

## BTBのエントリ有無を参照した分岐予測器の提案

齋藤 史子<sup>†</sup> 山名 早人<sup>††</sup>

命令間の制御依存によってパイプライン処理を滞らせないために、近年のプロセッサでは、分岐予測が採用されている。分岐予測によって未解決の分岐命令を投機的実行することが可能となる。しかし、近年、命令パイプライン長の深化により、分岐予測ミスペナルティが増大している。そのため、分岐予測ミス率の低減は、プロセッサの性能向上のために不可欠な課題となっている。

分岐予測は、分岐条件の成立・不成立の予測と分岐先アドレスの予測で構成される。分岐先アドレス予測に利用される分岐先アドレスバッファ(BTB: Branch Target Buffer)は、分岐条件が成立する(Taken)分岐をエントリに登録する。BTBのエントリに存在しない分岐の分岐条件成立・不成立状況を解析した結果、BTBにエントリを持たない分岐の殆どが、分岐条件が成立しない(NotTaken)分岐であることが判った。そこで、BTBにエントリを持たない分岐をNotTakenと予測する分岐予測手法を提案する。提案手法では、さらに、分岐方向予測器のエントリ数削減により、競合を緩和するために、BTBにエントリのない分岐は、分岐方向予測器を更新しない。SimpleScalar 3.0c/PISA sim-bpred シミュレータで実験したところ、SPECint95(train)では、8KB Gshare 予測器で平均 1.5%、1.5KB Bi-Mode 予測器で平均 0.4%の予測ミス率が低減した。

### The Branch Predictor referring a BTB Entry Existence

FUMIKO SAITO<sup>†</sup> and HAYATO YAMANA<sup>††</sup>

The branch prediction is installed on the recent processor to avoid stalling pipeline. Branch prediction is a kind of speculative execution for control dependence. In the recent year, the deeper pipeline gets, the higher branch miss prediction penalty reaches. Thus, branch miss prediction rate must lower to rise processor performance.

The branch prediction predicts a branch direction and a branch target address. BTB(Branch Target Buffer) registers **Taken** branch. We found that the most branches, which do not have BTB entry are **NotTaken** branches. We propose the branch predictor referring a BTB entry existence. The proposed predictor only updates the entry of the branch whose target address is registered in BTB, in order to alleviate aliasing. In SPECint95(train), branch prediction miss rate lowers average 1.5% on 8KB Gshare predictor and average 0.4% on 1.5KB Bi-Mode predictor.

#### 1. はじめに

命令間の制御依存によってパイプライン処理を滞らせないために、近年のプロセッサでは、分岐予測が採用されている。分岐予測は、分岐条件の成立・不成立や、分岐先命令を予測して、投機的に分岐命令を実行する技術である。しかし、近年、命令パイプライン長が深化し、分岐予測ミスペナルティの増大が懸念されている。たとえば、Pentium 4 を模した環境では、分岐予測ミスペナルティが 52 サイクルを超えると、処

理性能が低下してしまうことと、分岐予測ミスペナルティの影響がなければ、45~90%プロセッサの処理性能が向上することが示されている<sup>1)</sup>。このように、分岐予測ミス率の低減は、プロセッサの性能向上のためには不可欠の課題となっている。

分岐予測は、分岐条件の成立・不成立の予測と分岐先アドレスの予測によって、分岐命令を投機的に実行する。本稿では、分岐条件の成立・不成立の予測を分岐方向予測、分岐先アドレスの予測を分岐先アドレス予測と呼ぶ。分岐方向予測器は、複数の飽和カウンタ(予測表, PHT: Pattern History Table)で構成される。予測を決定する飽和カウンタは、分岐命令アドレスや分岐履歴を基に選択される。この飽和カウンタは、分岐条件の成立・不成立で遷移し、飽和カウンタの状態によって、分岐条件の成立・不成立を予測する。分

<sup>†</sup> 早稲田大学大学院理工学研究科  
Graduate School of Science and Engineering,  
Waseda University  
<sup>††</sup> 早稲田大学理工学部  
School of Science and Engineering, Waseda University

岐方向予測器は、全条件分岐を対象とする。分岐先アドレス予測器は、分岐先アドレスを格納したレジスタで構成される。この分岐先アドレスを格納したレジスタは、分岐先アドレスバッファ(BTB: Branch Target Buffer)<sup>2)</sup>と呼ばれている。

本稿では、分岐条件が成立する分岐 (Taken 分岐) のみが BTB に分岐先アドレスを格納することに着目する。つまり、分岐条件が常に不成立である分岐 (Always-NotTaken 分岐) は、BTB にエントリを持たない。BTB にエントリが存在する場合と、存在しない場合に分けて、分岐条件の成立・不成立を調査した結果、BTB にエントリが存在しない場合、分岐条件が不成立である分岐 (NotTaken 分岐) である可能性が高いことが判った。また、分岐方向予測の性能低下要因は破壊的競合にあると言われている<sup>3)</sup>。競合は、異なる分岐命令が同一の飽和カウンタを基に予測を行う現象である。特に、破壊的競合は、飽和カウンタを共有した分岐が互いに異なる分岐成立傾向を示し、本来ならば無関係の分岐に予測結果が干渉される現象である。そこで、BTB エントリが存在しない場合に、分岐を不成立と予測し、さらに、競合を緩和するために、分岐方向予測器にエントリを用意しない分岐予測手法を提案する。

本稿の構成を次に示す。2 は、関連研究、3 は、実験環境、4 は、BTB エントリ有無を参照した予測器について述べる。5 は、まとめである。

## 2. 関連研究

シミュレーション対象とした分岐方向予測器と、BTB と組み合わせて動作する既存予測分岐予測器について説明する。

### 2.1 主要な分岐方向予測器

分岐方向予測器は、単一の予測表で構成される単体予測器と複数の予測表で構成されるハイブリッド予測器に分類できる。

#### 2.1.1 単体予測器

単体予測機構は、ハイブリッド予測機構の構成要素となる予測機構である。代表例として、Bimodal 予測機構<sup>4)</sup>、Gshare 予測機構<sup>5)</sup>が挙げられる。Bimodal 予測機構は、分岐命令アドレスの下位ビットを予測表のインデクスとする。これに対して、Gshare 予測機構では、分岐命令アドレス XOR (Global 分岐履歴) をインデクスとする。

#### 2.1.2 ハイブリッド予測器

主なハイブリッド予測器として、分岐傾向によって予測表を選択する Bi-Mode 予測器<sup>6)</sup>と信頼度によっ

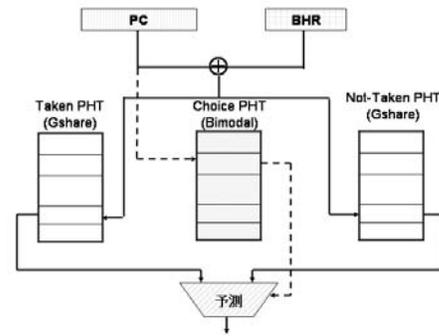


図 1 Bi-Mode 分岐予測器<sup>6)</sup>  
Fig. 1 Bi-Mode Branch Predictor<sup>6)</sup>

て予測表を選択する Combining 予測器<sup>5)</sup>を説明する。Bi-Mode 予測器<sup>6)</sup>

Bi-Mode 予測機構は、分岐命令アドレスによってインデクス付けられた Choice PHT、(分岐命令アドレス XOR Global 分岐履歴) によってインデクス付けられた、2 つの Direction PHT で構成される (図 1)。Choice PHT によって、Direction PHT を選択し、Direction PHT が予測を行う。つまり、Choice PHT において Taken と予測される時、Taken PHT における予測が採用され、Choice PHT において Not-Taken と予測される時、Not-Taken PHT における予測が採用される。Choice PHT は、常に更新されるが、Direction PHT は、予測結果を採用された PHT のみ更新される。

### Combining 予測器<sup>5)</sup>

Combining 予測器は、複数の単体予測器を予測表として組み合わせて構成されている (図 3)。予測の成功・失敗で遷移する 2-bit 飽和カウンタの表である選択器によって、採用する予測表を決定する。予測表の採用・不採用に関わらず、常に予測表を更新する。

### 2.2 BTB を利用した分岐方向予測器

BTB を利用した分岐方向予測器として、BTB に格納された命令アドレスを参照する予測器<sup>7)</sup>や、BTB のエントリに分岐傾向ビットを付加した予測器<sup>8),9)</sup>が挙げられる。

BTB に格納された命令アドレスを参照する予測機構には、PAx 予測器や XAp 予測器がある。PAx 予測器は、分岐命令アドレス毎の分岐履歴、XAp 予測

本稿で実験対象とした Combining 予測器である Gshare 予測器と Bimodal 予測器を組み合わせた構成を示す。

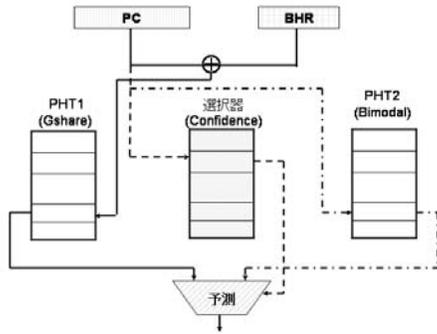


図 2 Combining 分岐予測器<sup>5)</sup>

図 3 Combining Branch Predictor<sup>5)</sup>

器は、分岐命令アドレス毎の予測表で構成される 2 レベル適応型予測器である。これらの予測器では、分岐履歴や予測表の割当て時に参照する分岐命令アドレスを BTB の格納内容を共有することでも実現できる<sup>7)</sup>。

BTB エントリに分岐傾向ビットを付加した予測器では、Chang らの予測器<sup>8)</sup>では、BTB ヒット状況と分岐傾向ビットの信頼度に基づいて、分岐方向予測器の予測、もしくは、分岐傾向ビットを予測として採用する。Sprangle らの Agree 予測器<sup>9)</sup>では、BTB エントリ毎に予め設定された分岐傾向ビットに基づいて飽和カウンタの遷移方向と予測状態を反転する。

### 3. 実験環境

シミュレーターは、SimpleScalar 3.0c/PISA sim-bpred<sup>10)</sup>、ベンチマークは、SPECint95(train 入力) (表 1) を採用した。プログラムは、gcc 2.7.2.3 -O2 -funroll-loops でコンパイルされている。BTB は、4-way set associative 512 エントリ、シミュレーション対象分岐方向予測器は、1KB、8KB Bimodal 予測器 (Bimodal)、Gshare 予測器 (Gshare)、1.5KB、12KB Bi-Mode 予測器 (Bi-Mode)、Combining 予測器 (Comb) とした。

### 4. BTB エントリ有無を参照した分岐予測

BTB は、分岐命令に分岐先アドレスをマッピングした表であり、分岐命令の分岐先を予測するために利用される。しかし、分岐条件が不成立である場合、後続の命令が実行されるため、分岐先アドレスを BTB に記録する必要がない。したがって、Always-NotTaken 分岐は、BTB にエントリを持たない。

4.1 BTB エントリ有無毎の分岐傾向と予測ミス率  
BTB は、Always-NotTaken 分岐にエントリを用意しない。そのため、BTB にエントリを持たない分岐は、textbf{Always-NotTaken} 分岐、もしくは、まだ BTB に登録されていない Taken 分岐である。

#### 4.1.1 BTB にエントリのない分岐の種類

BTB にエントリのない分岐には、

- 分岐条件が常に不成立である (Always-NotTaken) 分岐
- 分岐条件が初めて成立した (First-Taken) 分岐
- BTB のエントリ入替えてエントリから除外されていた (BTB-Replace) 分岐

が考えられる。BTB は、分岐条件が成立した分岐しかエントリを持たないことを考えると、BTB のエントリを保持するためには、分岐は、Taken でなければならない。つまり、BTB-Replace 分岐は、長い間分岐命令自体が実行されなかったか、NotTaken 方向に偏向している (NotTaken 偏向) 分岐であると考えられる。以上からも、BTB にエントリのない分岐が、NotTaken 偏向分岐である可能性が高いと推察できる。

#### 4.1.2 解析結果

表 2 に、BTB にエントリがない動的分岐の割合、分岐傾向と、分岐方向予測ミス率を示す。BTB にエントリがない動的分岐は、全条件分岐の 5.52 ~ 44.10% を占める。また、BTB にエントリがない場合、動的分岐の分岐条件が成立する割合は、どのプログラムにおいても、1%未満になる。このことから、BTB にエントリを持たない分岐は、NotTaken 分岐である確率が高いことが判る。

次に、BTB にエントリを持たない分岐を NotTaken と予測した場合の予測ミス率低減について考える。BTB にエントリを持たない分岐の分岐方向予測ミス率が、BTB にエントリがない動的分岐の分岐条件の成立割合よりも大きい場合に、BTB にエントリを持たない分岐を NotTaken と予測することで、予測ミス

表 1 採用ベンチマーク [100 万命令]  
Table 1 SPECint95(train)[million instructions]

プログラム	入力	全命令	条件分岐命令
099.go	50 9 2stone9.in	554	63
124.m88ksim	dcrand	114	14
126.gcc	amptjp.i	1,280	194
129.compress	1000 q 2131	36	4
130.li	train.lsp	182	24
132.jpeg	vigo.ppm	1,407	87
134.perl	jumble	2,293	299
147.vortex	persons.250	2,608	287

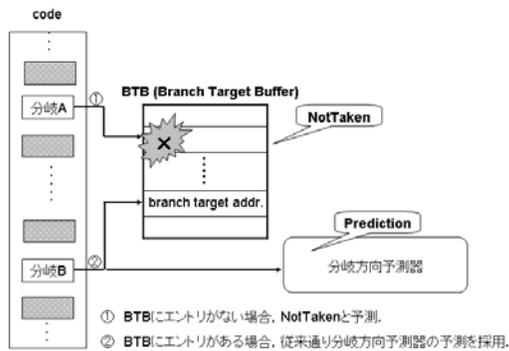


図 4 BTB エントリ有無を参照した分岐方向予測手法  
Fig. 4 Branch Predictor referring a BTB Entry Existence

率が低減する。BTB にエントリがない動的分岐の分岐条件の成立割合は、殆どの場合、BTB にエントリを持たない分岐の分岐方向予測ミス率よりも小さい。そのため、BTB にエントリを持たない分岐をNotTaken と予測するだけでも、予測ミス率は低減する。1.0KB Gshare 予測器では、2.208%予測ミス率が低減する。

#### 4.2 分岐方向予測器更新抑制の与える影響

BTB にエントリがない分岐を分岐方向予測器更新対象から除外することで、NotTaken に偏向する分岐が少なくなり破壊的競合が緩和される可能性を確かめる。

##### 4.2.1 提案予測手法

4.1 節から、BTB にエントリのない分岐は、Not-Taken である確率が高いことが判った。そこで、BTB にエントリのない分岐を分岐方向予測対象から除外することで、予測器に競合を緩和することを目的とした予測手法 (図 4) を提案する。

具体的な予測と更新は、次のようになる。

BTB にエントリが存在しない場合 NotTaken と予測する。分岐がTaken であった (BTB にエントリが登録された) 場合だけ、分岐予測カウンタが更新される。分岐履歴は、常に更新される。

BTB にエントリが存在する場合 分岐方向予測器の予測に従う。分岐予測器は常に更新される。

分岐がTaken であった場合、BTB にエントリが登録される。次に同じ分岐が実行される時、分岐予測対象となるので、分岐方向予測器を更新した。

##### 4.2.2 分岐予測ミス率の低下

表 3 に、BTB にエントリがない分岐をNotTaken と予測し、分岐方向予測器の更新対象から除外した場合の予測ミス率を示す。8KB Bimodal 以外のすべての予測器で、予測ミス率が低減した。Bimodal と

BTB のインデックスは分岐命令アドレスで共通であり、シミュレーションに用いた BTB が 4 エントリであった。この BTB のエントリは、8KB Bimodal の 32,768 エントリと比べて小さく、Bimodal の分岐予測精度の方が高かった。そのため、BTB にエントリがない分岐もBimodal で予測した方が予測ミス率が低減したと考えられる。

##### 4.2.3 分岐方向毎の低減予測ミス数

BTB にエントリのない分岐をNotTaken と予測することによって得られる予測ミス率低減は、表 2 に示す通りである。つまり (BTB にエントリのない動的分岐の予測ミス数) - (BTB にエントリがない分岐の分岐条件が成立する回数) によって、BTB にエントリのない分岐をNotTaken と予測することによる予測ミス低減数になる。

最も予測ミス率が高いプログラムである 099.go を対象に、Gshare, Bi-Mode の分岐方向毎の予測ミス低減数を、図 5 に示した。この予測ミス低減数には、BTB にエントリがなかった場合、NotTaken と予測した分も含まれる。BTB にエントリがない分岐の分岐結果で、分岐方向予測器を更新しなかった場合、Taken 分岐の予測ミス低減数の方が、NotTaken 分岐の予測ミス低減数よりも大きい。これは、NotTaken 偏向分岐に利用される分岐方向予測器エントリの減少によって、Taken 偏向分岐との破壊的競合が減少したためと考えられる。しかし、NotTaken 分岐の予測ミス低減数は、予測器更新抑制前よりも、小さくなっている。これは、NotTaken 偏向分岐の更新が抑制されたため、予測器がTaken と予測する傾向を持ち、NotTaken 分岐の予測を失敗し易くなったためと考えられる。

##### 4.2.4 予測表の使用割合

ハイブリッド予測器は、複数の予測表で構成される。BTB にエントリのない分岐の予測器に対する更新抑制が、予測表の採用に与える影響を調べる。図 6 に、go における予測表の使用割合を示す。

##### Bi-Mode 予測器

Bi-Mode 予測器は、Bimodal 予測器 (Choice PHT) の予測によって、採用する予測表を決定する。BTB にエントリのない分岐の更新抑制によって、Bimodal 予測器が、NotTaken を示す頻度が減少する。そのため、Taken PHT の使用回数が増加する。予測表毎の予測ミス率には、特に何の傾向も見られなかった。

##### Combining 予測器

Combining 予測器は、選択器の示す信頼度によって、採用する予測表を決定する。採用している予測表が予

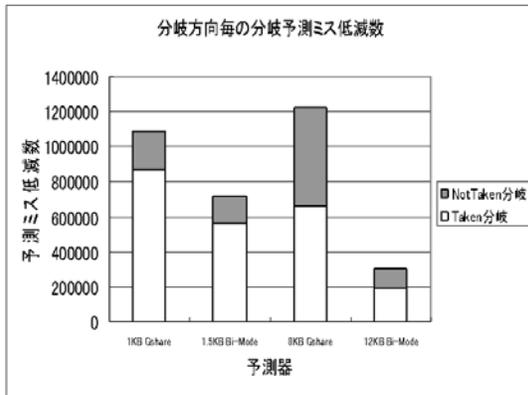


図 5 分岐方向毎の予測ミス低減数  
Fig. 5 Branch Prediction Miss Decrease on Branch Direction

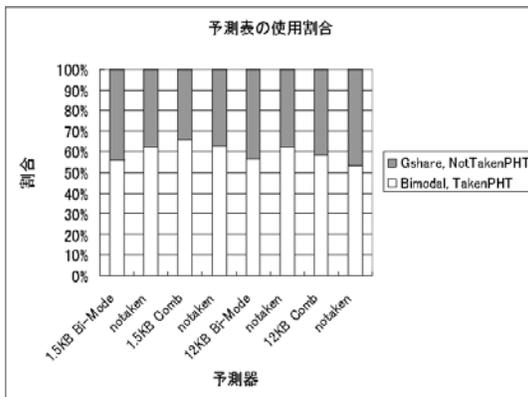


図 6 予測表の採用割合  
Fig. 6 Used Pattern History Table Ratio

測ミスし、かつ、採用しなかった予測表が予測成功している場合に、採用している予測表の信頼度が低下する。BTB にエントリのない分岐の更新抑制によって、Gshare 予測器の使用割合が増えている。また、選択ミス率（採用予測表が予測ミスした場合、不採用予測表の予測が正解している確率）は、低下した（表 4）。

## 5. ま と め

分岐予測は、分岐条件の成立・不成立の予測と分岐先アドレスの予測で構成される。分岐先アドレス予測に利用される分岐先アドレスバッファ(BTB: Branch Target Buffer) は、分岐条件が成立する (Taken) 分岐をエントリに登録する。BTB のエントリに存在しない分岐には、Always-NotTaken 分岐、First-Taken 分岐、BTB-Replace 分岐が考えられる。Always-NotTaken 分岐は、NotTaken 偏向分岐であり、textbfBTB-Replace 分岐は、NotTaken 偏

向分岐の可能性がある。BTB のエントリに存在しない分岐の分岐条件成立・不成立状況を解析した結果、BTB にエントリを持たない分岐の殆どが、分岐条件が成立しない (NotTaken) 分岐であることが判った。

そこで、BTB にエントリを持たない分岐を NotTaken と予測し、さらに、分岐方向予測器の予測対象としない分岐予測手法を考えた。BTB にエントリのない分岐の分岐方向予測器に対する更新を抑制することで、分岐方向予測器のエントリ数が削減し、競合が緩和される。SimpleScalar 3.0c/PISA sim-bpred シミュレータで実験したところ、SPECint95(train) では、8KB Gshare 予測器で平均 1.5%、1.5KB Bi-Mode 予測器で平均 0.4%の予測ミス率が低減した。

謝辞 本研究の一部は、21世紀 COE プログラム「プロダクティブ ICT アカデミア」によるものである。

## 参 考 文 献

- 1) Sprangle, E. and Carmean, D.: Increasing Processor Performance by Implementing Deeper Pipelines, *Proc. of 29th ISCA*, pp.25–36 (2002).
- 2) Lee, J. K. F. and Smith, A. J.: Branch Prediction Strategies and Branch Target Buffer Design, *IEEE Trans. on Comput.*, Vol. 17, No. 1, pp. 6–22 (1984).
- 3) Young, C., Gloy, N. and Smith, M.D.: A Comparative Analysis of Schemes for Correlated Branch Prediction, *Proc. of 22nd ISCA* (1995).
- 4) Smith, J. E.: A Study of Branch Prediction Strategies, *Proc. of 8th ISCA*, pp. 135–148 (1981).
- 5) McFarling, S.: Combining branch predictors, Technical Report TN-36, Digital Western Research Laboratory (1993).
- 6) Lee, C. C., Chen, I. K. and Mudge, T. N.: The Bi-Mode Branch Predictor, *Proc. of MICRO-30*, pp. 4–13 (1997).
- 7) Yeh, T. Y. and Patt, Y. N.: A Comparison of Dynamic Branch Predictors that use Two Levels of Branch History, *Proc. of 20th ISCA*, pp. 257–266 (1993).
- 8) Chang, P.-Y., Evers, M. and Patt, Y. N.: Improving Branch Prediction Accuracy by Reducing Pattern History Table Interference, *Proc. of PACT* (1996).
- 9) Sprangle, E., Chappell, R. S. et al.: The Agree Predictor: A Mechanism for Reducing Negative Branch History Interference, *Proc. of 24th ISCA*, pp. 284–291 (1997).
- 10) Burger, D. and Austin, T. M.: The SimpleScalar Tool Set, Version 2.0, Technical report (1997).

表 2 BTB エントリのない動的分岐の分岐条件成立確率と予測ミス率 (%)

Table 2 Taken Rate and Branch Miss Rate on the Dynamic Branch without a BTB Entry(%)

予測器	099.go	124.m88ksim	126.gcc	129.compress	130.li	132.jpeg	134.perl	147.votex
BTB にエントリのない動的分岐の全条件分岐に占める割合								
none	8.99	14.75	22.94	15.06	20.53	5.52	23.40	44.10
分岐条件が成立(Taken) し, かつ, BTB にエントリのない動的分岐の全条件分岐に占める割合								
none	0.554	0.007	0.402	0.008	0.002	0.007	0.000	0.022
BTB にエントリのない動的分岐の全条件分岐を基底とした分岐予測ミス率								
1.0KB Bimodal	0.659	0.224	0.671	0.012	0.006	0.022	0.062	0.335
1.0KB Gshare	1.649	0.544	2.710	0.139	0.207	0.791	0.457	1.045
1.5KB Bi-Mode	1.501	0.056	1.235	0.041	0.075	0.084	0.070	0.237
1.5KB Comb	0.923	0.024	0.887	0.013	0.002	0.017	0.004	0.237
8.0KB Bimodal	0.339	0.006	0.235	0.009	0.002	0.003	0.000	0.016
8.0KB Gshare	1.859	0.130	1.400	0.093	0.063	0.125	0.083	0.225
12KB Bi-Mode	0.885	0.008	0.551	0.022	0.018	0.833	0.003	0.022
12KB Comb	0.358	0.006	0.293	0.009	0.002	0.005	0.031	0.017

表 3 BTB エントリのない動的分岐の更新を抑制した分岐予測ミス率 (%)

Table 3 The Branch Miss Rate of the Branch Predictor

exclusing Dynamic Branch without a BTB Entry(%)

予測器	099.go	124.m88ksim	126.gcc	129.compress	130.li	132.jpeg	134.perl	147.votex	平均
従来予測器									
1.0KB Bimodal	24.91	5.51	13.04	14.42	12.78	11.53	8.61	2.11	11.61
1.0KB Gshare	29.38	3.02	14.16	8.26	6.44	11.53	4.48	3.33	10.07
1.5KB Bi-Mode	24.61	2.14	9.67	8.05	6.30	10.02	3.47	1.20	8.18
1.5KB Comb	23.03	1.97	10.03	8.40	6.47	9.99	3.78	1.38	8.13
8.0KB Bimodal	23.90	5.21	12.05	14.42	12.77	11.49	9.01	1.34	11.27
8.0KB Gshare	21.89	1.92	8.29	7.28	5.20	9.62	2.45	1.23	7.24
12KB Bi-Mode	18.04	1.66	5.95	7.53	5.21	10.02	2.34	0.51	6.41
12KB Comb	18.70	1.57	6.19	7.93	5.48	9.43	2.42	0.62	6.54
BTB エントリのない動的分岐の更新を抑制した予測器									
1.0KB Bimodal	24.61	5.26	12.50	14.42	12.77	11.51	7.92	1.61	11.32
1.0KB Gshare	27.64	2.03	10.29	8.06	6.08	10.50	3.74	0.94	8.66
1.5KB Bi-Mode	23.47	2.06	8.22	8.00	6.14	9.89	3.41	0.74	7.74
1.5KB Comb	22.25	1.91	8.77	8.37	6.37	9.90	3.44	0.80	7.73
8.0KB Bimodal	24.13	5.18	12.11	14.42	12.77	11.50	7.92	1.33	11.17
8.0KB Gshare	19.93	1.66	6.73	7.17	5.11	9.41	2.31	0.57	6.61
12KB Bi-Mode	17.55	1.66	5.74	7.51	5.17	9.17	2.34	0.50	6.21
12KB Comb	18.58	1.55	6.12	7.92	5.46	9.39	2.40	0.56	6.50

表 4 選択ミス率 [099.go](%)

Table 4 PHT Selection Miss Rate[099.go](%)

予測器容量	選択ミス率	
	Comb	提案
1.5KB	53.73	52.81
12KB	56.30	55.77