

## 分岐パス情報に基づく分岐予測方式の提案

岩田 靖 勝野 昭 木村 康 則

高性能パイプラインプロセッサにおいて、分岐命令に起因するパイプラインハザードを回避し、効率良くプロセッサを動作させるためには高精度な分岐予測方式を導入することが必要である。分岐予測方式はパイプライン段数の深いプロセッサや、スーパースカラプロセッサのように複数のパイプラインを持つプロセッサではますます重要となる。現在までに提案されている予測方式は、予測する分岐命令の過去の分岐履歴を予測情報として用いる。本報告では分岐予測を行うために用いる情報として、分岐パス情報を導入し、予測精度を向上させる方法を提案する。

### Branch Prediction Method Based on Branch Path Models

YASUSHI IWATA, AKIRA KATSUNO and YASUNORI KIMURA

It is necessary to introduce accurate branch prediction methods for pipeline processors in order to achieve high performance. Accurate branch prediction methods are especially required for super scalar processors and deeply pipelined processors. It has been proposed the prediction method that uses branch prediction histories for each branch instructions. This paper presents new branch prediction methods which utilize branch path histories.

#### 1. はじめに

実行プログラム中の分岐命令がパイプラインプロセッサに投入されると、分岐方向が確定するまで後続命令が決定されない。分岐方向の結果は一般に分岐命令の実行直前まで確定しないため、その間はパイプラインが停止する状態、いわゆるパイプラインハザードが発生する。現在の高性能プロセッサのパイプライン段数は増加する傾向にあり、このようなパイプライン段数の長いプロセッサにおいては、分岐命令に起因するパイプラインハザードはプロセッサの性能を劣化させる。さらに、スーパースカラプロセッサなどの複数のパイプラインを持つ最近のプロセッサではさらにその問題は深刻である。

分岐命令によるパイプラインハザードを回避し、プロセッサの性能劣化を抑えるためには、分岐予測と後続命令の投機的実行が重要である。分岐予測方式は一般に静的な分岐予測方式と動的な分岐予測方式がある。静的な分岐予測方式はコンパイラにより実現される。動的な分岐予測方式は実行時にハードウェアにより分岐方向を予測する。近年、動的な分岐予測方式の研究<sup>1)</sup>がさかんであり、現在までにいくつかの動的な分岐予測方式が提案されている。最も基本的な分岐予測方式は2ビット予測方式<sup>2)</sup>である。これは分岐命令のPC(Program Counter)と、そのPCの分岐しやすさの傾向を示す2

ビットで表された「状態」を1対1に対応させるテーブルを参照することによって分岐予測を行うものである。さらに予測精度を高める目的で最近になり2レベル分岐予測<sup>3)</sup>と呼ばれる予測方式が提案されている。これは分岐命令の過去数回の分岐結果の履歴と、これらの履歴ごとに分岐しやすさの指標となる2ビットで表された「状態」を参照するものである。2レベル予測で達成される予測確率は95%程<sup>4)</sup>ある。

図1に分岐予測の正解確率とある構成のスーパースカラプロセッサの性能の関係をIPC (Instructions Per Cycle) を尺度として示している。図1からわかるように、分岐予測正解確率の伸びに対するプロセッサの性能(IPC)の伸び(グラフの傾き)は、90%から100%の区間が大きく、現在2レベル分岐予測により達成されている予測精度をさらに上回することはプロセッサの性能を向上させる上で重要である。

以上の観点から本報告では、2レベル分岐予測を上回る予測精度を達成させるために、分岐パス情報に基づく分岐予測方式を提案する。本提案の分岐予測方式は、予測情報として分岐パス情報を用いている。本報告では第2章でまず現在までに提案されている2レベル分岐予測方式を整理し、その上で第3章において本提案のパス情報に基づく分岐予測方式について述べる。そして第4章においてシミュレーション実験により、2レベル分岐予測と本提案の分岐予測方式の予測確率を明らかにする。さらに第5章において、予測が容易である分岐履歴の系列とそれが困難である系列を明らかにする。最後に第6

† (株)富士通研究所  
Fujitsu Laboratories LTD

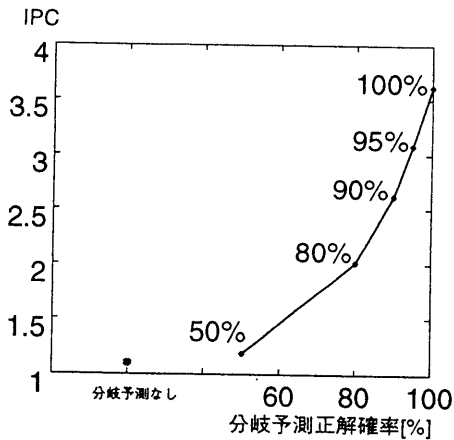


図1 分岐予測の正解確率とIPCの関係

章で本報告の全体をまとめる。

## 2. 2レベル分岐予測方式

2レベル分岐予測方式<sup>3)</sup>は、分岐履歴テーブルに格納された各分岐命令ごとの分岐履歴をインデックスにして、2ビットの飽和型カウンタに保持された「状態」を参照することにより分岐方向を決定する。図2を例にして2レベル分岐予測方式のアルゴリズムを説明する。

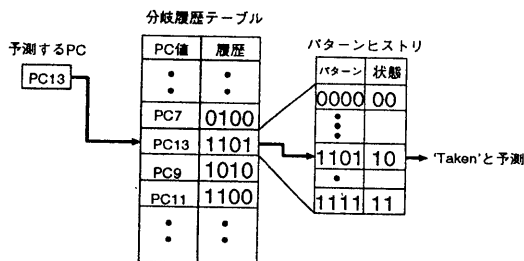


図2 2レベル分岐予測方式

現在予測しようとしている分岐命令のPC(Program Counter)がPC13だとする。2レベル分岐予測ではまず、分岐命令のPC値をエン트리とする分岐履歴テーブルを参照することにより、この分岐命令の分岐履歴を知る。図2の場合、分岐履歴テーブルを参照することにより現在予測しようとするPC13の過去の分岐履歴は“1101”であることがわかる。ここで“1”は分岐成立(Taken)を示し“0”は分岐不成立(Not Taken)を表している。2レベル分岐予測ではさらに、この分岐履歴のパターンをインデックスにして当該分岐命令の分岐しやすさを示す「状態」を参照する。分岐命令の分岐履歴パターンとその状態はパターンヒストリテーブルに対応づけて格納している。最終的にその分岐命令が分岐成立するか不成立かの判断はこの状態をもとにして行う。こ

の状態は2ビットで表される4状態オートマトンを形成し、状態が00と01の時はともに分岐不成立と予測し、10と11の時はともに分岐成立と予測する<sup>6)</sup>。

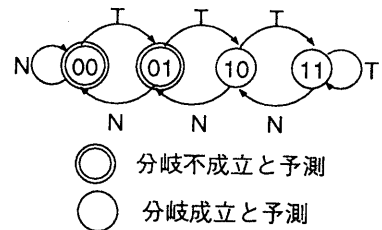


図3 状態更新アルゴリズム

図3にこの状態更新アルゴリズムを示す。これは実際の分岐が成立した場合は現在の状態を1だけインクリメントし、分岐不成立だった場合は状態を1だけデクリメントする。これは2ビットの飽和型アップダウンカウンタにより容易に実現することができる。このように2レベル分岐予測は2段階にテーブルを参照することにより分岐方向を予測する。2レベル分岐予測方式において、各分岐命令の分岐履歴パターンを参照せずに分岐命令のPCとそのPCにおける状態を直接参照する方式は2ビット予測方式と呼ばれる<sup>2)</sup>。

## 3. パス情報に基づく分岐予測方式

### 3.1 分岐履歴情報

2レベル分岐予測方式において、予測を行うために利用している最も重要な情報は、各分岐命令ごとに保存している過去の分岐履歴情報である。この分岐履歴情報は分岐履歴テーブルに各PCごとに格納して参照する。すなわち、各PCごとに独立した分岐履歴情報を保持している。本報告では、このような各PCごとの分岐履歴情報をローカルヒストリと呼ぶことにする。それに対して、もう1つの分岐履歴を考慮することができる。すなわちそれは現在予測しようとしている分岐命令に至るパス情報である。これを本報告ではグローバルヒストリと呼ぶことにする。図4に当該分岐命令に至るパスモデル(グローバルヒストリモデル)を示す。図4においてノードは分岐命令をアークは分岐方向を示す。

図4のモデルでは分岐命令 br.0 から br.3cへ至るパスすなわちグローバルヒストリは011, 101および110の3種類があることがわかる。以降の節では、ローカルヒストリとグローバルヒストリを用いて分岐予測を行う方式を3レベル分岐予測方式と呼び、その説明を行う。

### 3.2 3レベル分岐予測方式

#### 3.2.1 パターンベース3レベル分岐予測

図5にグローバルヒストリとローカルヒストリの関係を示す。図5では現在予測を行おうとしている分岐命令へ至るグローバルヒストリが011であるとする。3

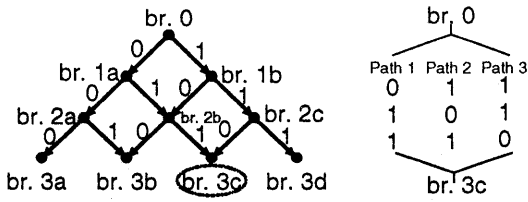


図4 予測する分岐命令へ至るパスモデル (1)

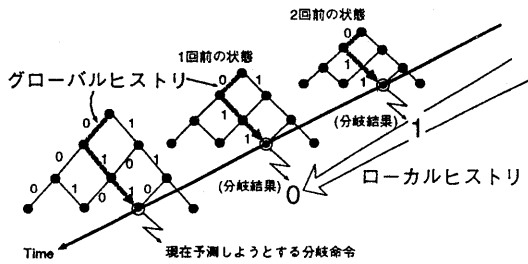


図5 ローカルヒストリとグローバルヒストリ

レベル分岐予測ではこのグローバルヒストリを辿ったという条件の時、当該分岐命令はどのように分岐成立と分岐不成立を繰り返してきたかという情報（ローカルヒストリ）を用いる。

図6に3レベル分岐予測方式における予測方式を示す。現在予測しようとしている分岐命令のPCをPC13とする。その時、PC13に至るグローバルヒストリ（図では1101）を参照する。さらに、その時のローカルヒストリを参照する。ここで参照するローカルヒストリはグローバルヒストリが1101という場合におけるローカルヒストリであることに注意する。最後に実際の予測方向を2ビットで表された「状態」を参照することによって決定する。この「状態」は2章で示した2レベル分岐予測方式と同様にして、実際の分岐結果を基にして更新する。以上示したように、本提案の3レベル分岐予測方式は3段階のテーブル参照を行う。

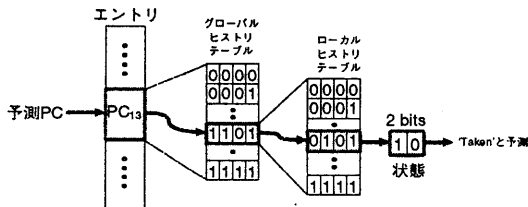


図6 3レベル分岐予測方式の概念図

3レベル分岐予測と2レベル分岐予測の最も大きな差異は、2レベル分岐予測ではグローバルヒストリに関わらずローカルヒストリだけに注目しているのに対して、本提案の3レベル分岐予測では、グローバルヒストリの違いによる当該分岐命令の振舞いを場合分けしていると

ころが本質的に異なる点である。これによりローカルヒストリの規則性を抽出することを狙いとしている。以上示したように、当該分岐命令に至るパス情報をグローバルヒストリとして、0と1のパターンの系列で取得し利用している3レベル分岐予測方式のことを本報告では特にパターンベース3レベル分岐予測方式と呼ぶことにする。

### 3.2.2 PCベース3レベル分岐予測方式

3.2.1節に示したパターンベース3レベル分岐予測方式では、グローバルヒストリを0と1のパターンとして保存し、利用している。これに対してグローバルヒストリを当該分岐命令に至るまでの分岐命令のPC値そのもので保存し、利用することも考えられる。本報告では、グローバルヒストリを分岐命令のPC値そのもので保存し、利用する予測方式を前述のパターンベース3レベル分岐予測と区別するために、PCベース3レベル分岐予測方式と呼ぶことにする。PCベース3レベル分岐予測方式では、パターンベース3レベル分岐予測方式では分類不可能であるグローバルヒストリも分類し区別することが可能となる。その例を図7に示す。

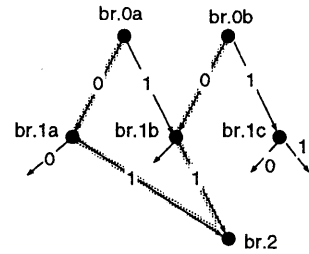


図7 パスモデル (2)

図7において、現在br.2における分岐方向を予測すると仮定する。br.0aとbr.0bから出発しbr.2へ至るパスは図7に示すパスモデルの場合には2本ある。グローバルヒストリをパターンベースで保存する場合、そのどちらも01と表される。すなわちパターンベースでグローバルヒストリを保存する場合は、この例のような実際に経由する異なる2本のパスを区別することができない。これに対してグローバルヒストリを経由する分岐命令のPC値として直接保存する場合は、これらの区別が可能となる。たとえば今の場合、分岐命令br.0、br.1およびbr.2のそれぞれのPCがbr.0、br.1およびbr.2であるとすれば、PCベースでグローバルヒストリを保存する場合、これら2本のパスはそれぞれ“br.0、br.1a、br.2”と“br.0a、br.1b、br.2”のように区別することが可能となる。これによって、当該分岐命令へ至るパスの差異が予測しようとしている分岐命令の振舞いをより正確に分類することが可能となる。そしてさらに、このような考え方を導入することで予測確率そのものの向上が期待できる。

PC ベース 3 レベル予測方式もパターンベース 3 レベル予測方式と同様にして図 8 のように考えることができる。図 8 では当該分岐命令の 4 回前の分岐バス情報を

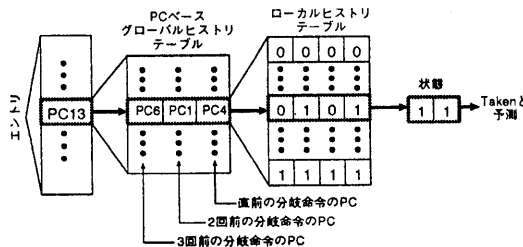


図8 PC ベース 3 レベル分岐予測方式

PC ベースで保存する例を示している。PC ベースでグローバルヒストリを保存する場合、一般に PC は 32 ビット程度のビット数で表されるため、パターンベースでそれを保存する場合に比較してより多くのビット数(すなわちハードウェア量)が必要となる。

#### 4. 実験結果

##### 4.1 予測正解率

表 1 に 3 レベル分岐予測の正解率を、現在までに提案されている 2 ビット予測および 2 レベル分岐予測のそれと比較する。さらに、表 1 のデータを図 9 にグラフとして示す。

表1 分岐予測の正解率の比較

	2bit	2 レベル L(4)	3 レベル分岐予測	
			パターン G(4) +L(4)	PC PC(32) +L(4)
gcc	88.26%	91.87%	93.37%	92.62%
compress	89.63%	90.84%	91.10%	90.98%
li	89.16%	95.00%	96.62%	95.26%
eqntott	92.95%	95.48%	95.85%	95.81%
espresso	85.88%	94.53%	95.80%	95.06%
sc	96.25%	97.85%	97.16%	97.95%
平均	90.35%	94.26%	95.14%	94.61%

表 1 の結果は、SPECint92 のベンチマークプログラムにおいて 10M 回の分岐命令について実験を行い、予測正解率を得ている。表中において L(4) や G(4) としているのは、ローカルヒストリとグローバルヒストリのビット数をそれぞれ 4 ビットとしていることを表している。さらに、PC ベース 3 レベル分岐予測において、PC(32) と示しているのは、32 ビットで表される PC を全ビット完全に保持していることを示している。ここで実験している PC ベース 3 レベル分岐予測では、保存する PC は予測を行う PC の一段だけ上位の PC である。また、上位 PC のバリエーション数はプログラム

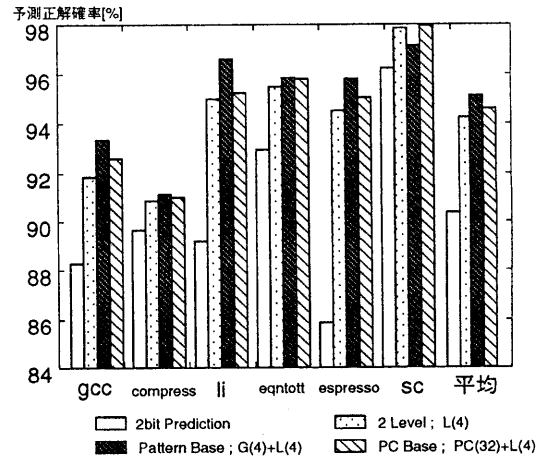


図9 正解率の比較

上で制限していない。すなわち、インプリメンテーション時のハードウェア量を考慮せず、出現する PC を全て保存するモデルによりシミュレーションを行っている。

表からわかるように、パターンベース 3 レベル分岐予測方式では、予測に利用する情報としてグローバルヒストリを 4 ビット加えることにより 0.88% の予測率の向上が認められる。また、PC ベース 3 レベル予測として、上位の一段だけ遡った PC の情報を分岐予測の情報として付加することにより、0.35% の予測率の向上が認められる。この結果を考えると、顕著な予測率の向上は得られていない。その理由は現在の実験では、保存するバス情報を上位の一段だけの PC としていることが考えられる。

図 10 では、パターンベース 3 レベル分岐予測において、ローカルヒストリとグローバルヒストリのビット長をともに等しくし、そのときの SPECint92 の各ベンチマークプログラムに対して予測正解率の平均値を 2 レベル分岐予測のそれと比較している。図 10 から、ヒス

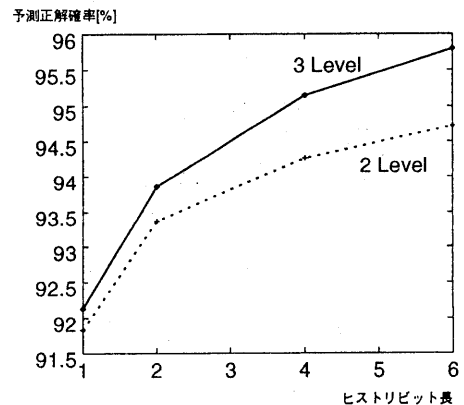


図10 ヒストリビット長と予測正解率の関係

トリビットを長く取る場合、予測正解確率が3レベルの場合でも2レベルの場合でも向上することがわかる。

3レベル分岐予測では、予測を行うためにグローバルヒストリを導入しているため、テーブルサイズすなわちハードウェア量が増加する。3レベル分岐予測のハードウェア量をテーブルサイズを評価の尺度として、2レベル分岐予測と2ビット予測のそれとの比較を、表2に示す。表2において、 $k$ と $l$ はそれぞれ3レベル分岐予

表2 テーブルサイズの比較

	Table Size[bit]	Entry=256, L=G=4
3 Level 予測	$m2^{k+l+1}$	131Kbit
2 Level 予測	$m2^{p+1}$	8Kbit
2 bit 予測	$m2$	512Bit

測のグローバルヒストリとローカルヒストリのビット長であり、 $p$ は2レベル分岐予測のローカルヒストリのビット長である。また、 $m$ はテーブルのエントリ数である。表からわかるように、3レベル分岐予測のテーブルサイズは2レベル分岐予測のそれより $2^k$ 倍になる。そこで、テーブルサイズを等しくした場合の3レベル分岐予測と2レベル分岐予測の予測正解確率を調べる。

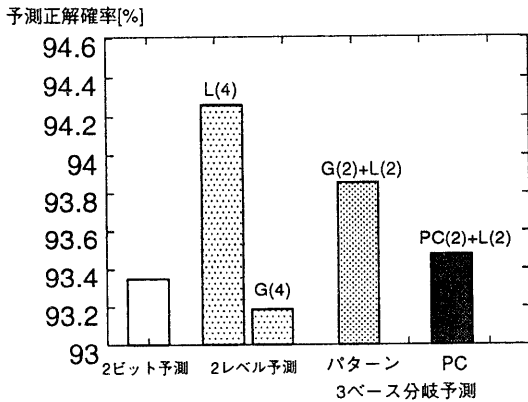


図11 予測正解確率の比較

図11に3レベル分岐予測のローカルヒストリとグローバルヒストリをともに2とし、それと同じテーブルサイズを有する2レベル分岐予測としてローカルヒストリを4とした場合を比較する。この条件の時、図11に示すように、パターンベース3レベル分岐予測の正解確率が2レベルのそれよりも低くなっている。さらに、PCベース3レベル分岐予測において、下位2ビットだけを保存した場合も、2レベル分岐予測と比較し、予測正解確率が下がる。

#### 4.2 グローバルヒストリの特徴

4.1節に示したように、現時点でハードウェア量を等しくした場合、パターンベース3レベル分岐予測もPCベース3レベル分岐予測もどちらも、2レベル分岐予

測に比較して良好な予測正解確率を得ることができない。本節ではPCベース3レベル分岐予測において、そのグローバルヒストリの特徴を調べる。

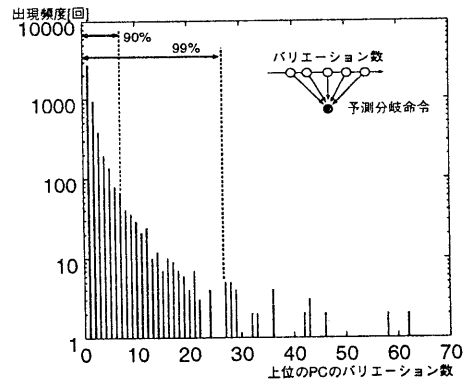


図12 上位のPCのバリエーション数と出現頻度

図12に、予測しようとするある分岐命令の上位の分岐命令のバリエーション数と、出現頻度の関係を表している。図12はアプリケーションとしてgccを選んでいる。100回以上予測を行っている分岐命令のPCのみを数えあげている。図12に示すように、上位のPCのバリエーション数を7とすれば全体のバリエーションの90%をカバーし、同様に27とすれば全体の99%をカバーする。

## 5. 考 察

3レベル分岐予測方式において、高い確率で予想が正解する分岐命令のローカルヒストリの例と、正解確率が低い分岐命令のローカルヒストリの例を以下に示す。

- 予測確率が高い例。
  - 規則的系列  
TTNN TTTT NNNT TTNN TTTT NNNT TTNN
- 予測確率が低い例。
  - 不規則系列  
NTTN NTNT TTTN TTIN NTNT NTNT NTNN
  - 不連続系列  
TTTT TTNN TTTT TTTT TTTT TTTT TTTT

表3 予測確率

グループ	3レベル分岐予測方式		2レベル
	PCベース	パターン	
規則的系列	91.38%	87.21%	91.4%
不規則的系列	48.43%	51.96%	48.44%
不連続系列	64.43%	72.60%	63.85%

また、表5には、上で示したPCの例における予測正解確率を示す。

予測確率が高い例は、ローカルヒストリが規則的な履歴をたどる場合である。このように高い予測確率を達成することができる分岐命令のローカルヒストリの系列を周期的系列と呼ぶことにする。この例では、4回のT(分岐成立)のあとに、2回のN(分岐不成立)が出現するという周期が認められる。この例では、表5に示すように、3レベル分岐予測でも2レベル分岐予測でもどちらも高い予測正解確率が得られている。もちろん、Tだけ、もしくはNだけが連続して出現する系列では、いずれの予測方式でも正しく予測することができる。

予測確率が低いローカルヒストリの系列は、TとNが不規則に出現するグループと、いくつかのTまたはNが連続した後に不連続なTまたはNが出現するグループに分類することができる。ここでは前者を不規則系列、後者を不連続系列と呼ぶことにする。不連続系列は、どの時点で不連続となるかを正しく予測することができれば、正解確率をあげることができると考えられる。不規則系列がランダムに分岐と非分岐を繰り返すのであれば、どのような方式でも正確には予測することはできない。

本報告で提案している3レベル分岐予測方式は、バス情報を導入することにより、これらの不規則的なローカルヒストリを辿る分岐命令をグローバルヒストリで分類することにより、不規則な系列を有するローカルヒストリから規則性を抽出する作業を行っていることに他ならない。

## 6. ま と め

本報告ではバス情報を予測情報として用いる分岐予測方式として3レベル分岐予測方式を提案した。バス情報の取り扱いに関して、グローバルヒストリを0と1のパターンにより取得し利用する方法と、履歴バスのPC値を直接利用する方式を提案した。ローカルヒストリだけを予測情報として用いる2レベル分岐予測方式に比較して、グローバルヒストリを導入する本提案の3レベル分岐予測方式では、わずかではあるが、予測正解確率の向上が認められる。しかし、ハードウェア量を一定にして2レベル予測のそれと比較した場合は十分な予測精度を達成することができなかった。さらに、これらの予測方式で正確に予測することが可能なローカルヒストリの系列と予測が困難な系列をそれぞれ明らかにし、分岐予測確率をより向上させるために必要な着眼点を整理した。

## 謝 辞

本研究を遂行するにあたり常にご支援・ご指導して下さった、林プロジェクト部長に感謝します。また、貴重なコメントを頂いた津田プロジェクト部長代理に感謝します。たえずディスカッションして頂いたプロセッサ研究部の西本氏、川崎氏をはじめ、研究員諸氏にお礼申し上げます。

## 参 考 文 献

- 1) 山名, 佐藤, 児玉, 坂根, 坂井, 山口, “投機的実行研究の最新動向とタスク間投機的実行の有効性”, 情報処理学会第51回全国大会講演論文集, 1P-3 (1995).
- 2) J.E. Smith, “A Study of Branch Prediction Strategies”, Proceedings of the 8th International Symposium on Computer Architecture, pp. 135-148 (1981).
- 3) T. Yeh and Y. Patt, “Two-Level Adaptive Branch Prediction”, Proceeding of the 24th International Symposium on Microarchitecture, pp. 51-61 (1991).
- 4) L. Gwennap, “New Algorithm Improves Branch Prediction”, Microprocessor Report, pp. 17-21 (March 27 1995).
- 5) T. Yeh and Y. Patt, “Alternative Implementations of Two-Level Adaptive Branch Prediction”, Proceedings of the 19th Annual International Symposium on Computer Architecture, pp. 124-134 (1992).
- 6) R.Nair, “Optimal 2-Bit Branch Predictors”, IEEE Transactions on Computers, Vol. 44, No. 5, pp. 698-702 (1995).