み付き分岐結果値は、ステージ0において総和が求

50

められ、その総和が閾値と加算され予測値が求められる。

【 0 1 5 5 】

このようにすることによっても、分岐予測部21は、予測精度を向上させることができる。ただし、この場合、図8を参照して説明したA<sup>3</sup>PBPによる予測処理よりも、予測精度は低下する。

【 0 1 5 6 】

以上のような各構成において行ったシミュレーション結果をグラフ化したものを図25に示す。図25のグラフは、各予測器の記憶容量別の予測ミス率の関係を示すグラフである。シミュレーション方法および各条件は、図22の場合と同様である。ここでは、図25に示されるように、図8に示されるA<sup>3</sup>PBP、図23に示される、ローカル履歴86を用いない新パイプライン分岐予測器(New Pipeline)、PTBPの8本鎖(PTBP8x)、並びに、図24に示される、PTBPの閾値選択にローカル履歴86(多重化アドレス83)を用いるようにした分岐予測器(PTBP+MA)を比較した。

10

【 0 1 5 7 】

図25に示されるように、予測ミス率は、黒のひし形の点と実線の曲線で示されるA<sup>3</sup>PBP、PTBP+MA、New Pipeline、PTBP8xの順で低い。すなわち、図8に示されるA<sup>3</sup>PBPの構成の内、ローカル履歴86(多重化アドレス83)を用いて重み読み出しや閾値読み出しを行う構成と、ステージnからステージ1のパイプライン構成のいずれも、従来のPTBP8xの場合よりも予測ミス率を低減させることができる。従って、分岐予測部21は、それらのうち、少なくともいずれか一方を適用することにより、分岐予測の精度を向上させることができ、さらに、それらを図8に示されるように併用するようにすることにより、予測ミス率をさらに低減させ、分岐予測の精度をさらに向上させることができる。

20

【 0 1 5 8 】

なお、以上において、閾値の選択において、閾値の候補の読み出しとローカル履歴86からの分岐結果の読み出しを並行して行うように説明したが、これに限らず、例えば、図26に示されるように、最初にローカル履歴86より分岐結果を読み出して多重化アドレス83を生成し、その多重化アドレス83をインデックスとして閾値選択用の重みテーブル82-0より重み(閾値)を読み出すようにしてもよい。

【 0 1 5 9 】

図26は、図8のステージ0の構成例を示すブロック図である。

30

【 0 1 6 0 】

図26の例の場合、実行パス履歴85の最新のアドレスA[0]は、ローカル履歴86(LHT)のインデックスとして使用され、分岐結果の選択に用いられる。ローカル履歴86(LHT)より読み出された分岐結果(LH)は多重化アドレス生成部221に供給される。多重化アドレス生成部221には、また、実行パス履歴85の最新のアドレスA[0]も供給される。多重化アドレス生成部221は、実行パス履歴85の最新のアドレスA[0]の下位ビットを削除して分岐結果を挿入することにより、多重化アドレス83(MA)を生成する。生成された多重化アドレス83は、重みテーブル82-0のインデックスとして使用され、閾値の読み出しに使用される。この場合、多重化アドレス生成部221はステージ0処理部52を構成する。

【 0 1 6 1 】

このように閾値を選択する前に多重化アドレス83を生成しても、分岐予測部21は、図9の場合と同様の結果(閾値)を得ることができる。ただし、この場合、ローカル履歴86からの分岐結果の読み出しと重みの読み出しを並行して行わないため、図8の構成の場合と比較して閾値の算出のレイテンシが増大する。その代わりに、ステージ0において多重化アドレス生成部221で生成された閾値選択用の多重化アドレス83は、現在ステージ0で分岐予測を行っている分岐命令の次以降の分岐命令についてのステージnからステージ1における重み読み出しのための多重化アドレス83に流用することができるので、別途多重化アドレス生成部74を設けて、現在のステージ0処理と並行して多重化アドレスを生成する必要がない。

40

【 0 1 6 2 】

50

図27のフローチャートを参照して、A<sup>3</sup>PBPのステージ0における閾値選択のための構成が図26に示される構成である場合の、ステージ0処理の流れの例を説明する。このフローチャートは、図15のフローチャートに対応する。

【0163】

ステージ0処理が開始されると、予測値算出部73は、ステップS261において、レジスタ44-1に保持されている重み付き分岐結果累算値81(REG(1, j))を全て読み出し、加算器11を用いて、それらの総和を算出する。この処理と並行して、閾値算出部72は、ステップS262において、実行パス履歴85の最新のアドレスA[0]をインデックスとしてローカル履歴86(LHT[A[0]])より分岐結果(LHT(A[0]))を読み出す。読み出された分岐結果は、実行パス履歴85の最新のアドレスA[0]とともに、多重化アドレス生成部221に供給される。多重化アドレス生成部221は、ステップS263において、供給された実行パス履歴85の最新のアドレスA[0]の下位2ビットを削除し、ステップS264において、削除されたその2ビットに分岐結果を挿入し、多重化アドレス83(MA[0])を生成する。

10

【0164】

ステップS265において、閾値算出部72は、多重化アドレス生成部221において生成した多重化アドレス83(MA[0])をインデックスとして閾値用の重みテーブル82-0(WT[0, MA[0]])より閾値を取得する。

【0165】

ステップS266において、予測値算出部73は、加算器113を用いて重み付き分岐結果累算値の総和と閾値を加算し、ステップS267において、加算結果の最上位ビットの値を反転させ、ステップS268において、その反転結果(最上位ビット)を予測値として出力する。

20

【0166】

多重化アドレス生成部74は、ステップS269において、ステップS264において生成した多重化アドレス83(MA[0])を更新処理部43に出力し、更新処理部43によりレジスタの多重化アドレス83が更新される。

【0167】

以上のように、ステージ0処理部52は、多重化アドレス83を閾値選択の前に算出するようにすることもできる。

【0168】

また、以上においては、図8のステージ1の上から1本目の鎖において、グローバル履歴84の最新の分岐結果GHR[0]を用いるように説明したが、これに限らず、例えば、ステージ0において1回前の処理において得られた予測値を用いるようにしてもよい。つまり、グローバル履歴84の分岐結果を用いる代わりに予測値を用いることにより、予測処理部41は、演算部24(図5)において分岐結果が確定する前に重み付き分岐結果の累算を行うことができる。

30

【0169】

図28は、この場合のA<sup>3</sup>PBPの構成の例を示す模式図である。

【0170】

この図28の例の場合、ステージ1の重みテーブル82-1からは、実行パス履歴85の最新のアドレスA[0]をインデックスとして重みを読み出され、予測値が分岐成立(「1」)、不成立(「-1」)の両方向を想定して、それぞれの値に対して読み出した重みで重み付けされる。そして、それぞれに重み付けされた値が加算器241と加算器242に供給され、加算器241および加算器242は、それぞれに重み付けされた値と、レジスタ44-2より読み出した重み付き分岐結果累算値81(REG(2, 1))とを加算し、加算結果がセクタ243に供給される。

40

【0171】

セクタ243は、予測値(前の分岐命令に対する分岐予測におけるステージ0処理により得られた予測値)に基づいて、加算器241および加算器242において算出された2つの重み付き分岐結果累算値81のいずれか一方を選択してレジスタ44-1に保持させる(REG1, 1)。

【0172】

50

以上のようにすることにより、演算部24(図5)において分岐結果が確定する前に重みの累算を行うことができる。

【0173】

なお、以上においては、セレクタ243をステージ1に設けるように説明したが、このセレクタ243をステージ0に設け、加算器241および加算器242の両方の加算結果をレジスタ44-1に保持させるようにしてもよい。その場合、セレクタ243は、ステージ0処理において、前回算出された予測値を用いて、このレジスタ44-1に保持されている2つの重み付き分岐結果累算値81、すなわち、加算器241において算出された加算結果と、加算器242において算出された加算結果のうち、いずれか一方を選択し、その選択した値を、上から1本目の鎖の重み付き分岐結果累算値81として加算器111に供給する。

10

【0174】

上述した一連の処理は、ハードウェアにより実行させることもできるし、ソフトウェアにより実行させることもできる。この場合、例えば、図29に示されるようなパーソナルコンピュータとして構成されるようにしてもよい。

【0175】

図29において、パーソナルコンピュータ300のCPU301は、ROM(Read Only Memory)302に記憶されているプログラム、または記憶部313からRAM(Random Access Memory)303にロードされたプログラムに従って各種の処理を実行する。RAM303にはまた、CPU301が各種の処理を実行する上において必要なデータなども適宜記憶される。

【0176】

CPU301、ROM302、およびRAM303は、バス304を介して相互に接続されている。このバス304にはまた、入出力インタフェース310も接続されている。

20

【0177】

入出力インタフェース310には、キーボード、マウスなどよりなる入力部311、CRTやLCDなどよりなるディスプレイ、並びにスピーカなどよりなる出力部312、ハードディスクなどより構成される記憶部313、モデムなどより構成される通信部314が接続されている。通信部314は、インターネットを含むネットワークを介しての通信処理を行う。

【0178】

入出力インタフェース310にはまた、必要に応じてドライブ315が接続され、磁気ディスク、光ディスク、光磁気ディスク、或いは半導体メモリなどのリムーバブルメディア321が適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部313にインストールされる。

30

【0179】

上述した一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、ネットワークや記録媒体からインストールされる。

【0180】

この記録媒体は、例えば、図29に示されるように、装置本体とは別に、ユーザにプログラムを配信するために配布される、プログラムが記録されている磁気ディスク(フレキシブルディスクを含む)、光ディスク(CD-ROM、DVDを含む)、光磁気ディスク(MDを含む)、もしくは半導体メモリなどよりなるリムーバブルメディア321により構成されるだけでなく、装置本体に予め組み込まれた状態でユーザに配信される、プログラムが記録されているROM302や、記憶部313に含まれるハードディスクなどで構成される。

40

【0181】

なお、本明細書において、記録媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並行的あるいは個別に実行される処理をも含むものである。

【0182】

また、本明細書において、システムとは、複数のデバイス(装置)により構成される装置全体を表すものである。

【0183】

50

なお、以上において、一つの装置として説明した構成を分割し、複数の装置として構成するようにしてもよい。逆に、以上において複数の装置として説明した構成をまとめて一つの装置として構成されるようにしてもよい。また、各装置の構成に上述した以外の構成を付加するようにしてももちろんよい。さらに、システム全体としての構成や動作が実質的に同じであれば、ある装置の構成の一部を他の装置の構成に含めるようにしてもよい。つまり、本発明の実施の形態は、上述した実施の形態に限定されるものではなく、本発明の要旨を逸脱しない範囲において種々の変更が可能である。

産業上の利用可能性

【0184】

本発明は、分岐予測器に適用することが可能である。

【符号の説明】

【0185】

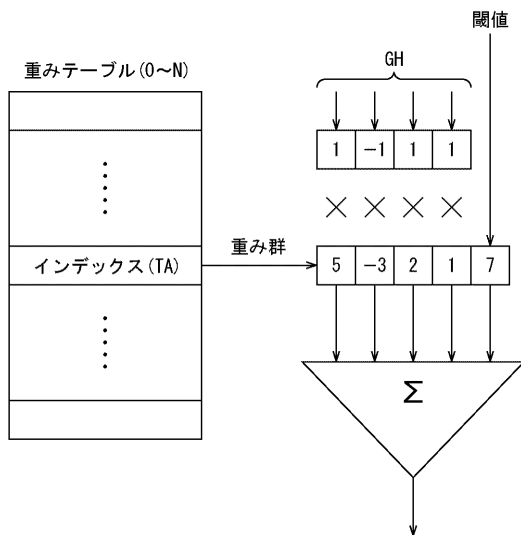
11 CPU, 21 分岐予測部, 41 予測処理部, 42 学習処理部, 43 更新処理部, 51 ステージ1-n処理部, 52 ステージ0処理部, 61 重み選択部, 62 重み付き分岐結果値累算部, 72 閾値算出部, 73 予測値算出部, 74 多重化アドレス生成部, 81 重み付き分岐結果累算値, 82 重みテーブル, 83 多重化アドレス, 84 グローバル履歴, 85 実行パス履歴, 86 ローカル履歴, 87 ターゲットアドレス, 101 反転部, 102 加算器, 111 加算器, 112 セレクタ, 113 加算器, 221 多重化アドレス生成部, 241 加算器, 242 加算器, 243 セレクタ

10

20

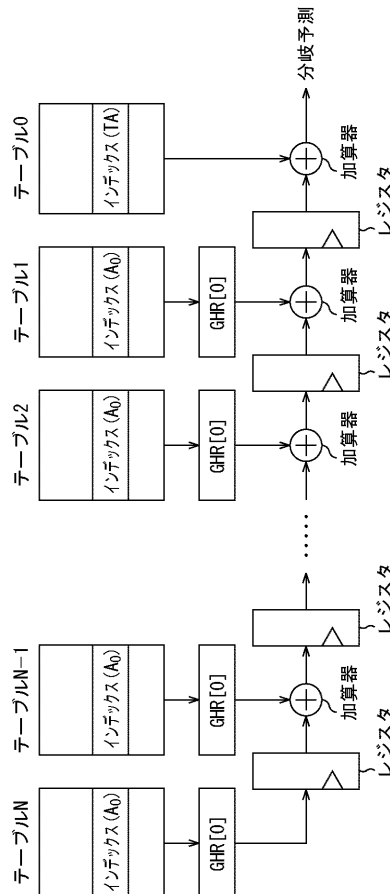
【図1】

図1



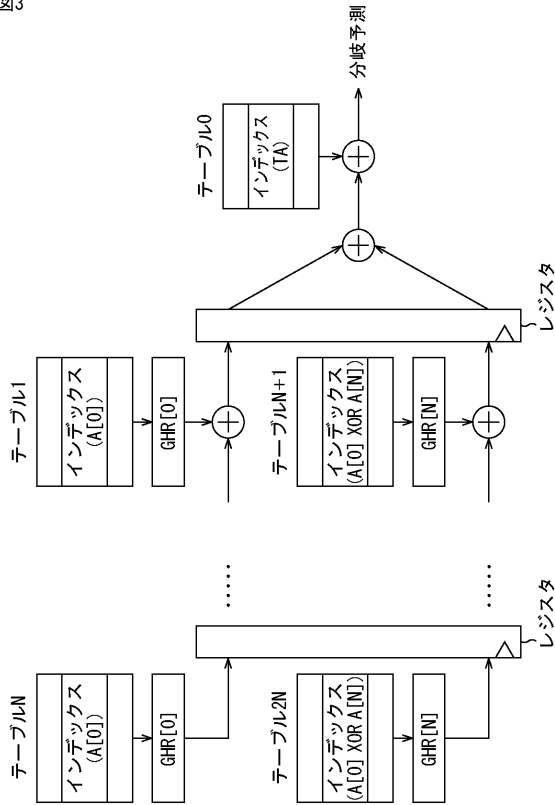
【図2】

図2



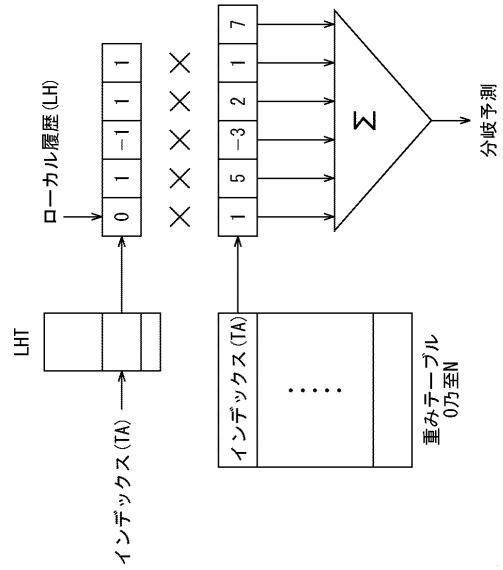
【図3】

図3



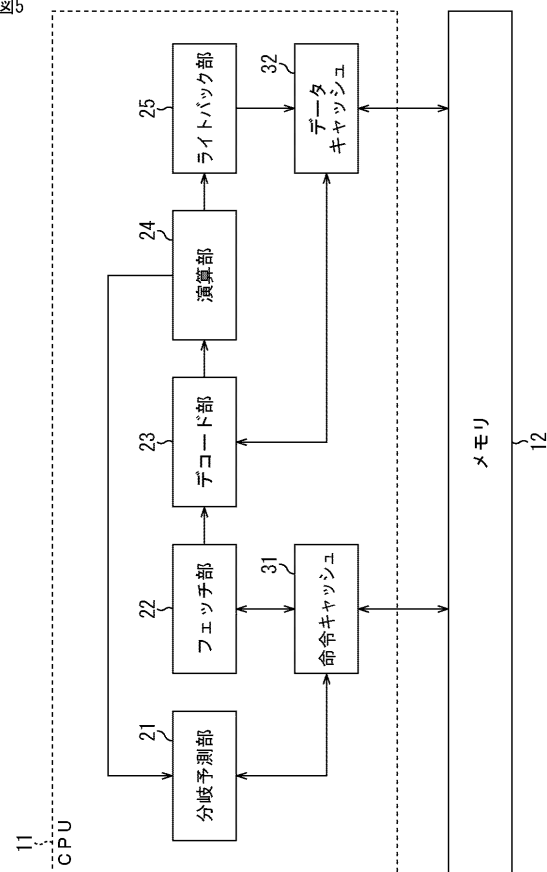
【図4】

図4



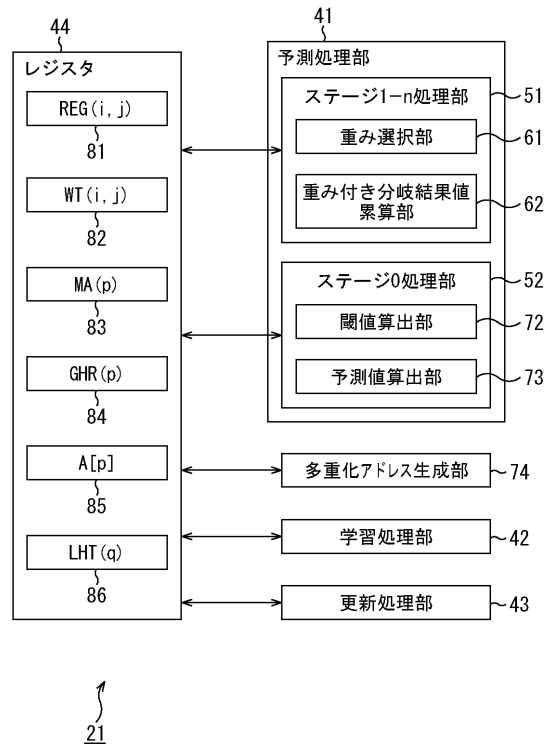
【図5】

図5

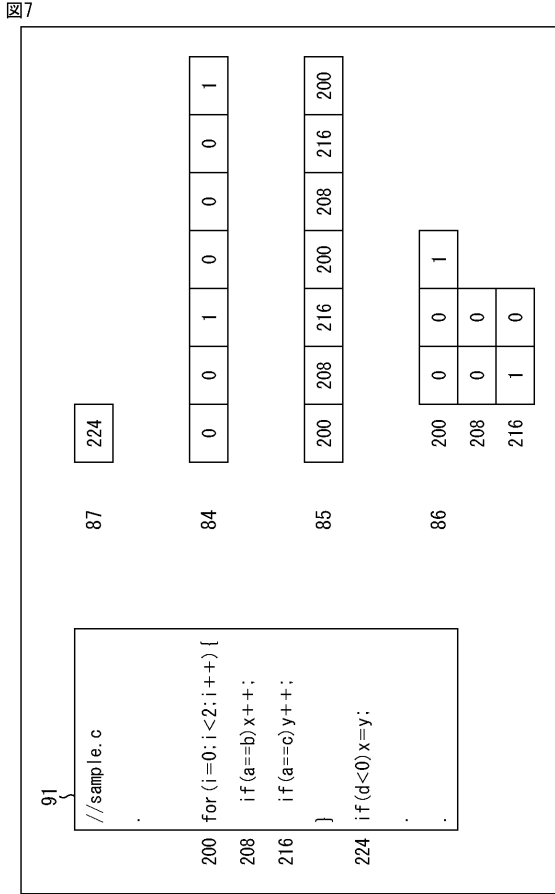


【図6】

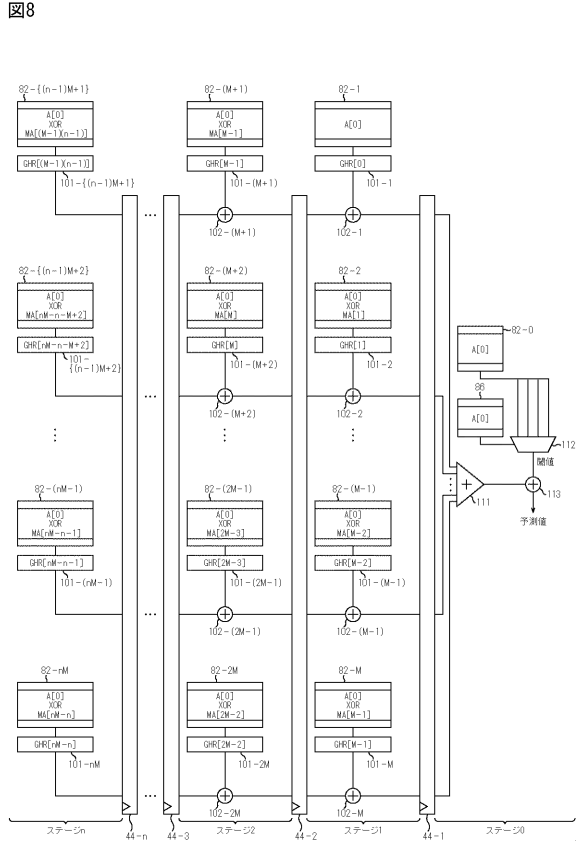
図6



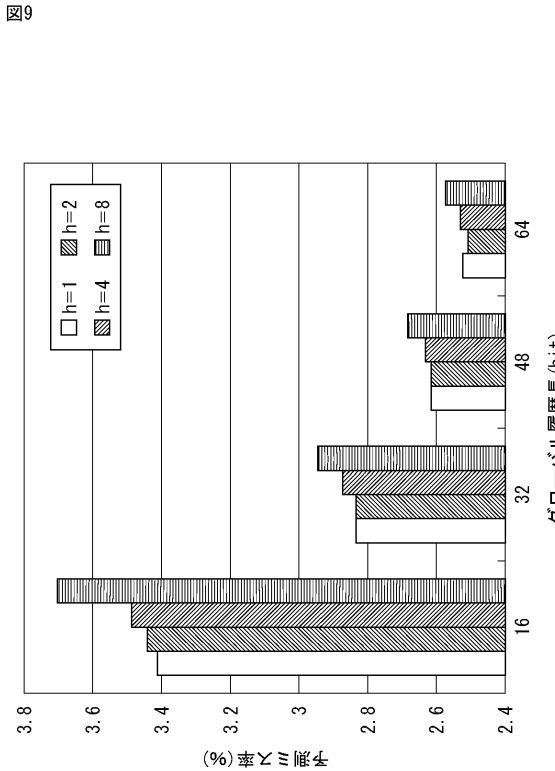
【 図 7 】



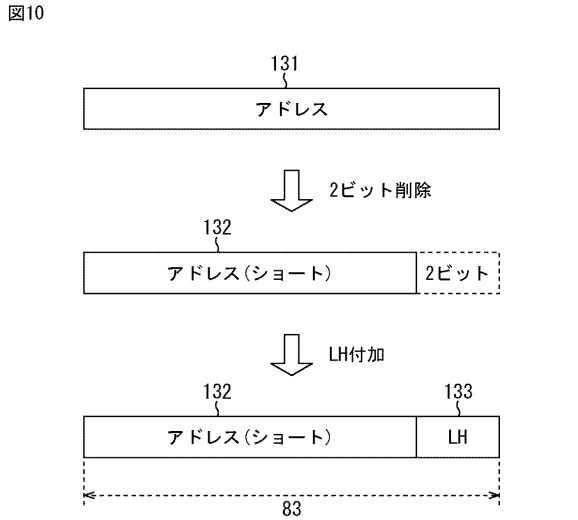
【 図 8 】



【 図 9 】

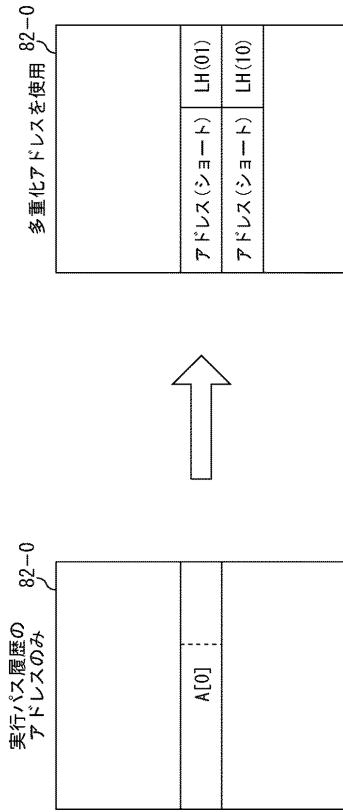


【 図 10 】



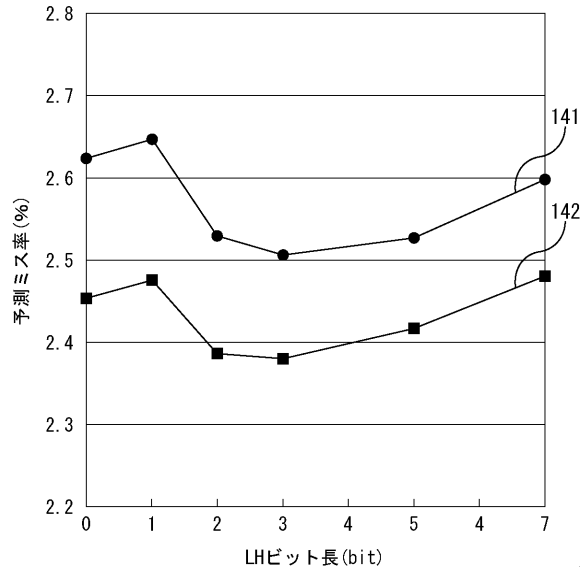
【図11】

図11



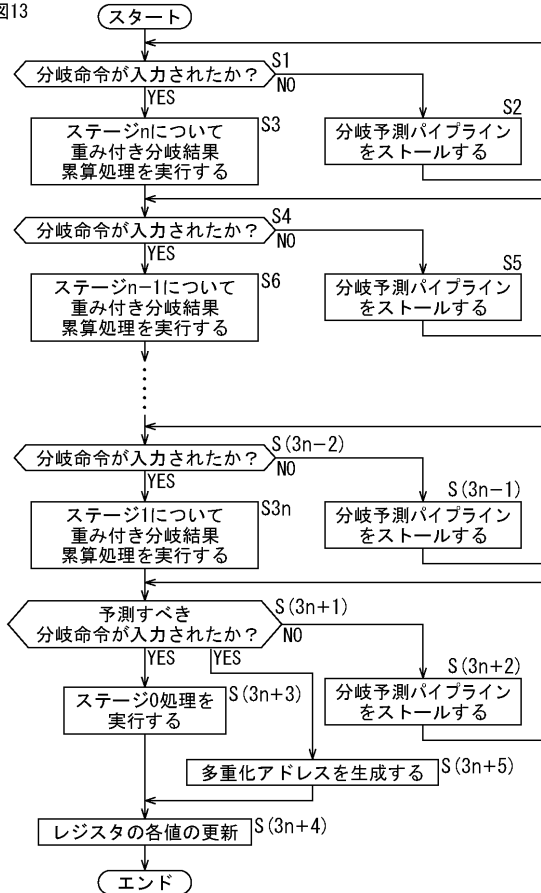
【図12】

図12



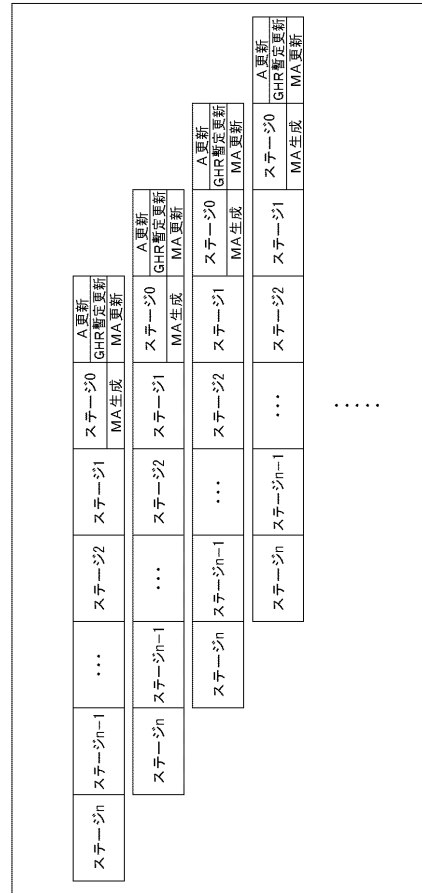
【図13】

図13



【図14】

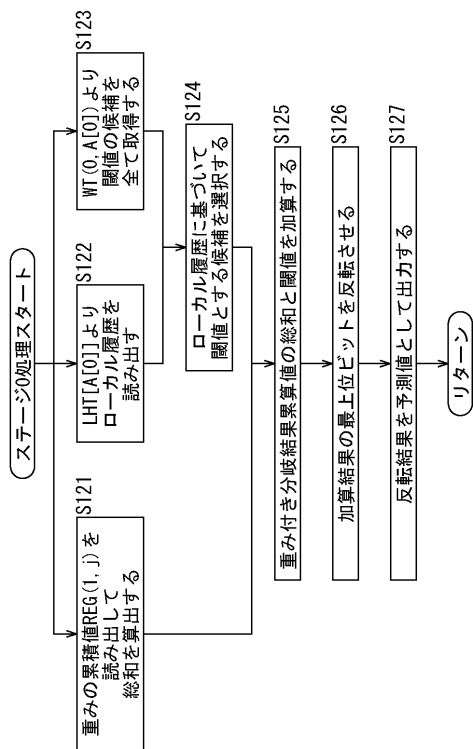
図14





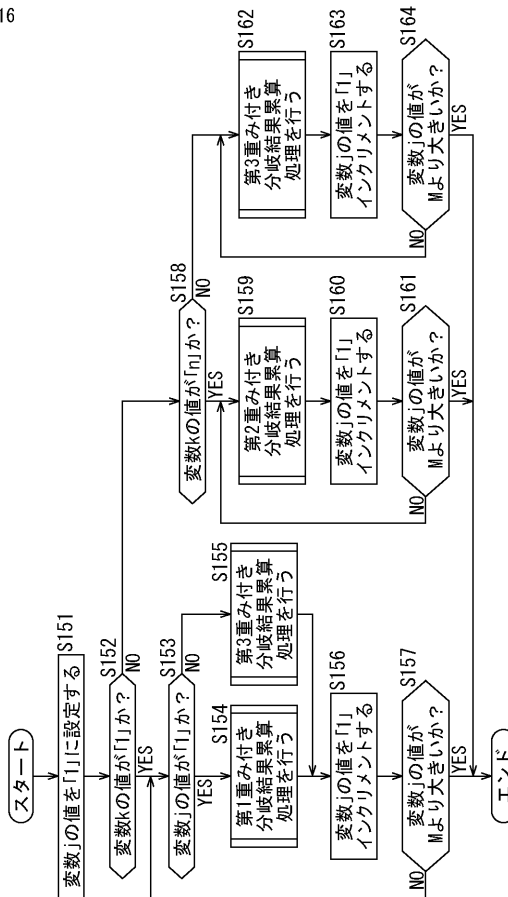
【図 15】

図15



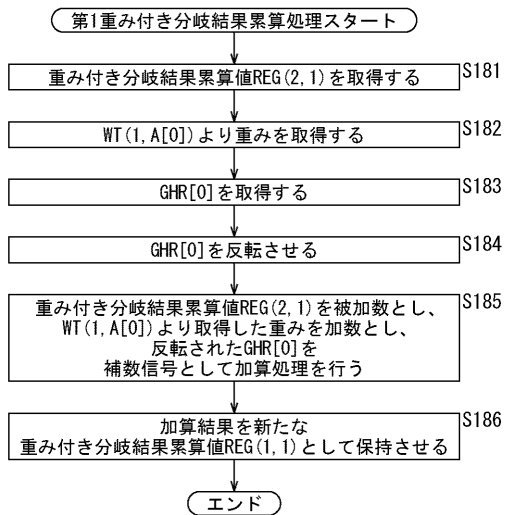
【図 16】

図16



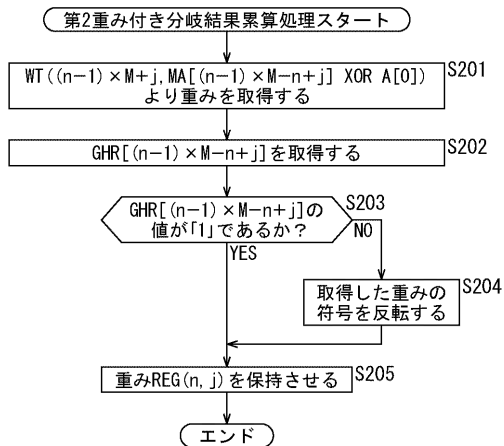
【図 17】

図17



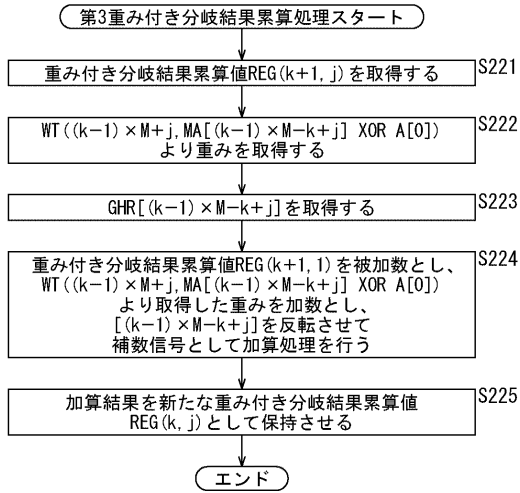
【図 18】

図18



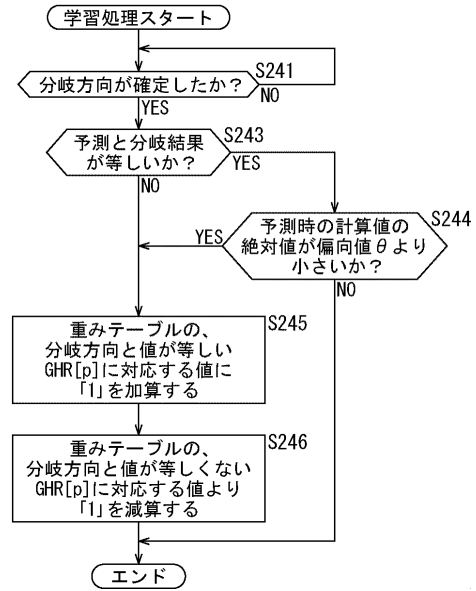
【図 19】

図19



【図 20】

図20



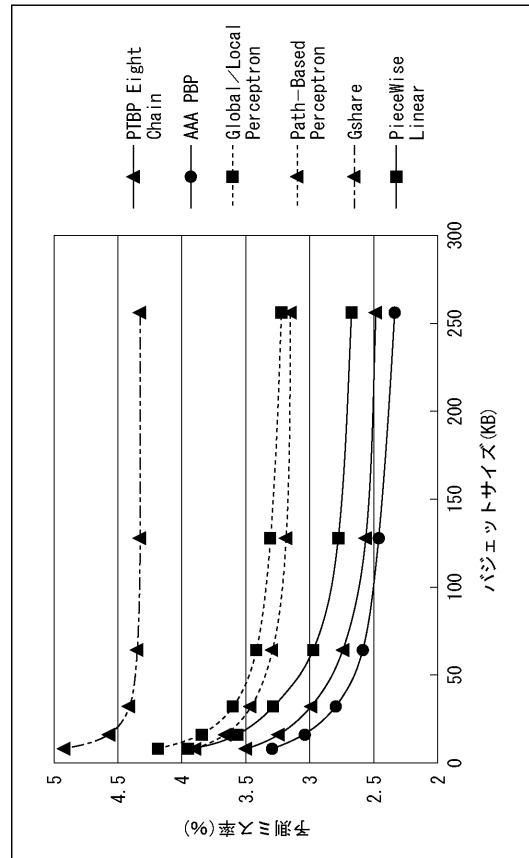
【図 21】

図21

バジェットサイズ	gshare	Globsl Local	Path Based	Piece wise	PTBP	A <sup>3</sup> PBP
8KB	15	34/12	32	20	22	16
16KB	16	38/14	34	24	26	26
32KB	17	40/14	34	26	32	36
64KB	18	50/18	37	43	43	43
128KB	19	60/20	40	50	50	63
256KB	20	70/20	42	51	51	63

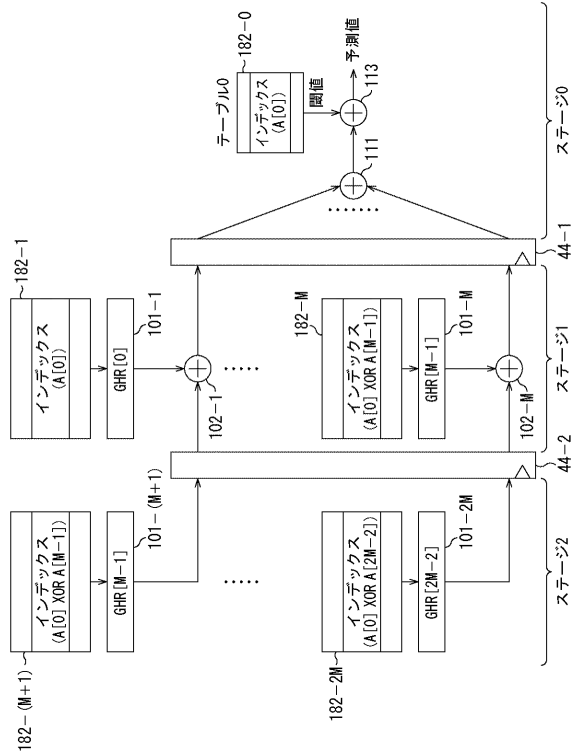
【図 22】

図22



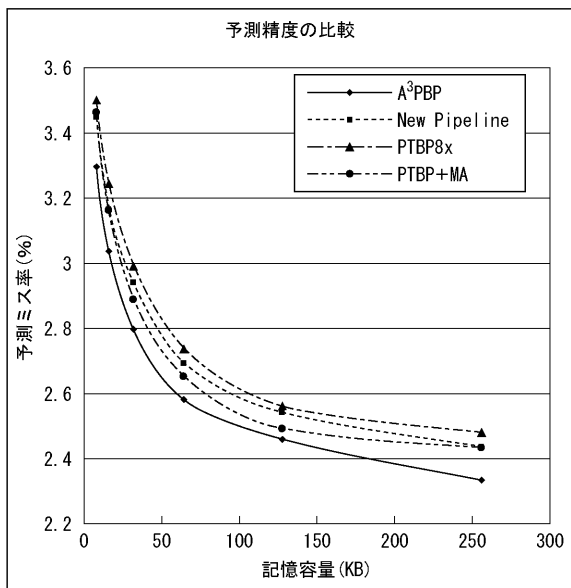
【図 23】

図23



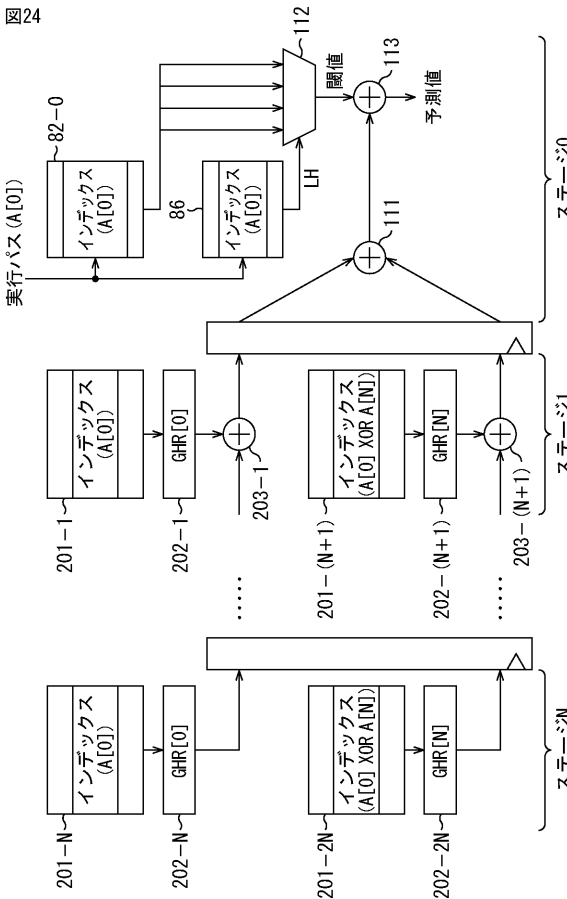
【図 25】

図25



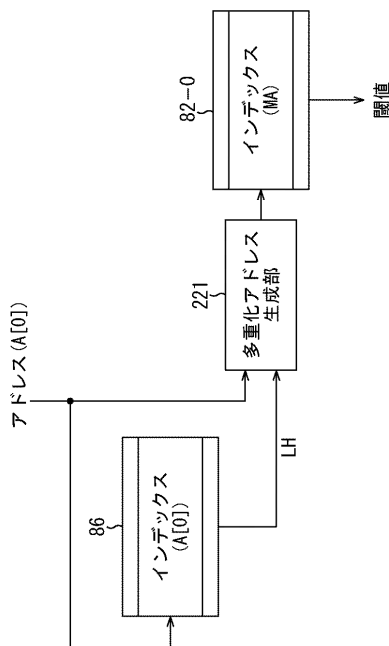
【図 24】

図24



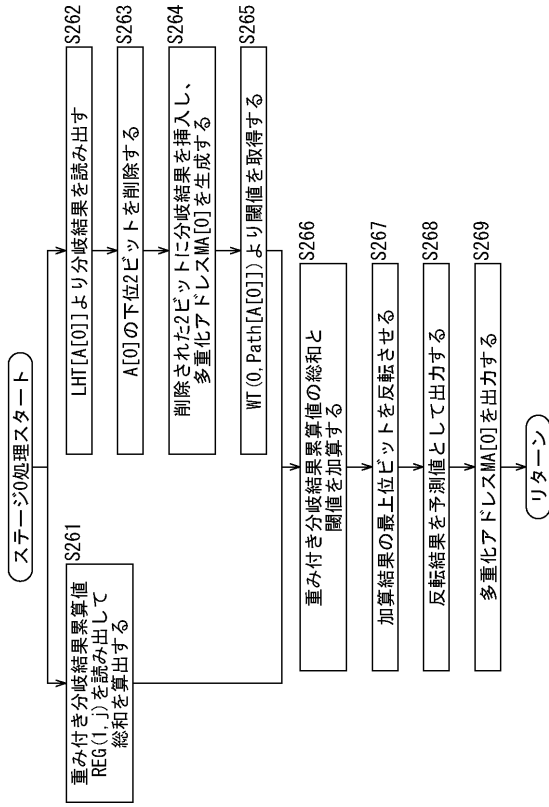
【図 26】

図26



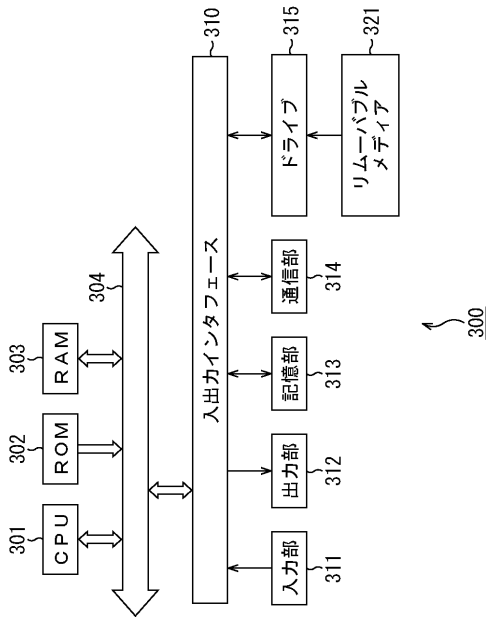
【図27】

図27



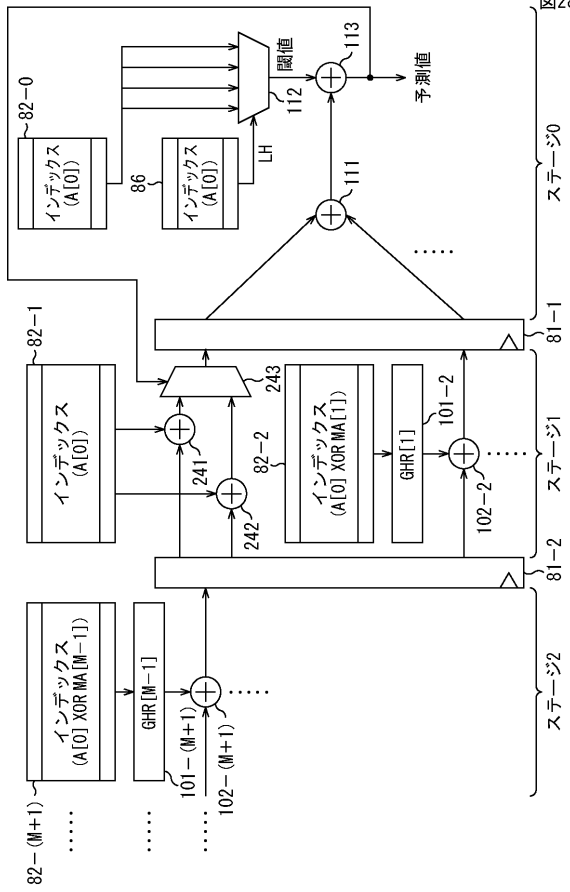
【図29】

図29



【図28】

図28



---

フロントページの続き

- (56)参考文献 ニノ宮康之, 実行パスとローカル履歴を重み選択に利用したパーセプトロン分岐予測器, 電気通信大学 平成17年度情報工学科コンピュータ学講座 卒業研究発表資料, 日本, 国立大学法人電気通信大学, 2006年 2月 6日  
石井康雄ほか1名, 実行パス履歴情報を利用した分岐予測手法, 情報処理学会論文誌 第47巻 No. SIG3 (ACS13), 日本, 社団法人情報処理学会, 2006年 3月15日, p. 58-72  
Tse-Yu Yeh et al., A Comparison of Dynamic Branch Predictors that use Two Levels of Branch History, Proceedings of the 20th Annual International Symposium on Computer Architecture, 1993, IEEE, 1993年 5月16日, pp.257-266  
DANIEL A. JIMENEZ, Improved Latency and Accuracy for Neural Branch Prediction, ACM Transactions on Computer Systems, 米国, ACM, 2005年 5月, Vol. 23, No. 2, pp.197-218

(58)調査した分野(Int.Cl., DB名)

G06F 9/38

G06N 3/08

JSTPlus(JDreamII)

Scopus