

データ幅を考慮したデータ値予測機構のハードウェア量削減

佐藤 寿倫 有田 五次郎

九州工業大学 情報工学部 知能情報工学科

命令レベル並列性向上のために、演算結果を予測することで真の依存関係を解消するデータ予測の手法が検討されている。そのためには分岐予測機構と同様にテーブルに保持された過去の履歴を利用することが一般的であるが、演算結果のデータ幅は元々大きいため、分岐予測機構と比較してハードウェア量が非常に大きくなる。加えて高い予測成功率を得るためには値履歴表のエントリ数を増やす必要があり、さらにハードウェア量が大きくなるという問題がある。本稿では演算結果の有効なデータ幅の特徴を利用して、データ値予測機構のハードウェア量を削減することを検討する。単純に予測機構に保持されるデータ幅を削減すると予測精度は著しく低下するが、提案する2モード値予測機構を用いることで予測精度の低下を抑えつつハードウェア量削減を達成できる。プロセス性能の評価では、用いたベンチマークプログラムのほとんどで、わずかに低下した予測精度の影響はほとんど無いことが確認できている。

Reducing Hardware Budget of Data Value Predictors by Exploiting Narrow Bitwidth

TOSHINORI SATO and ITSUJIRO ARITA

KYUSHU INSTITUTE OF TECHNOLOGY

Recently, the practice of speculation in resolving data dependences has been studied as a means of extracting more instruction level parallelism (ILP). An outcome of an instruction is predicted by value predictors. The instruction and its dependent instructions can be executed simultaneously, thereby exploiting ILP aggressively. One of the serious hurdles for realizing data speculation is huge hardware budget of the predictors. In this paper, we propose a technique reducing the budget by exploiting narrow width values. Simulation results show that the technique, called 2-mode predictor, maintains processor performance with slight decrease of the value prediction accuracy.

1. はじめに

現在に至る計算機システムの性能向上は目覚しいが、なお一層の性能向上が期待されている。この目的のためには、粗粒度並列処理を利用する並列プロセッサを構成する方法と、細粒度の命令レベル並列処理を利用してプロセッサ単体の性能を向上させる方法が考えられる。われわれは将来のマルチメディア処理の重要性を鑑みて、システムLSIのコアとなるマイクロプロセッサに着目し、命令レベル並列処理の研究を行なっている。

命令レベル並列性はプログラムに内在する並列性であり、制御依存関係とデータ依存関係によって制限される。制御依存関係の解消には従来から分岐予測などの様々な検討が行なわれており、商用のマイクロプロセッサにも広く採用されている。データ依存関係のうち出力依存関係と逆依存関係はレジスタ名前替えにより解消可能であり、動的な命令スケジューリングを行なうハイエンドのマイクロプロセッサで利用されている。データ依存関係に関しても、近年データ値予測とそれに基づいた投機的実行によって解消可能であることがわかり、多方面から注目され多くの検討が開始されている。

しかし現在のところ、データ値予測機構に関する検討は主に予測方式に関してであり、実際にマイクロプロセッサに実装するためのハードウェアに関する考察はほとん

どされていない。これまでに予測精度が高いと報告されている機構はいずれも、およそ現在のマイクロプロセッサには実装できないほどのハードウェア量を必要としている。そのため、データ値予測精度を維持しつつ予測機構のハードウェア量を削減することを検討する必要がある。いくつか報告例も現われつつある^{8),11)}。そこでわれわれも、従来の検討とは異なる角度からデータ値予測機構のハードウェア量を削減することを検討している。

一方、マイクロプロセッサの64ビット化に伴いデータパスのビット幅は64ビットになっているが、Brooksら¹⁾によればほとんどのデータのビット幅は狭く、SPECint95ベンチマークの場合で50%のデータは16ビット以下である。そこでわれわれは、データ値予測機構に保持されるデータのビット幅を制限し、値履歴表のハードウェア量削減を検討する。例えば、予測容易なデータのビット幅が狭く、一方でデータ幅の大きなデータは予測困難であれば、値履歴表に保持されるデータ幅に制限を加えても予測精度にはそれほどの影響は無いと考えられる。

以下に本稿の構成を示す。次節でこれまでに提案されている値予測機構と関連研究をまとめる。3節では評価環境を説明する。4節でデータ幅とデータ値予測精度の関係を調査する。5節でデータ幅を考慮してハードウェア量の削減を検討し評価する。最後に6節で全体のまとめを行ない、今後の課題を述べる。

2. 関連研究

データ予測に基づいたデータ投機実行^{5),7)}は、命令間の依存関係を緩和し命令レベルの並列性を抽出する新たなパラダイムとして注目されている。これまでに様々な予測機構が提案されているが、比較的予測精度の高い2レベル値予測機構¹⁴⁾、ハイブリッド値予測機構¹⁴⁾、そしてコンテキストベース値予測機構¹²⁾はハードウェア量が非常に大きくなるという問題がある。Moranchoら⁸⁾やRychlikら¹¹⁾は値予測機構のハードウェア量を削減することを検討している。演算結果の予測容易性に基づいて命令を分類し、予測容易な命令はハードウェア量の小さな予測機構を使用し、予測困難な命令だけ上述のハードウェア量の大きな予測機構を使用する。つまり、ハイブリッド値予測機構のハードウェア量削減に着目している。しかし、構成要素となる各値予測機構のハードウェア量は削減できてはいない。

分岐予測機構の領域では分岐先バッファのハードウェア量削減が検討されている。Fagin⁴⁾はタグ部のビット幅と予測精度との間のトレードオフを調査し、タグのビット幅を2ビットに制限しても予測精度はほとんど低下しないと報告している。山田ら¹⁸⁾の2レベル表方式はアドレス部のビット幅を制限することを検討しており、その場合に正しく表現できない分岐先アドレスのために上位アドレスを保持する第2の表を用いている。その結果、従来方式の予測精度をほとんど低下させることなくハードウェア量を約40%削減できている。Driesenら³⁾のカスケード予測機構は分岐命令を分岐先アドレスの予測の容易さで分類し、予測容易な分岐命令はハードウェア量の小さな分岐先バッファで予測し、予測困難な命令だけをハードウェア量の大きな2レベル適応型予測機構で予測している。全ての命令を2レベル適応型予測器で予測する場合と比較して、ハードウェア量が1/4~1/2に削減できている。ただし上述した2つの方法とは異なり、カスケード予測機構ではビット幅の制限は検討されていない。

さらにキャッシュのハードウェア量を削減するために、Wangら¹³⁾はCAT(caching address tags)と呼ばれる方法を提案している。タグ部に現われるアドレスの特徴に着目してタグ部をキャッシュし、データレイの各エントリに対応するタグの値はキャッシュされたタグ情報から生成する。これによってタグレイのビット幅が削減できる。

3. 評価環境

本節では、シミュレーションに用いたプロセッサのモデルとベンチマークプログラムを紹介し、提案手法の評価環境について説明する。

3.1 プロセッサモデル

評価には2種類のプロセッサモデルを用いる。データ予測精度は命令レベルの機能シミュレータを用いて評価する。一方、プロセッサ性能の評価にはサイクルレベルのタイミングシミュレータを用いる。どちらのシミュレータもSimpleScalarツールセット(ver.3.0a)²⁾を用いて作

成されており、モデルは32ビットプロセッサである。

データ予測器は、最終値予測機構⁷⁾、ストライド値予測機構^{5),14)}、2レベル値予測機構¹⁴⁾、ストライド値予測と2レベル値予測のハイブリッド値予測機構¹⁴⁾を評価する。ストライド値予測機構の状態遷移機械は文献14)で提案されているものを用いる。さらに、文献14)に倣って、2レベル値予測機構とハイブリッド値予測機構の履歴の数はどちらも6、予測を行なう閾値はそれぞれ3と6にする。

性能評価に用いられるプロセッサモデルはレジスタ更新ユニット(RUU: register update unit)に基づいて動的命令スケジューリングを行なう動的命令スケジューリングを行なうスーパースカラプロセッサであり、以下の構成となっている。RUUの容量は128エントリである。命令フェッチ幅と命令発行幅は8命令であり、8つの機能ユニットは完全に対称であらゆるタイプの命令を実行できる。各演算のレイテンシは乗算(4サイクル)と除算(12サイクル)を除いて全て1サイクルである。データキャッシュは4ポートのノンブロッキングキャッシュで、容量128KB・ブロックサイズ32Bの2ウェイセットアソシアティブで構成される。データキャッシュのレイテンシはデータアドレス計算の他に、ヒット時1サイクル、ミス時6サイクルである。また全てのメモリアクセス命令は、先行するストア命令が完了していなければ実行できない。命令キャッシュは容量128KBで、ブロックサイズ32Bの2ウェイセットアソシアティブキャッシュである。2次キャッシュはデータと命令で共有されており、容量8MB・ブロックサイズ64Bのダイレクトマップキャッシュである。2次キャッシュミス時のレイテンシは、最初のワードが得られるまでが18サイクルで、後続のワードのアクセスにはそれぞれ2サイクルを必要とする。分岐先アドレスの予測にはエントリ数1K・4ウェイセットアソシアティブの分岐先バッファとエントリ数8のリターンアドレススタックを、分岐方向の予測にはエントリ数4Kのgshareタイプ2レベル適応型分岐予測器を用いた。これらの分岐予測器の更新は命令のリタイア時に行なわれる。

値予測に失敗した場合には、正しい実行を保証するために、間違った予測値を用いて投機的に実行された命令を、正しいオペランドで再実行する必要がある。このための回復機構には、予測失敗命令に後続する全命令を破棄して再実行する方法と、予測失敗命令と実際にデータ依存の関係にある命令だけを選択的に再実行する方法がある。前者の方法は予測ミス時のペナルティが大きいいため後者の方法を選択し、文献16)で議論されている実装法をタイミングシミュレータに組み込んだ。

3.2 ベンチマークプログラム

Brooksら¹⁾の報告を検証し本検討の一般性を確認する意味で、本稿では評価の第1段階としてSPEC95の整数系ベンチマークプログラムを用いる。それぞれのプログラムの入力セットは現実的な時間でシミュレーションが終了するように調整してある。プログラムと入力セットを表1にまとめる。これらのプログラムはMIPSアーキテクチャを拡張したSimpleScalar/PISAアーキテクチャ²⁾

をターゲットとしてコンパイルされている。132.jpeg は GNU GCC(version 2.6.3) を用いて最適化オプション -O3 でコンパイルした。残りのプログラムはウイスコンシン大学が配付しているコンパイル済みのバイナリを利用した。表 1 には各ベンチマークプログラムの実行命令数と値予測の対象となる命令の割合も示してある。つまり、分岐命令やストア命令などのデータ値予測の対象とならない命令を除いた割合である。命令は nop 命令を除いたリタイアした命令だけを数えている。

表 1 ベンチマークプログラム

プログラム	入力セット	実行命令数	(%) 対象
099.go	9 9	133M	77.69
124.m88ksim	dcrand.big	120M	68.52
126.gcc	genrecg.i	117M	65.96
129.compress	14000 e 2231	48M	66.25
130.li	queens 7	202M	58.76
132.jpeg	specmun.ppm	54M	79.22
134.perl	primes.in	10M	62.04
147.vortex	persons.250	101M	59.94

4. データ幅と予測精度の関係

本節では予測データのビット幅と予測精度の関係を調べる。シミュレーションには機能シミュレータを用い、各予測機構の値履歴表のエントリ数は無限であるとする。

4.1 有効ビット長と予測精度の関係

図 1~4 に、SPECint95 の全プログラムについて、各予測機構で予測されたデータの有効ビット長と予測精度の関係を示す。ここで有効ビット長は、データの値を決定する最上位ビットの位置で決定される。図 1~4 には、データ値予測精度が 50-60% (○), 60-70% (+), 70-80% (□), 80-90% (×), そして 90-100% (△) となる静的な命令群に対して、それらが生成するデータの平均有効ビット長を累積的に表してある。例えば最終値予測機構で予測精度が 90% 以上である命令は、その 61.3% がビット長が 8 以下であることがわかる。容易にわかるように、いずれの予測機構の場合でも予測精度が 90% 以上となる命令は、データ幅が狭いほうに偏りがある。全ての場合で、データ幅が 8 ビット以下となる命令で全体の 60% 以上を占めている。つまり予測容易な命令はデータ幅が狭く、データ幅の広い命令は予測困難であるということになる。

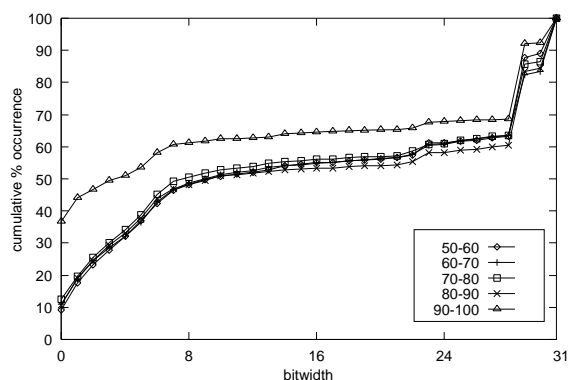


図 1 (%) ビット幅と予測精度の関係 (最終値)

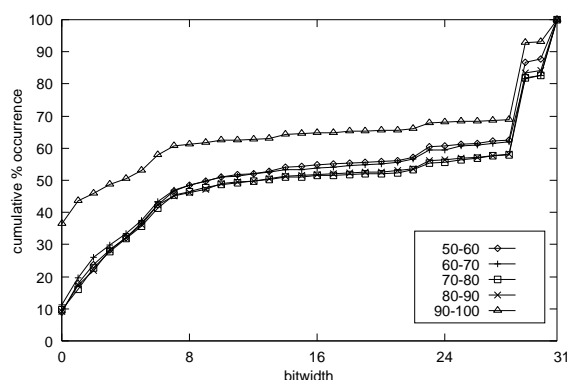


図 2 (%) ビット幅と予測精度の関係 (ストライド)

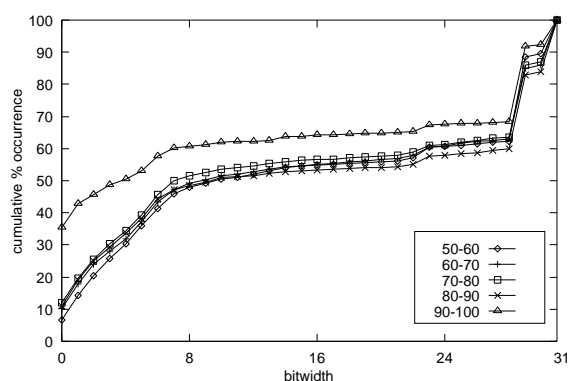


図 3 (%) ビット幅と予測精度の関係 (2 レベル)

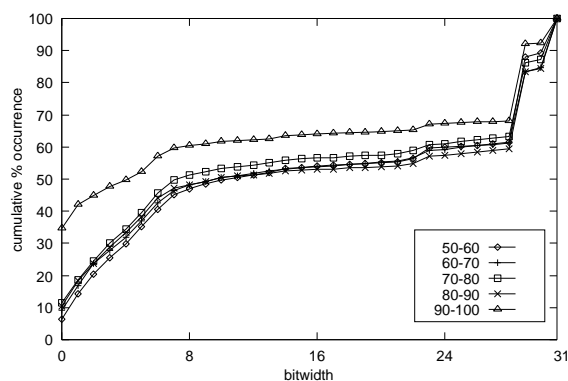


図 4 (%) ビット幅と予測精度の関係 (ハイブリッド)

図 5~8 に、各予測精度を示す命令が全体としてどの程度実行されているかを表す。データ値予測精度が 50-60%, 60-70%, 70-80%, 80-90%, そして 90-100% となる命令が実際にどのくらい実行されているかが表されている。例えば最終値予測機構では 099.go の場合、予測対象となる全実行命令の 60.5% が予測精度 90% 以上の命令群に属していることがわかる。総じてどの予測機構のどのプログラムにおいても、予測精度が 90% を越える命令が全実行命令の 50% 以上であることが確認できる。

以上の図 1~4 と図 5~8 に示される結果から以下のことがわかる。90% 以上の予測精度を持つ予測容易な命令はデータ幅が狭く、しかも全実行命令中に占める割合が

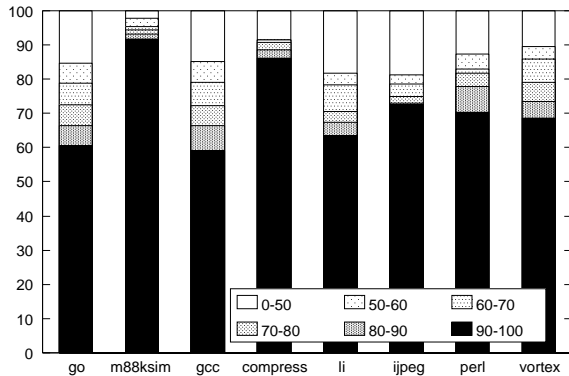


図 5 (%) 予測精度と実行命令数の関係 (最終値)

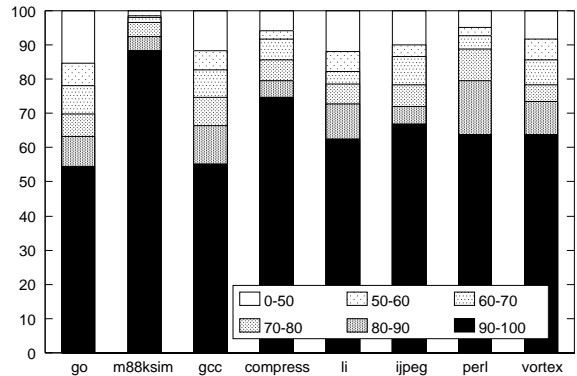


図 8 (%) 予測精度と実行命令数の関係 (ハイブリッド)

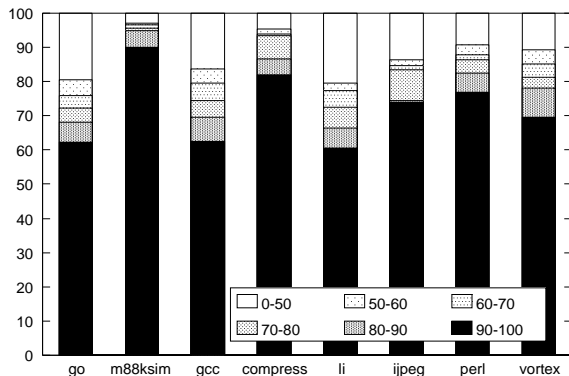


図 6 (%) 予測精度と実行命令数の関係 (スライド)

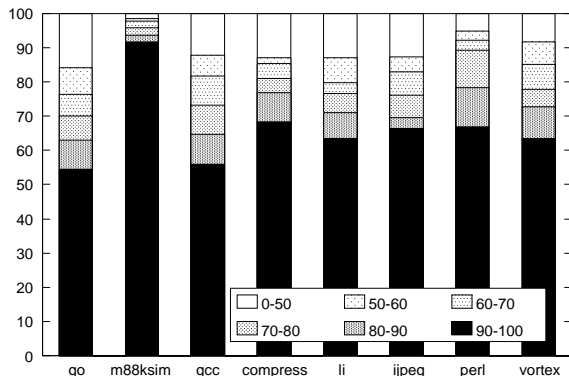


図 7 (%) 予測精度と実行命令数の関係 (2 レベル)

張する．図 9 ~ 12 に，データ幅が 32 ビット (オリジナル)，16 ビット，8 ビットの場合のデータ予測率を示す．値履歴表のエントリ数は無限大である．各グラフは 2 つの部分に分けられており，下部が正しく予測された命令の割合であり，上部が誤って予測された命令の割合である．残りの部分は，予測対象命令であるにもかかわらず予測信頼性などの評価から予測されなかった命令となる．予想とは異なり，予測データのビット幅を制限すると著しく予測率が低下していることが確認された．つまり，予測容易な命令が全予測対象命令の半数以上を占め，しかもそれらのデータ幅は 60% 以上が 8 ビット以下であっても，予測データのビット幅を制限することにより本来予

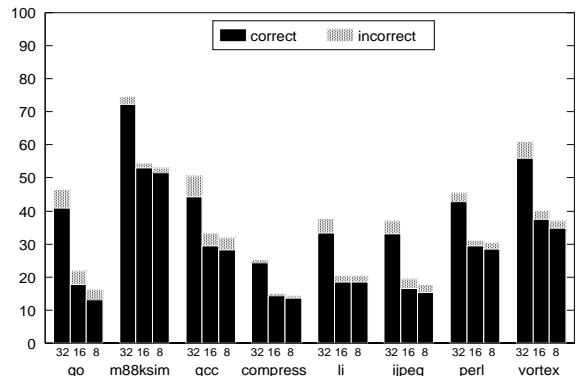


図 9 (%) データ幅と予測率の関係 (最終値)

大きい．つまり，予測されるデータのビット幅を制限しても，予測容易な命令は元々データ幅が狭いので深刻な影響が現われないと予想される．さらに，ビット幅を制限することで予測できなくなる命令は元々予測困難である上に，全体に占める割合は相対的に小さい．以上から，予測データのビット幅を制限してもデータ予測可能性は著しくは低下しないだろうと予想される．

4.2 データ幅を制限した場合の予測率

上記のシミュレーション結果をふまえて，予測データ値のビット幅を 8 ビットおよび 16 ビットに制限した場合を検討する．一般に整数データは 2 の補数表現をされるので，ビット幅を制限した場合には予測データを符号括

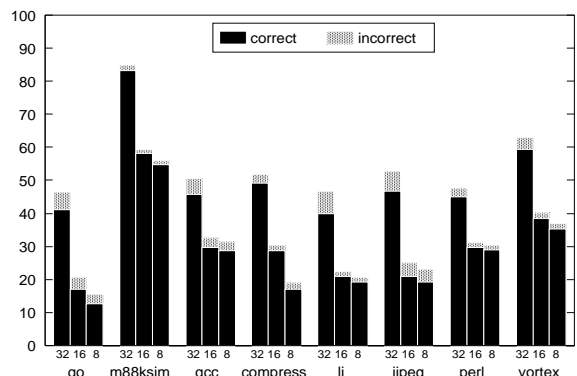


図 10 (%) データ幅と予測率の関係 (スライド)

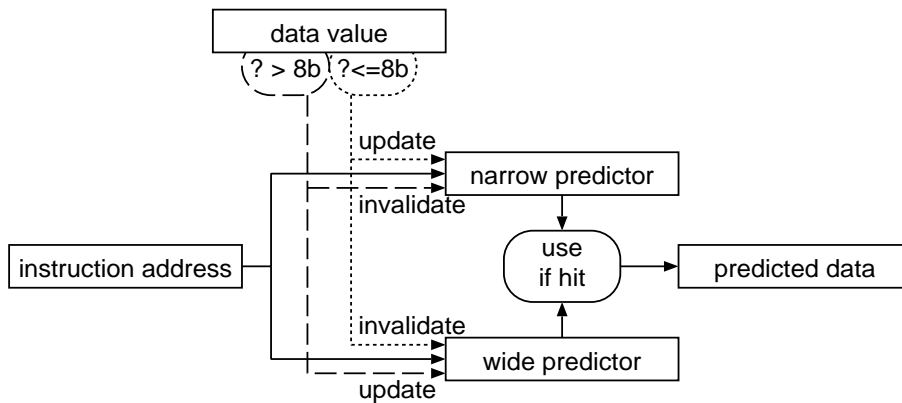


図 13 2モード値予測機構

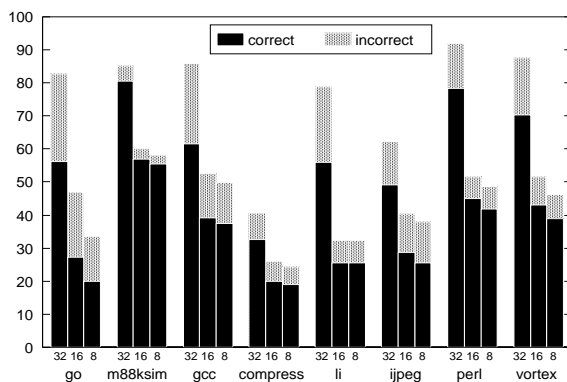


図 11 (%) データ幅と予測率の関係 (2 レベル)

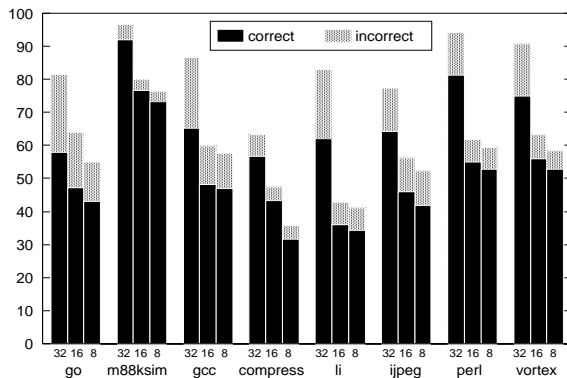


図 12 データ幅と予測率の関係 (ハイブリッド)

測可能であった命令の多くが予測できなくなってしまうことの影響は深刻である。一方でデータ幅の制限に関しては、16ビットに制限する場合と8ビットに制限する場合で大差無いことがわかる。

5. 2モード値予測機構

前節で、単純に予測データのビット幅を制限すると、予測率に深刻な悪影響を及ぼすことがわかった。本節では、ビット幅を制限したために予測不可能になったデータを再び予測可能にするために、2モード値予測機構を提案する。2モード値予測機構を提案する基本的な背景は、データ幅制限によって予測不可能になった命令は少

ないということである。

5.1 データ幅の変動

表 2 に SPECint95 の全てのプログラムについて、各命令が正しく予想できたデータのビット幅の変動を示す。すなわち各命令について、正しく予想できたデータの最大のビット幅と最小のビット幅の差を表している。容易にわかるようにビット幅の変動は非常に小さく、約 80% の命令では正しく予測できたデータのビット幅は全て等しい。さらに、全ての予測機構において、90%以上の命令に対するデータ幅の変動は4ビット以下である。このことから以下のことがわかる。ある特定の命令に注目すると、その命令が生成するデータのビット幅は大きく変動することは無いので、データ幅が制限された予測機構であっても、一旦正しく予測できれば将来に渡って予測可能であるということになる。見方を変えると、データ幅が制限されたために予測不可能になった命令は限定されるので、それらだけを選択的に予測する予測機構を備えれば、データ幅を制限する予測機構をサポートできるということにもなる。

表 2 (%) 最大・最小データ幅の差

予測機構	ビット長					
	0	1	2	3	4	>4
最終値	83.92	4.51	2.50	1.69	1.47	5.91
ストライド	83.40	4.20	2.77	1.81	1.55	6.27
2レベル	80.37	5.24	2.96	1.97	1.79	7.67
ハイブリッド	79.13	5.32	2.99	2.05	1.93	8.58

5.2 2モード値予測機構

前節の結果から、ビット幅の小さいデータと大きいデータとはそれぞれ異なるデータ値予測機構で予測できると予想される。この考察を元に、ビット幅の小さいデータを予測する値予測機構とビット幅の大きなデータを予測する値予測機構とを組み合わせた2モード値予測機構を提案する。

図 13 に2モード値予測機構の概念図を示す。2モード値予測機構は、予測可能なデータのビット幅が異なる二つの予測機構から構成される。二つの予測機構で用いられる予測方式は同じであるとする。これ以降、予測データの幅の大きい予測機構を広データ幅予測機構、予測データの幅の小さい予測機構を狭データ幅予測機構と呼ぶこと

にする。予測データ幅の異なる二つの予測機構は値履歴表の更新時に使い分けられる。予測対象である命令の演算結果が閾値のビット幅よりも大きい時には広データ幅予測機構に、そうでない時には狭データ幅予測機構に登録される。同時に、選択されなかった方の予測機構ではその命令に相当するエントリが無効化される。つまり、ある特定の命令に関する履歴は、二つの予測機構のうちどちらか一方にしか保存されない。値を予測する時には二つの機構が並行にアクセスされる。上記の説明からわかるように、予測データは高々どちらか一方の予測機構からしか供給されない。したがって予測データを選択するための特別な機構は必要無い。

このように予測データのビット幅に最適な予測機構を動的に選択することで、前節で評価した予測機構では予測不可能なビット幅の大きなデータを効率良く予測可能になる。大部分のデータは狭データ幅予測機構で予測可能なので広データ幅予測機構の容量を小さくでき、2モード化による深刻なハードウェア量の増大は避けられている。

二つの予測機構を異なる予測手法に基づいて構成することも可能であるが、本稿では評価の第1段階として二つの予測機構が同一の予測手法に基づく2モード予測機構を検討する。また、三つ以上の予測データ幅の異なる予測機構を組み合わせることも可能であるが、やはり本稿では評価の対象外とする。これらの評価は将来の課題である。

5.3 評価

つづいて2モード値予測機構の評価を行なう。まず予測率を評価し、つづいてプロセッサ性能への影響を調べる。

予測率の評価では、狭データ幅予測機構の値履歴表エントリ数を無限とし、広データ幅予測機構の値履歴表のエントリ数を制限する。今回はダイレクトマップで構成された512エントリと1024エントリの広データ幅予測機構を評価する。表3に前節までに評価した値履歴表で利用されたエントリ数を示す。この表から512エントリおよび1024エントリは、全体に対してそれぞれ0.7~24.5%および1.4~49.1%を占めることがわかる。ここで、広データ幅予測機構はダイレクトマップで構成されるため、命令の衝突が生じることによって全てのエントリが効率良く利用されるわけではないことに注意していただきたい。一方狭データ幅予測機構は実質的にはフルアソシアティブに構成されるために、全てのエントリが効率良く利用される。

図14~17に1024エントリ(B1024)と512エントリ(B512)の値履歴表を持つ広データ幅予測機構を用いた2モード値予測機構の予測率を表す。前節で8ビットの狭

表3 利用された値履歴表のエントリ数

プログラム	エントリ数
099.go	40,496
124.m88ksim	7,064
126.gcc	71,032
129.compress	2,087
130.li	3,768
132.jpeg	13,240
134.perl	7,498
147.vortex	35,152

データ幅予測機構と16ビットの狭データ幅予測機構で予測率に大差無いことがわかっている。以下の評価ではハードウェア規模の小さな8ビットの狭データ幅予測機構を用いる。図には予測データ幅が32ビット(32b)と8ビット(8b)である従来の予測機構の結果も合わせて示してある。各グラフは2つの部分から構成されているが、図9~12の場合と同様である。全ての場合で予測率の改善が観察できる。このことから、狭データ幅予測機構でデータ幅の狭い命令をフィルタリングし、残った命令だけを広データ幅予測機構で予測することが可能であることが確認できた。

表4に、一旦一方のデータ幅予測機構に登録された命

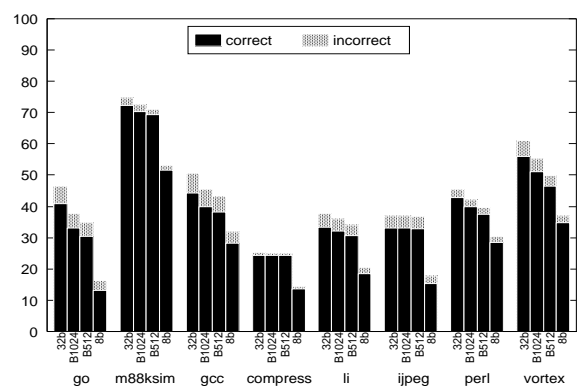


図14 (%)2モード値予測機構の予測率(最終値)

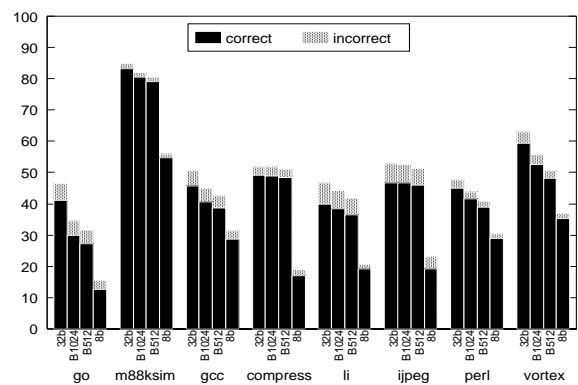


図15 (%)2モード値予測機構の予測率(ストライド)

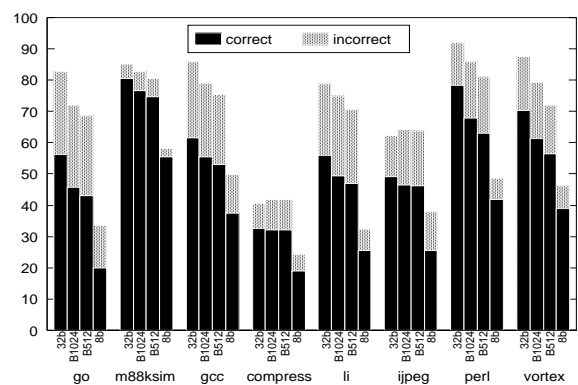


図16 (%)2モード値予測機構の予測率(2レベル)

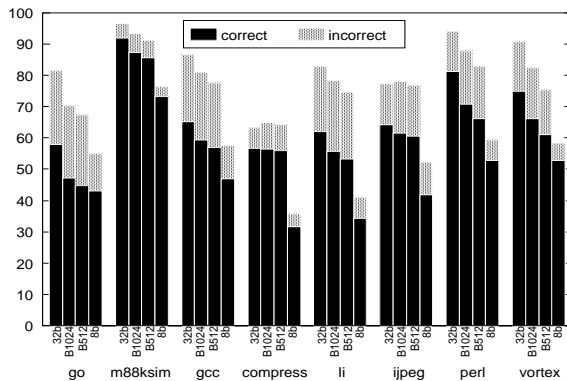


図 17 (%)2 モード値予測機構の予測率 (ハイブリッド)

令が無効化され他方に再登録される頻度を示す。機能シミュレータでの評価なので、各予測機構での結果が同じになることに注意されたい。予想したように再登録の頻度は非常に小さく、このことからフィルタリングが可能であることが確認された。

表 4 (%) 再登録の頻度

プログラム	(%) 頻度	
	512	1024
099.go	1.74	1.81
124.m88ksim	1.16	1.22
126.gcc	3.20	3.36
129.compress	2.29	2.29
130.li	4.44	4.50
132.jpeg	1.83	1.87
134.perl	2.81	3.01
147.vortex	3.40	3.85

図 14 ~ 17 に戻ると、以下のことも観測される。2 レベル予測機構とハイブリッド予測機構では、最終値予測機構とストライド値予測機構と比較して、2 モード化により予測の失敗率が增大する場合が見られる。これは、これら二つの予測機構が履歴に大いに依存しているためである。一旦一方のデータ幅予測機構に登録された命令が無効化され他方に再登録されると、これまでの履歴が失われてしまう。このことは予測精度の低下を招くが、履歴に大きく依存しているこれら二つの予測機構で著しい影響が現われる。表 4 で見た小さな頻度の再登録であっても、これらの予測機構では予測失敗率の低下を引き起こしている。

つづいてプロセッサ性能への影響を評価する。現実的なプロセッサで評価するために、前節までとは異なり値履歴表のエントリ数は有限とする。すなわち、各予測機構はそれぞれダイレクトマップで構成され、値履歴表のエントリ数は狭データ幅予測機構が 4096、広データ幅予測機構が 512 とする。オリジナルの予測機構ではエントリ数は 4096 である。ここで 4096 というエントリ数は過去の文献で頻りに用いられている値である。また、2 レベル値予測機構とハイブリッド値予測機構のパターン履歴表は、広データ幅予測機構と狭データ幅予測機構とで共有する。以上のハードウェア構成の時の 2 モード化によるハードウェア量削減の効果は以下のとおりである。い

ずれの予測方式でも、データレイに要するハードウェア量が 62.5%削減されている。タグレイなどを考慮したハードウェア全体で評価すると、最終値、ストライド、2 レベル、ハイブリッド値予測機構で、それぞれ 31.3%、43.5%、42.7%、45.1%の削減を達成できている。

図 18 に 2 モード予測機構を用いた時のプロセッサ性能の向上を示す。評価には 1 サイクル当たりの完了命令数 (committed instructions per cycle: IPC) を用いた。カウントされる命令には nop 命令を含んでいない。プロセッサの性能向上は、増加した IPC をデータ予測を行わない基本モデルの IPC で割った増加率で示す。図 18 には 2 モード化されないオリジナルの予測機構の場合も合わせて示してある。各プログラムに対して左から順に、最終値、ストライド、2 レベル、ハイブリッド、2 モード型最終値、2 モード型ストライド、2 モード型 2 レベル、そして 2 モード型ハイブリッド値予測機構を用いたときの結果である。図 18 より、130.li の 2 レベル値予測機構とハイブリッド値予測機構の場合を除いて、2 モード化によりハードウェア量を削減しても、プロセッサ性能向上率に深刻な影響は現われないことがわかる。このことから、2 モード値予測機構の有効性が確認できる。2 モード値予測機構は、データ投機実行による効果を維持しつつ、従来のデータ値予測機構と比較して大幅にハードウェア量を削減できる。

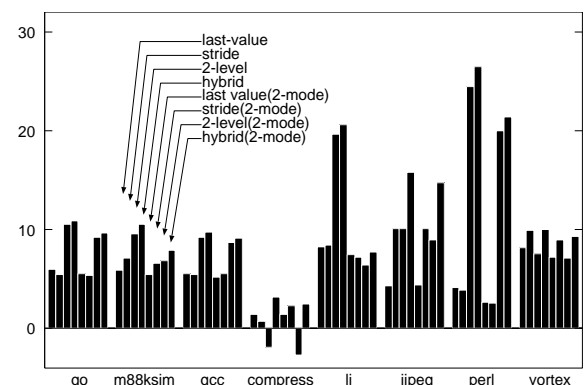


図 18 (%) プロセッサ性能向上率

表 5 に、タイミングシミュレーション時の値履歴表の再登録頻度を示す。タイミングシミュレーションであるので、各予測機構によって結果が異なることに注意されたい。表 4 では 130.li が顕著な再登録頻度を示していたが、表 5 では 147.vortex と同程度の頻度である。147.vortex では性能向上率の低下はほとんど観察されず、この再登録頻度からだけでは 130.li での性能向上率の著しい低下は説明できない。この原因に関しては現在調査中である。

6. おわりに

本稿では、大規模なハードウェア量が問題となっているデータ値予測機構を現実的なものとするために、そのハードウェア量を削減することを検討した。演算結果の有効なデータ幅の特徴を利用して予測データのビット幅

表 5 (%) 再登録の頻度

プログラム	(%) 頻度			
	last	stride	2-level	hybrid
099.go	0.76	0.75	0.81	0.81
124.m88ksim	1.01	1.01	1.03	1.04
126.gcc	1.93	1.93	2.03	2.04
129.compress	1.58	1.58	1.51	1.60
130.li	3.07	3.06	3.11	3.14
132.jpeg	1.40	1.46	1.42	1.57
134.perl	1.96	1.94	2.30	2.32
147.vortex	2.94	2.97	2.97	3.00

を制限することで目標を達成する。単純に予測機構に保持されるデータ幅を削減すると予測精度は著しく低下することが確認された。この問題を解決するために2モード値予測機構を提案し、予測精度の低下を抑えることができることを確認した。2モード値予測機構はハードウェア量を約40%削減することに成功している。2モード予測機構を用いたプロセッサ性能の評価では、ほとんどのベンチマークプログラムでわずかに低下した予測精度の影響はほとんど無いことが確認できている。

将来の課題のひとつは、MediaBench⁶⁾などのマルチメディア系アプリケーションを用いて評価することである。現在、MediaBenchを用いた評価を行う準備を開始している。データ幅を制限する効果はマルチメディアアプリケーションで有効であると期待している。さらにマルチメディア応用の観点からは、MMX命令¹⁰⁾などのマルチメディア向けSIMD命令への展開も検討している。また今回の評価では、最終値予測機構、ストライド値予測機構、2レベル値予測機構、そしてストライド値予測と2レベル値予測のハイブリッド予測機構にのみ、2モード値予測方式を適用した。今後は、より予測精度の向上しているコンテキストベース値予測機構^{9),12)}やマルチストライド値予測機構¹⁵⁾においても本方式の有効性を確認する必要があると思われる。また今回はデータレイ部のハードウェア量のみに着目したが、分岐予測機構で検討されているFaginの方式⁴⁾によるタグレイ部のハードウェア量削減¹⁷⁾との併用も検討する必要がある。

本稿は九州工業大学で現在進行中であるハイパフォーマンスマイクロプロセッサCOSMOSの検討における一つの成果である。<http://www.mickey.ai.kyutech.ac.jp/~tsato/cosmos/>からは追加情報が入手可能である。

参 考 文 献

- Brooks,D., Martonosi,M.: Dynamically exploiting narrow width operands to improve processor power and performance, *5th International Symposium on High Performance Computer Architecture* (1999).
- Burger,D., Austin,T.M.: The SimpleScalar tool set, version 2.0, *ACM SIGARCH Computer Architecture News*, vol.25, no.3 (1997).
- Driesen,K., Holzle,U.: The cascaded predictor: economic and adaptive branch target prediction, *31st International Symposium on Microarchitecture* (1998).
- Fagin,B.: Partial resolution in branch target buffers, *IEEE Transactions on Computers*, vol.46, no.10 (1997).
- Gabbay,F.: Speculative execution based on value prediction, *Technical Report #1080*, Department of Electrical Engineering, Technion (1996).
- Lee,C., Potkonjak,M., Mangione-Smith,W.H.: MediaBench: a tool for evaluating and synthesizing multimedia and communications systems, *30th International Symposium on Microarchitecture* (1997).
- Lipasti,M.H., Wilkerson,C.B., Shen,J.P.: Value locality and load value prediction, *International Conference on Architectural Support for Programming Languages and Operation Systems VII* (1996).
- Morancho,E., Llaberia,J.M., Olive,A.: Split last-address predictor, *International Conference on Parallel Architectures and Compilation Techniques* (1998).
- Nakra,T., Gupta,R., Sofa,M.L.: Global context-based value prediction, *5th International Conference on High Performance Computer Architecture* (1999).
- Peleg,A., Wilkie,S., Weiser,U.: Intel MMX for multimedia PCs, *Communications of the ACM*, vol.40, no.1 (1997).
- Rychlik,B., Faistl,J.W., Krug, B.P., Kurland, A.Y., Sung,J.J., Velev,M.N., Shen, J.P.: Efficient and accurate value prediction using dynamic classification, *Technical Report CMuART-98-01*, Department of Electrical Computer Engineering, Carnegie Mellon University (1998).
- Sazeides,Y., Smith,J.E.: Implementations of context based value predictors, *Technical Report TR-ECE-97-8*, Department of Electrical Computer Engineering, University of Wisconsin-Madison (1997).
- Wang,H., Sun,T., Yang,Q.: CAT - caching address tags: a technique for reducing area cost of on-chip caches, *22nd International Symposium on Computer Architecture* (1995).
- Wang,K., Franklin,M.: Highly accurate data value prediction using hybrid predictors, *30th International Symposium on Microarchitecture* (1997).
- 吉瀬謙二, 坂井修一, 田中英彦: マルチレベル・ストライド値予測機構による命令レベル並列性の向上, 並列処理シンポジウム *JSP'99* (1999).
- 佐藤寿倫: アドレス名前替えによるロード命令の投機的実行, 並列処理シンポジウム *JSP'98* (1998).
- 佐藤寿倫, 有田五次郎: タグビット幅を考慮したデータ値予測機構のハードウェア量削減, 信学技報 *CPSY2000* (2000).
- 山田祐司, 小林良太郎, 安藤秀樹, 島田俊夫: 2レベル表構成の導入による分岐先バッファの容量削減, 並列処理シンポジウム *JSP'99* (1999).