論　文

キャッシュ/バッファ内フラグ方式による命令先取り分配方式

吉田　昌司[†]　田中　成弥[†]　松尾康太郎[††]　堀田多加志[†]
澤本　英雄[†††]　清水　照久[††††]

Instruction Fetch and Dispatch Scheme with Flag in Cache/in IBR

Shoji YOSHIDA[†], Shigeya TANAKA[†], Kotaro MATSUO[††], Takashi HOTTA[†], Hideo SAWAMOTO[†††], and Teruhisa SHIMIZU[††††]

あらまし　(1)命令キャッシュ内に備えた逐次処理フラグと命令バッファ内並列処理可否判定を併用するキャッシュ/バッファ内フラグ方式と，(2)分岐命令の無効化を分岐履歴テーブルにより予測する動的無効化予測方式，の二つからなる命令先取り分配方式を提案する．本方式により，1サイクルに複数命令を並列処理するスーパスカラプロセッサにおいて動作周波数を落とさずに複数演算器に有効に命令を供給することができる．0.3 μm CMOS を用いた 4.5 M トランジスタの 2 命令並列処理スーパスカラプロセッサにおいて本方式を適用し動作周波数 150 MHz を達成し，その有効性を実証した．

キーワード　RISC プロセッサ，スーパスカラ，命令フェッチ，命令バッファ，分岐予測，無効化

1. まえがき

プロセッサの性能は，LSI の微細化に伴い著しい高性能化を遂げている[1],[2]．命令機能を簡略化し1命令を1サイクルで実行する RISC (Reduced Instruction Set Computer) が提案され実用化された[3]．更なる高性能化のため，1サイクルに複数命令を実行するスーパスカラプロセッサが出現した[4]．最近ではその命令処理方法として，プログラム順と異なった順序で命令を実行する乱発行乱終了 (Out-of-Order) 方式が開発された．

プロセッサの性能は動作周波数に比例し，1命令処理するのに要する平均サイクル数 (CPI: Clock cycles Per Instruction) に反比例する．CPI は基本 CPI と ΔCPI の和で表せる．基本 CPI は，パイプラインの乱れがない場合の CPI で，スカラプロセッサの場合 1，多重度 m のスーパスカラプロセッサの場合 $1/m$ となる．ΔCPI は，パイプラインの乱れがある場合，CPI が基本 CPI より増加する増加分である．パイプラインの乱れの原因には，二つの命令は同時に一つしかない資源を使えないという資源競合，後続命令が前命令の結果データを使うとき後続命令を実行できないデータ依存，命令の流れが変化するときに後続命令を実行できない制御依存の三つがある．プロセッサの性能を向上させるには，動作周波数を向上させ，また資源競合，データ依存，制御依存で生ずる ΔCPI を抑えることが必要である．

スーパスカラプロセッサでは，次に実行すべき命令アドレスを予測しその命令アドレスに従い命令キャッシュから複数の命令を同時に読み出す命令先行フェッチ予測と，同時実行するための資源競合/データ依存を検出して複数の演算器に命令を同時に分配する並列処理可否判定からなる命令先取り分配方式により ΔCPI が大きく左右される．プロセッサの性能向上のためには，いかに有効に命令を演算器に供給するかの命令先取り分配方式が重要である[5]．スーパスカラプロセッサ特有の並列処理可否判定はプロセッサの動作周波数を決めるクリティカルパスになりスカラプロセッサと比較して動作周波数が低くなる．また，並列処

† (株)日立製作所日立研究所，日立市
Hitachi Research Laboratory, Hitachi Ltd., 7-1-1 Omika, Hitachi-shi, 319-12 Japan
†† (株)日立製作所オフィスシステム事業部，海老名市
Office System Division, Hitachi Ltd., 810 Shimoimaizumi, Ebina-shi, 243-04 Japan
††† (株)日立製作所汎用コンピュータ事業部，泰野市
General Purpose Computer Division, Hitachi Ltd., 1 Horiyamashita, Hadano-shi, 259-13 Japan
†††† (株)日立製作所デバイス開発センタ，青梅市
Device Development Center, Hitachi Ltd., 2326 Imai, Ohme-shi, 198 Japan

理可否判定回路のための追加ステージを備えた場合は動作周波数を高められるが，制御依存による⊿CPIが増加する．本論文では，動作周波数を保つべく，プログラム順で命令を実行する順発行順終了（In-Order）方式とし，そのCPIを改善する命令先取り分配方式について述べる．命令先取り分配方式の課題は，(1)スーパスカラプロセッサ特有の回路である並列実行可否判定に要する時間を縮めて動作周波数の低下を抑え，資源競合/データ依存/制御依存で生ずる⊿CPIの増加を抑えること，(2)命令先行フェッチの確からしさを向上させる命令先行フェッチ予測方式により，データ依存/制御依存で生ずる⊿CPIの増加を抑えることである．

これらの課題を満足する解決案として，2.では，命令キャッシュ内に備えた逐次処理フラグと命令バッファ内並列処理可否判定を併用するキャッシュ/バッファ内フラグ方式，3.では動的無効化予測方式について述べる．更に，4.で0.3 μm CMOSプロセス技術を用いて開発したスーパスカラプロセッサにこれらの方式を適用した結果を評価する．

2. 並列処理可否判定方式

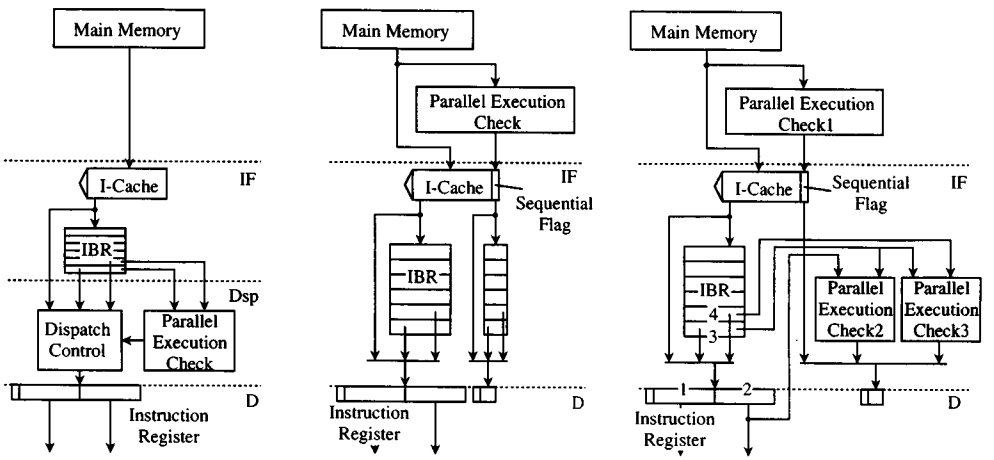
従来の2命令並列処理可否判定方式には，並列処理可否判定のためにステージを追加する追加ステージ方式[6],[7]と，命令キャッシュに並列処理可否判定の結果である逐次処理フラグを登録するキャッシュ内フラグ方式[8]がある．本章では，まずこれらについて説明した後，本論文で提案する命令バッファ内並列処理可否判定と逐次処理フラグを併用するキャッシュ/バッファ内フラグ方式について説明する．

2.1 追加ステージ方式

図1(a)を用いて追加ステージ方式を説明する．本方式は，命令アドレスによって指される命令を命令キャッシュ（I-Cache）から読み出し，命令バッファ（IBR）へ格納する命令フェッチステージ（IF）と，命令バッファから発行しようとする2命令間のデータ依存/資源競合を検出して，並列処理できるか否かを判定するための追加ステージ（Dsp）からなる．本方式の利点は，命令先取り分配機能を2サイクル（IF+Dsp）で実行し，スーパスカラ特有の並列処理可否判定（Parallel Execution Check）を追加ステージで処理するため，この部分が動作周波数を決めるクリティカルパスとはならない．しかしながら，命令先取り分配機能に最短でも2サイクル（IF+D）のレイテンシが必要であり，制御依存が発生して命令フェッチから再度実行する場合，1サイクル分余分にペナルティが必要で⊿CPIが増加する．

2.2 キャッシュ内フラグ方式

次に図1(b)を用いてキャッシュ内フラグ方式を説明する．本方式は，命令キャッシュがミスヒットして主メモリ（Main Memory）から命令列をブロック転送する際に，ブロック転送単位内の隣り合う連続命令を対象に並列実行可否判定を行い，その結果を逐次処理フラグ（Sequential Flag）として命令キャッシュ



(a) Additional stage　(b) Flag in cache　(c) Flag in cache/in IBR

図1 並列処理可否判定方式
Fig. 1 Parallel execution check scheme.

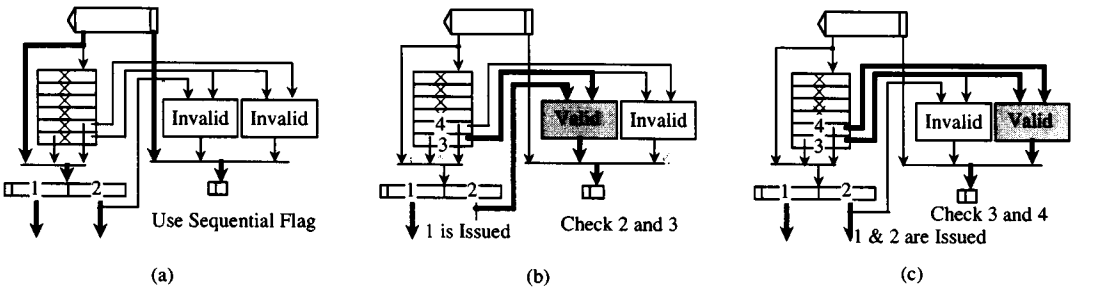


図 2　キャッシュ/バッファ内フラグ方式の動作
Fig. 2　Operation of flag in cache/in IBR. (a) IBR is empty and instructions are fetched from I-Cache directly (b) IBR isn't empty and instruction (1) is issued (c) IBR isn't empty and instruction (1) and instruction (2) are issued.

に命令と共に格納するものである．この方式の利点は，命令キャッシュから命令と共に読み出される逐次処理フラグをもとに並列実行可否判定を行うため，命令アドレスによって指される命令を命令キャッシュから読み出し，命令バッファへ格納し，並列実行可否判定するまでを1サイクルで実行できる．しかしながら，ブロック転送単位をまたがった命令間では逐次処理フラグでの判定ができず，常に逐次処理とせざるを得ない欠点がある．つまり，ブロック転送単位内の連続アドレスのときのみ並列実行可能で，それ以外のケースでは並列処理できない．この並列処理制約のため性能向上に限界がある．

2.3　キャッシュ/バッファ内フラグ方式

追加ステージによるレイテンシ増加がなくかつ並列処理制約を解消することができる，キャッシュ/バッファ内フラグ方式を提案する．本方式の基本的な概念は，命令バッファが空のときは命令キャッシュからの逐次処理フラグをもとに並列処理し，命令バッファに命令が発行待機中のときは待機サイクル中に命令間の並列処理可否判定を行い並列処理する点にある．この方式によれば，動作周波数を低下させることなく，追加ステージなしに，かつ並列処理制約なしに並列処理可否判定が可能となる．

図1(c)に本方式を示す．命令キャッシュには逐次処理フラグを用意し，命令キャッシュから読まれた命令は，命令バッファに命令実行順に格納される．本方式では，スーパスカラプロセッサ特有の回路である並列処理可否判定を三つもつ．並列処理可否判定1は命令キャッシュ内の逐次処理フラグを生成するためのものである．残りの二つは命令バッファの中にもち，並列処理可否判定2は命令レジスタ内の2番目の命令と命令バッファに保持される3番目の命令間の並列処理可否を調べるものであり，並列処理可否判定3は命令バッファに保持される3番目と4番目の命令間の並列処理可否を調べるものである．命令レジスタから演算器への命令発行数と命令バッファの状態によって，三つの並列処理可否判定を選択して使い分ける．これによって追加ステージなしに並列処理することが可能となる．

図2は，本方式の具体的な動作を示す図である．図2(a)は，命令バッファが空で，命令キャッシュから直接フェッチする場合，図2(b)は，命令バッファが空でなく発行命令が一つの場合，図2(c)は，命令バッファが空でなく発行命令が二つの場合である．図2(a)の場合は，命令キャッシュから読み出された命令は命令バッファを経由せずに命令レジスタに直接格納されるので，命令バッファによる並列処理可否判定は用いることができない．この場合にのみ命令キャッシュから読み出された逐次処理フラグを選択する．これによって命令先取り分配機能は1サイクルで実行可能となる．図2(b)の場合は，命令バッファに命令が保持された状態で，命令レジスタ内の1番目の命令のみ発行されるので，次に発行されるべき命令組である，命令レジスタ内の残りの2番目の命令と命令バッファ内の3番目の命令との間の並列処理可否判定を選択する．図2(c)の場合は，命令バッファに命令が保持された状態で，命令レジスタ内の2命令が発行されるので，次に発行されるべき命令組である，命令バッファ内の3番目の命令と4番目の命令の間の並列処理可否判定を選択する．

命令バッファの二つの並列処理可否判定は，命令発行直前の命令間に対して，命令バッファに保持されている間に並列に行い，命令の発行条件によって切り換えて使用する．これにより，異なるブロック転送単位間でも命令バッファの中で隣り合って配置されたとき並列処理可否判定できるので，並列処理制約を除くことができる．

3. 命令先行フェッチ予測方式

本章では，適用したプロセッサの命令アーキテクチャについて述べた後，分岐予測方式，静的無効化方式について述べる．更に命令先行フェッチの予測確率を向上させる動的無効化予測方式を提案し，その実現手段として分岐/無効化混合型動的予測方式を提案する．

3.1 命令アーキテクチャ

我々が採用した命令アーキテクチャには，命令の流れを変化させる機能として条件分岐命令と無効化（Nullify）機能がある．条件分岐命令は命令の中で指定した条件が成立したとき命令アドレスを変化させる命令である．無効化は，命令の中で指定した条件が成立したとき次に続く命令を無効化する機能で，無効化された命令は，NOP命令と全く同様にマシンの状態を全く変えることなく終了されなければならない．つまり，無効化は次の1命令をスキップでき，命令アドレスを変えるためのアドレス計算などを省いた短い分岐とみなすことができる．命令の流れを変えるために分岐命令をおくのに比べると，無効化は無効化条件を通常の演算命令内で指定できるので特別な命令コードの消費がないという利点がある．また分岐と無効化を組み合わせて複雑な条件分岐として用いることが多い[9]．しかし無効化処理は，3.3で詳しく述べるように，データ依存/制御依存により⊿CPIが増加する原因となる．

3.2 分岐予測方式

分岐予測方式には，前もって一義的に予測を決定する静的分岐予測と，実行時に予測を決定する動的分岐予測がある[10],[11]．

静的分岐予測の代表的な例として，BTFN（Backward Taken Forward Not-taken）[10]がある．これは，分岐命令の命令アドレスより小さい命令アドレスへ分岐する場合分岐成立予測，分岐命令の命令アドレスより大きい命令アドレスへ分岐する場合分岐不成立予測するものである．

動的分岐予測の代表的な例として，命令フェッチとそのアドレスに存在する分岐命令の成立不成立を連想づける分岐履歴テーブル（BHT: Branch History Table）があり，予測確率を上げるために過去の分岐成立不成立の履歴のもち方などが研究されている[11]．履歴のもち方の代表的な例としては，2ビットカウンタ方式がある．これは，分岐成立時2ビットカウンタをカウントアップ，分岐不成立時2ビットカウンタをカウントダウンし，過去において2回以上分岐不成立が連続していない分岐に対し分岐予測するものである．

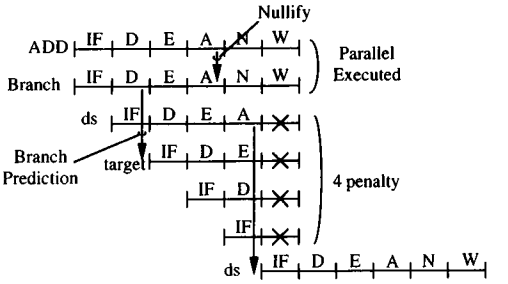


図3 無効化された分岐のパイプライン
Fig. 3 Pipeline chart of nullified branch.

3.3 静的無効化予測方式

静的無効化予測方式は，常に無効化しないという予測でパイプラインを実行させる．静的無効化予測が外れた場合，通常はその命令のみ無効化すればよいが，データ依存/制御依存が発生していると，その影響の及んでいる命令すべてを命令フェッチから再実行する必要があり，⊿CPIが増加する．図3に，無効化された分岐命令のパイプラインを示す．加算命令（ADD）の無効化条件成立により，次命令の分岐成立予測した分岐命令が無効化されると，制御依存が発生し，分岐の直後の命令である遅延スロット命令から再実行する．このとき，図のように4サイクルのペナルティが発生する．また，無効化された命令がレジスタに書込み後続命令が同一レジスタを用いているとき，無効化された命令の結果を後続命令へレジスタショートパスしてしまい誤演算となるため，データ依存が発生し，後続命令の命令フェッチから再実行する．このとき，やはり同様に4サイクルのペナルティが発生する．更に，無効化された命令によるデータキャッシュアクセスにおいても，同様のペナルティが発生する．

3.4 動的無効化予測方式

上述の⊿CPI増加を抑えるためには，無効化処理を動的に予測してパイプラインを乱れさせない機構が必

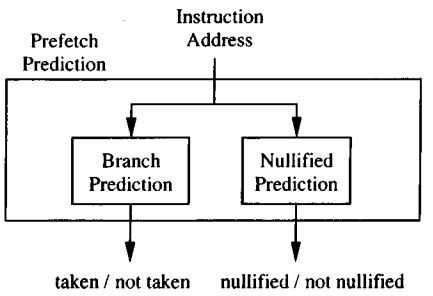


図 4 動的無効化予測方式の概念
Fig. 4 Concept of dynamic nullified prediction.

要となる．そこで，動的無効化予測方式を提案する．動的無効化予測は，すべての命令に対し，過去の無効化されたか否かについての結果の履歴をとっておき，それに基づいて無効化の動的予測を行う方式であり，無効化履歴テーブルを備え，無効化されたときに登録する．命令アドレスをもとに無効化と予測されると，無効化命令との間のデータ依存によるレジスタショートパスの抑止，分岐命令の無効化予測時分岐不成立処理，データキャッシュアクセス抑止の動作を行う．動的無効化予測をすることで静的無効化予測より予測成功率を高めれば，ΔCPI を低減できる．

動的無効化予測方式の概念を図 4 に示す．動的無効化予測は，動的分岐予測とは独立に無効化するか否かについて予測を決定し，命令を実行する．動的無効化予測と動的分岐予測との違いは，動的分岐予測は分岐命令が対象であるのに対して，動的無効化予測はすべての命令が対象であること，また，動的分岐予測が分岐命令の分岐先命令アドレスを制御するのに対して，動的無効化予測は無効化命令のデータ依存/制御依存といったパイプライン動作を制御する点である．

3.5 分岐/無効化混合型動的予測方式

前節で提案した動的無効化予測方式の概念の実現手段として，動的無効化予測の中で最もペナルティが大きい分岐命令の無効化に着目した分岐/無効化混合型動的予測方式を提案する．

動的無効化予測を分岐命令だけに限定した場合，分岐予測と分岐命令に対する動的無効化予測（無効化分岐予測）を組み合わせた分岐/無効化混合型動的予測方式の動作は表 1 のようになる．動的無効化予測を分岐命令に限ると，動的無効化予測の出力は動的分岐予測と同様に分岐の成立/不成立になるため，従来の動的分岐予測と同じ分岐履歴テーブルを流用することが可能となり，ハードウェアを小型化することが可能である．

表 1 分岐/無効化混合型動的予測の動作
Table 1 Operation of branch/nullified merged dynamic prediction.

分岐／無効化混合型動的予測		予測動作
分岐予測	無効化予測	
分岐成立と予測	無効化不成立と予測	分岐成立
	無効化成立と予測	分岐不成立
分岐不成立と予測	無効化不成立と予測	
	無効化成立と予測	

4. プロセッサの開発と方式評価

4.1 スーパスカラプロセッサの概要

開発したスーパスカラプロセッサチップの諸元を表 2 に示す．

プロセッサの全体構成ブロック図を図 5 に示す．32 ビット整数演算系は，二つの整数演算器と 4 リード 3 ライトのレジスタ 32 本で構成する．また，64 ビット浮動小数点演算系は，加減算器，乗除算器と 4 リード 4 ライトのレジスタ 128 本で構成する [12], [13]．キャッシュメモリは 2 階層構成をとる．1 次キャッシュはダイレクトマップ方式のそれぞれ 16 K バイト命令/データキャッシュを内蔵し，512～8 M バイト 2 次キャッシュは外部 SRAM を 16 バイト専用端子で接続する．更にメモリ管理，高速アドレス変換のための 256 エントリの命令/データ TLB を内蔵する．更に，本論文で提案した 1024 エントリの分岐/無効化混合型動的予測分岐履歴テーブルと 8 エントリの命令バッファをもつ．

4.2 キャッシュ/バッファ内フラグ方式の評価

4.2.1 ハード量

内蔵 1 次命令キャッシュはプログラムカウンタで指された 16 バイト境界の 4 命令を一度に読み出し命令バッファに登録する．一度に発行する最大命令数 2（8 バイト）に対して命令キャッシュから 16 バイト読み出すように設計することにより，命令バッファに命令を保持する確率を上げ命令バッファの中で保持中に検出する並列処理可否判定回路の活性化率を高めた．更に命令先行フェッチ予測方式と組み合わせると，異なるブロック転送単位間の命令が命令バッファに格納される可能性は高くなり，異なるブロック転送単位間での並列処理の効果が増す．

命令キャッシュがミスヒットした場合，2 次キャッ

表 2 チップの諸元
Table 2 Chip characteristics.

動作周波数	150MHz(ﾌﾟﾛｾｽworst, 環境worst)
プロセス	0.3μm 4層 CMOS
トランジスタ数	4.5M
チップサイズ	15.7mm × 15.7mm
電力	13W (150MHz時)
電源電圧	2.5V

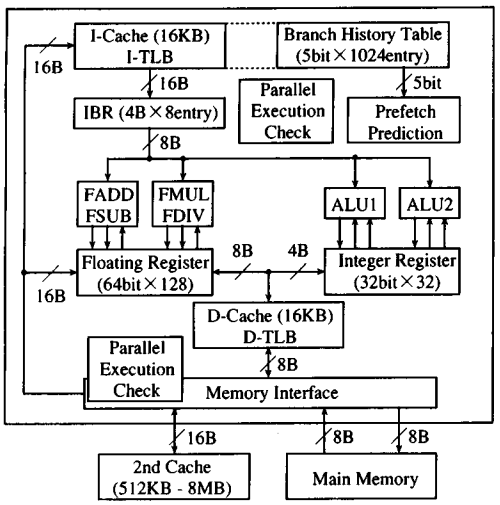


図 5 プロセッサのブロック図
Fig. 5 Block diagram of the processor.

シュから32バイトのブロック転送を行う．更に2次キャッシュがミスヒットした場合，主メモリから2次キャッシュにライン転送を行う．キャッシュミスヒット時逐次処理フラグを生成するが，その判定時間は1マシンサイクルかかる．本プロセッサでは2次キャッシュミスヒットによるライン転送時に逐次処理フラグを生成し，2次キャッシュにも命令と共に逐次処理フラグを格納するよう設計した．これにより1次キャッシュミス時のブロック転送時逐次フラグを生成する必要がなく，性能への影響の大きいブロック転送によるΔCPIの更なる増加を抑えることができる．各並列処理可否判定回路は，1個当り1270ゲートとなる．

4.2.2 性　　能

図6に並列処理不可で生ずるCPI増加（ΔCPI）による方式評価を示す．これは，WS上の性能評価シミュレータでSPECint 92を実行させて，命令間の依存関係により並列処理不可となり発生するΔCPIを計算した結果である．(x)がキャッシュ内フラグ方式によるΔCPI，(y)がキャッシュ/バッファ内フラグ方式によるΔCPIである．キャッシュ/バッファ内フラグ方式はキャッシュ内フラグ方式に比べ，異なるブロック転送単位間での並列処理制約を除くことでΔCPIは平均0.1改善できる．なお，本ベンチマークは全命令に占める分岐命令の割合が20〜27%と高く，このように分岐の発生頻度が高いアプリケーションで特にキャッシュ/バッファ内フラグ方式は有効である．

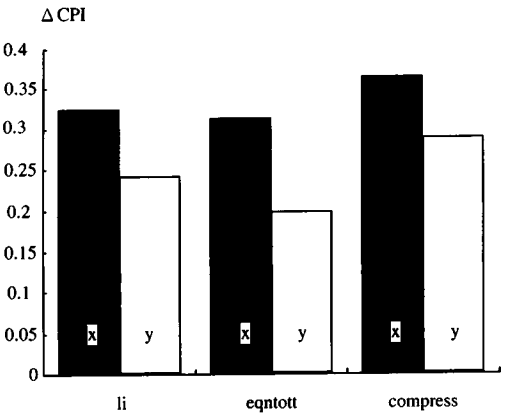


図 6 並列処理不可で生ずるCPI増加（ΔCPI）による方式評価
Fig. 6 The evaluation with ΔCPI of parallel execution check.
(x) Flag in cache
(y) Flag in cache/in IBR

4.3 分岐/無効化混合型動的予測方式の評価

4.3.1 ハード量

1024エントリの無効化予測混合型分岐履歴テーブルは，大きさは0.7平方ミリで，1エントリ当り5ビットの情報で構成する．2ビットは2ビットカウンタ方式の分岐履歴，2ビットはキャッシュから読み出す16バイト（4命令）の中で予測した分岐位置，1ビットはパリティビットである．また，無効化予測のためのハードウェア増加は，履歴テーブルへの登録論理部に無効化予測で分岐不成立を登録するよう追加しただけのため，数ゲート規模と小さい．

4.3.2 性　　能

図7は，分岐により生ずるCPI増加（ΔCPI）による方式評価である．評価方法は図6と同様にSPECint 89を実行させた結果である．(x)が静的分岐予測方式と静的無効化予測方式を組み合わせた場合の分岐によるΔCPI，(y)が分岐/無効化混合型動的予測方式（動的分岐予測方式と動的無効化予測方式を組み合わせた場合）の分岐によるΔCPIである．(x)

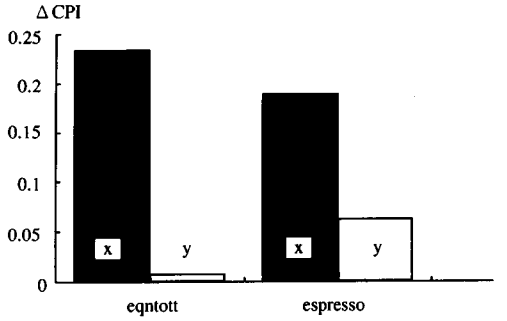


図 7 分岐で生ずる CPI 増加（⊿CPI）による方式評価
Fig. 7 The evaluation with ⊿CPI of branch.
(x) Static branch prediction and static nullified prediction
(y) Branch/nullified merged dynamic prediction (dynamic branch prediction and dynamic nullified prediction)

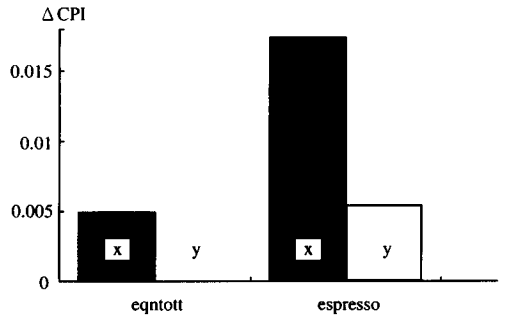


図 8 無効化分岐で生ずる CPI 増加（⊿CPI）による方式評価
Fig. 8 The evaluation with ⊿CPI of nullified branch.
(x) Static nullified prediction
(y) Branch/nullified merged dynamic prediction

では，静的分岐予測として BTFN を用いており，(y)では，動的分岐予測として 2 ビットカウンタ方式を用いている．分岐/無効化混合型動的予測方式は，静的予測に対して ⊿CPI にして最大 0.1～0.2 の効果がある．

図 8 は，無効化分岐で生ずる CPI 増加（⊿CPI）による方式評価である．評価方法は図 7 と同様である．(x)が静的無効化予測方式の場合の無効化分岐による ⊿CPI，(y)が分岐/無効化混合型動的予測方式の場合の無効化分岐による ⊿CPI である．分岐/無効化混合型動的予測方式は，静的無効化予測方式に比べ ⊿CPI にして最大 0.01 の効果がある．

5. むすび

スーパスカラプロセッサの命令先取り分配方式として，以下の方式を提案した．

(1) キャッシュ/バッファ内フラグ方式を提案した．これにより，命令先取り分配のレイテンシをのばさず，異なるブロック転送単位間での並列処理制約なしに並列処理可否判定が可能となり，従来方式から CPI 換算で 0.1 改善できる見通しを得た．

(2) 動的分岐予測と独立に無効化処理を動的に予測する動的無効化予測方式を提案し，分岐命令の無効化に限定した分岐/無効化混合型動的予測方式を実現した．これにより，従来方式からわずか数ゲートのハードウェア増加で性能を CPI 換算で 0.01 改善できる見通しを得た．

また，上記方式を採用した 150 MHz 多重度 2 のスーパスカラプロセッサを 0.3 μm CMOS プロセス技術を用いて開発し，その実用性を実証した．

謝辞 研究の機会を与えて頂いた日立コンピュータエンジニアリングの橋本眞宏氏，日立製作所汎用コンピュータ事業部の安斎昭夫氏に感謝致します．

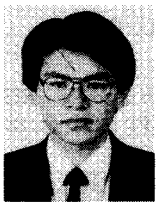
文　献

[1] P. Gronowski, et al., "A 433 MHz 64 b quad-issue RISC microprocessor," Proc. of ISSCC, [FP 13. 7], pp. 222-223, 1996.
[2] J. Montanaro, et al., "A 160 MHz 32 b 0.5 W CMOS RISC microprocessor," Proc. of ISSCC, [FP 13. 3], pp. 214-215, 1996.
[3] D. A. Patterson and C. H. Sequin, "A VLSI RISC," IEEE Computer, vol. 28, no. 1, pp. 8-21, Jan. 1985.
[4] G. F. Grohoski, "Machine organization of the IBM RISC System/6000 processor," IBM J. of Research and Development, vol. 34, no. 1, pp. 37-58, Jan. 1990.
[5] M. Johnson, "Superscalar microprocessor design," Prentice Hall, 1991.
[6] G. Blanck and S. Krueger, "The superSPARC microprocessor," IEEE Compcon 92, pp. 136-141, 1992.
[7] J. H. Edomondson, et al., "Superscalar instruction execution in the 21164 alpha microprocessor," IEEE micro, vol. 15, no. 2, pp. 33-43, April 1995.
[8] S. Tanaka, et al., "A 120-MHz BiCMOS superscalar RISC processor," IEEE J. Solid-State Circuits, vol. 29, no. 4, pp. 389-396, April 1994.
[9] R. Lee, M. Mahon, and D. Morris, "Pathlength reduction features in the PA-RISC architecture," IEEE Compcon 92, pp. 129-135, 1992.
[10] J. K. F. Lee and A. J. Smith, "Branch prediction strategies and branch target buffer design," IEEE Computer, vol. 17, no. 1, pp. 6-21, Jan. 1984.

[11] T. Y. Yeh and N. Patt, "Alternative implementations of two-level adaptive branch prediction," Proc. of the 19th ISCA, pp. 124-134, 1992.
[12] K. Shimamura, et al., "A superscalar RISC processor with pseudo vector processing feature," Proc. of ICCD, pp. 102-109, Oct. 1995.
[13] 斉藤拡二, 他"疑似ベクトル機構を有する並列コンピュータ向け RISC プロセッサ," 信学技報, vol. 95, no. 301, Oct. 1995.
[14] 中澤喜三郎, 他"超並列計算機 CP-PACS のアーキテクチャ," 情報処理, vol. 37, no. 1, pp. 18-28, Jan. 1996.
(平成 8 年 3 月 8 日受付, 6 月 14 日受付)

吉田 昌司 (正員)

昭 63 東大・工・計数卒. 同年(株)日立製作所入社. 以後, 同社日立研究所において, RISC プロセッサおよびコントローラの研究開発に従事. 現在, 同所情報制御第一研究部に所属.



田中 成弥 (正員)

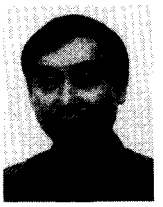
昭 58 豊橋技科大・工・電気電子卒. 昭 60 同大大学院修士課程了. 同年(株)日立製作所入社. BiCMOS 技術を使った高速マイクロプロセッサの研究開発の後, 平 7 より制御用コントローラの研究に従事. 現在, 同社日立研究所情報制御第一研究部に所属. 情報処理学会会員.

松尾康太郎

平 2 コーネル大・物理卒. 同年(株)日立製作所入社. 以後, 同社デバイス開発センターにおいて, RISC プロセッサの開発に従事.

堀田多加志 (正員)

昭 56 東大・工・電気卒. 昭 58 同大大学院修士課程了. 同年(株)日立製作所入社. 高速マイクロプロセッサ, およびコントローラの研究に従事. 現在, 同社日立研究所情報制御第一研究部に所属. IEEE, ACM, 情報処理学会各会員. 工博.



澤本 英雄 (正員)

昭 51 京大・工・電気卒. 昭 53 同大大学院修士課程了. 同年(株)日立製作所入社. 昭 60 エール大大学院修士課程了. 以後, 同社汎用コンピュータ事業部において, CPU, 汎用コンピュータ, RISC プロセッサの開発に従事. 現在, 同事業部プロセッサ開発センタ主任技師.

清水 照久 (正員)

昭 59 京大・工・情報卒. 昭 61 同大大学院修士課程了. 同年(株)日立製作所入社. 以後, 同社デバイス開発センターにおいて, 高性能プロセッサの開発に従事.