パイプラインステージ統合における ALU Inlining

ALU Inlining within Pipiline Stage Unification

尾形幸亮[†] 嶋田創[†] 中島康彦[††]
森眞一郎[†††] 富田眞治[†]
Kosuke Ogata,[†] Hajime Shimada,[†] Shinichiro Mori,[††]
Yasuhiko Nakashima[†††] and Shinji Tomita[†]

1. はじめに

近年，VLSI におけるトランジスタの集積度は飛躍的に向上し，プロセッサ 1 個を構成するトランジスタ数は増加する一方である．しかし，高集積化の代償としてプロセッサの消費電力も年々増大する傾向にあり，近年のプロセッサ設計においては性能向上とともに消費電力削減も重要な課題となっている．この消費電力の削減のために，近年のプロセッサにおいてよく利用されるのが Dynamic Voltage Scaling(以下 DVS) である．これは，プロセッサの負荷が低くなったときに電源電圧およびクロック周波数を低下させて電力削減を図る技術である．しかし，将来のプロセス技術では，ソフト・エラーの増加やサブスレッショルド・リーク電流対策の点から電源電圧を低下させるのが難しくなるため，DVS はその有効性を徐々に減らすと考えられる．

そこで DVS とは異なる電力削減手法としてパイプラインステージ統合 (Pipeline Stage Unification，以下 PSU) が提案されている [1,2]．PSU では，プロセッサ負荷とプログラムの並列度に応じてパイプラインを規定段数ごとに統合する．これにより，統合されたステージ内で使われなくなったパイプライン・レジスタにはクロックを供給する必要がなくなり，消費電力削減が行える．また，統合により短くなったパイプラインによって IPC(Instructions Per Cycle) が向上するため，同一の処理をより低いクロック周波数で行える．これによっても消費電力が削減される．

この PSU を適用したパイプラインにおいては現在，実行ステージを近隣のステージと統合しないと仮定している．そのため，PSU を適用しつつクロック周波数を下げると実行ステージの後半に何も処理が行われない空き時間が生じる．そこで，本稿ではこの空き時間を活用するための ALU Inlining という手法を適用する．この手法では，クロック周波数低下時に 1 つの ALU の出力をほかの ALU の入力につなぎ，データ依存関係にある演算命令を連続して実行する．これにより，PSU による IPC 向上はさらに大きくなる．

本稿では，まず第 2 章，第 3 章で PSU および ALU Inlining について詳細を述べ，PSU の特徴および ALU Inlining の意義について明確にする．次に，第 4 章で ALU Inlining を追加したプロセッサ・シミュレータによる評価結果を示し，ALU Inlining によってプロセッサの性能が向上することを示す．最後に第 5 章でまとめを述べる．

2. パイプラインステージ統合 (PSU)

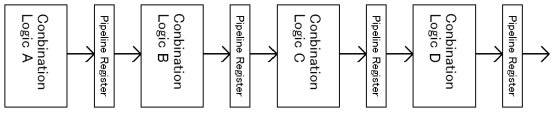

図 1 PSU 適用前のパイプライン

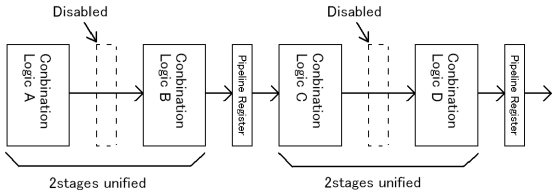

図 2 統合度 2 の PSU 適用後のパイプライン

PSU では，図 1，2 のようにパイプライン・レジスタの一部を無効化し，無効化されたパイプライン・レジスタの前後にある信号線をダイレクトに接続する．これにより，連続する複数のステージを統合し，1 クロックでデータに対して複数ステージの処理を行う．図 1，2 では例として 2 ステージ統合 (これを統合度 2 と呼ぶ) の PSU を示しているが，これと同様にして，

† 京都大学情報学研究科
Graduate school of Informatics, Kyoto University
†† 奈良先端技術大学院大学情報科学研究科
Graduate school of Information Science, NAIST
††† 福井大学工学研究科
Graduate school of Engineering, Fukui University

より多数のステージを統合することも可能である．統合するステージ数はプロセッサの負荷とプログラムの並列度に応じて動的に変化させ，基本的にプロセッサの負荷が低くて並列度が低いときほど多数のステージを統合するようにする．また，ステージ統合中は 1 クロック・サイクルで通過するゲート数が増えるため，クロック周波数を統合度に応じて 1/2 や 1/4 に引き下げる．

プロセッサに PSU を適用することにより，以下に示す 2 点の理由から消費電力を削減可能である．

- 無効化されたパイプライン・レジスタへのクロック供給を停止することにより，クロック・ドライバの消費電力を削減できる
- PSU によりパイプラインが短くなり，分岐予測ミス・ペナルティなどが減るため，IPC が増加する．これにより，より低いクロック周波数で同じ性能を達成できるため，クロック周波数低下に応じた消費電力を削減できる

文献 1) によれば，PSU を適用することにより，IPC は統合度 2 のとき平均 48%向上し，また統合度 4 のときは平均 95%向上する．

3. ALU Inlining

3.1 ALU Inlining の原理

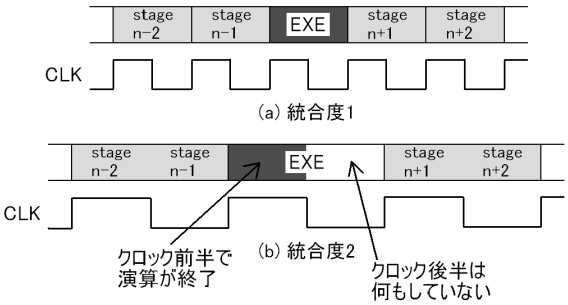

図 3 統合度 2 の PSU を適用した場合

現在の PSU の仮定では，実行ステージをほかのステージと統合しないようにしている．これは実行ステージの前後に結果フォワーディングのためのフィードバック・ループがあるためである．そのため，この仮定のもとでは PSU 適用時に実行ステージに何も処理が行われない空き時間が生じる．たとえば統合度 2 の PSU を適用する場合を考えると，図 3 のように実行ステージ以外のステージが 2 つずつ統合され，統合されたステージの処理時間に合わせてクロック周波数は PSU 適用前の 1/2 になる．この状態でパイプラインを動作させると，PSU 適用前に実行レイテンシが 1 クロックだった演算命令を実行した場合，ALU で処理が行われる時間はクロック・サイクル時間の前半だけになってしまい，クロック・サイクル時間の後半では何も処理が行われないことになる．

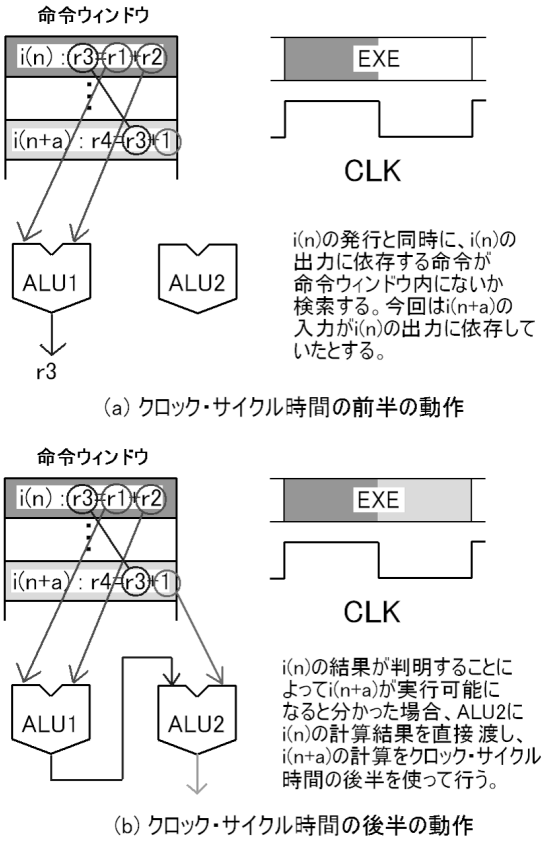

図 4 ALU Inlining の動作イメージ図

そこで本稿では，このクロック・サイクル時間の後半の何も処理が行われていない空き時間も演算処理に利用できる，ALU Inlining という手法を適用する．以下で統合度 2 の PSU を適用した場合における ALU Inlining を例に挙げて述べる．まず，命令ウィンドウにおいてデータ依存関係が満たされた演算命令 (命令 n とする) を ALU1 に発行する際，命令ウィンドウ内に命令 n と真の依存関係にある命令がないか検索する (図 4(a))．もし依存関係にある演算命令 (命令 n+a とする) を発見したら，命令 n の計算結果が出次第ただちに命令 n+a が実行可能となるかどうか調べる．また，同時に命令 n+a を実行可能な ALU が空いているかどうかも調べる．命令 n+a が上記の 2 条件を満たすとき，命令ウィンドウは命令 n と同時に命令 n+a も ALU2 へ発行する．なお，上記の 2 条件を満たすような命令が複数ある場合は，それらの中から命令ウィンドウ内で最も長く存在するものを選択する．ALU1 に発行された命令 n はクロック・サイクル時間の前半

を使って計算されるが，ここで ALU1 の出力をレジスタへ送るだけでなく ALU2 の入力へもバイパスするようにする (図 4(b))．ALU2 では ALU1 の出力を取り込んで，クロック・サイクル時間の後半を使って命令 n+a を実行する．このようにして，従来の PSU では何の処理も行われていなかった，クロック・サイクル時間の後半を有効利用することができる．なお，ALU Inlining の適用対象となる命令は，PSU 非適用時に実行レイテンシが 1 サイクルである演算命令とする．すなわち，整数加減算，論理演算，シフト演算といった単純な命令が対象となり，乗除算や浮動小数点演算など，実行レイテンシが 2 サイクル以上となる複雑な命令は除外する．

3.2 複数段の ALU Inlining

3 段以上の PSU では，その段数に応じた ALU Inlining を行える．3 段以上の PSU とともに適用する ALU Inlining は，以下のようなアルゴリズムで ALU Inlining を行う命令を選択する．

(1) 2 段の ALU Inlining と同様に，現在発行しようとしている演算命令と依存関係にある演算命令を命令ウィンドウから探す．複数候補がある場合は，命令ウィンドウ中に最も長く存在する演算命令を選択する

(2) 手順 (1) で選択した演算命令と依存関係にある演算命令を命令ウィンドウから探す．複数候補がある場合は手順 (1) と同様にして演算命令を 1 つ選択する

(3) 依存関係にある演算命令が見つからなくなるか，ALU の空きがなくなるか，統合度と同数の演算命令を選択するまで手順 (2) を続ける

(4) 現在発行しようとしている演算命令とともに，選択した演算命令を発行する

4. ALU Inlining の効果の検証

4.1 評価環境

ALU Inlining の評価には，SimpleScalar Tool Set[4] を用い，その中に含まれる out-of-order 実行シミュレータに ALU Inlining を実装した．PSU の統合度は 1(PSU 非適用)，2，4 のときを仮定し，統合度 2 のときは 2 段の ALU Inlining を，統合度 4 のときは 4 段の ALU Inlining を機能させてシミュレーションを行い，それぞれにおける IPC の値を測定した．また，従来の仮定と比較するため，各統合度の PSU 適用時において ALU Inlining を無効にしたときの IPC の値も測定し，ALU Inlining の効果を評価した．ベンチマーク・プログラムは，SPECint95 の 8 本を用いた．

4.2 評価結果

4.2.1 ALU Inlining による IPC 向上度

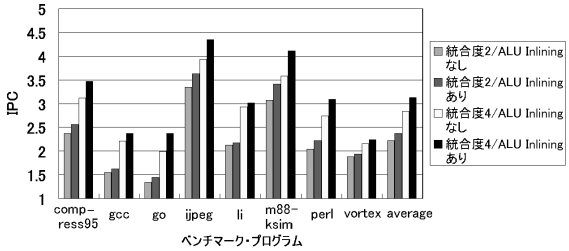

図 5 ALU Inlining による IPC 向上

図 5 に ALU Inlining による IPC 向上を示す．グラフ横軸はベンチマーク・プログラム，縦軸は IPC を表す．各ベンチマークに対する棒グラフは左から，PSU 統合度 2 で ALU Inlining を適用しなかった場合，PSU 統合度 2 で ALU Inlining を適用した場合，PSU 統合度 4 で ALU Inlining を適用しなかった場合，PSU 統合度 4 で ALU Inlining を適用した場合，となっている．いずれのベンチマークでも，ALU Inlining を適用することにより IPC の向上が見られた．特に ijpeg と m88ksim において IPC 向上が顕著であり，最も高い向上率を示した m88ksim の場合，2 段の ALU Inlining で 11.0%，4 段の ALU Inlining で 14.9%IPC が向上した．一方，li のように IPC がほとんど向上しない場合もあった．これは，演算命令間のデータ依存がほどんどなかったためと考えられる．平均 IPC 向上率は 2 段の ALU Inlining で 6.77%，4 段の ALU Inlining で 10.3%となった．同一の性能を必要とする場合，IPC の向上に反比例してクロック周波数を下げることができる．よって，性能向上を欲しない場合，それぞれ 6.34%と 9.34%クロック周波数を低下させることができ，それに応じた消費電力削減を達成できる．

4.2.2 ALU Inlining の適用率

図 6 に ALU Inlining の適用率を示す．ここで適用率とは，ALU Inlining の 2 段目以降に組み込まれた演算命令の数を，ALU Inlining の適用対象となる全ての演算命令の数で割った値を指す．すなわち，ALU Inlining によって本来は並列実行できなかった演算命令をどの程度並列実行に組み込めたかを示す．ただし，ここで集計した演算命令には分岐予測ミスで選択されたパス上の命令を含まない．グラフの横軸はベンチマーク・プログラムを，縦軸は適用率を表す．各ベンチマークに対する棒グラフは，左が PSU 統合度 2 のときの 2 段 ALU Inlining 適用率，右が PSU 統合度 4 のときの 4 段 ALU Inlining 適用率である．右側の

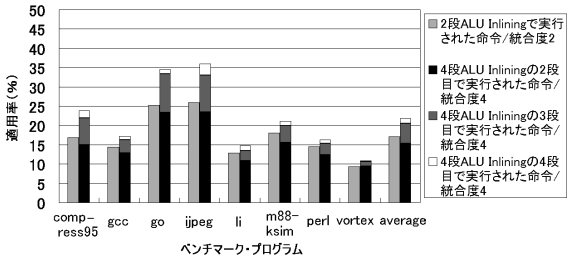

図 6 ALU Inlining の適用率

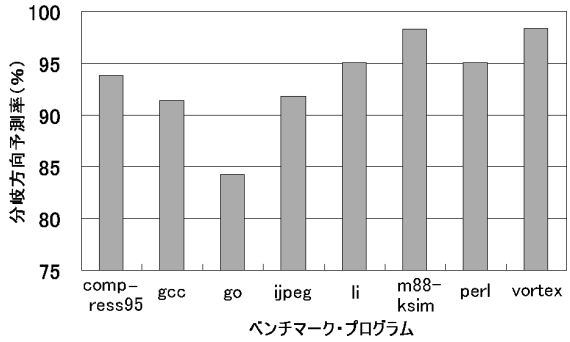

図 7 統合度 2 の PSU 適用時における分岐方向予測率

棒グラフについては，下から ALU Inlining の 2 段目に組み込まれた演算命令，3 段目に組み込まれた演算命令，4 段目に組み込まれた演算命令となっている．基本的に，図 5 において IPC 向上率が高いほど，図 6 でも適用率が高くなっていることが分かる．平均適用率は 2 段の場合 17.1%，4 段の場合は各段の合計で 21.9%となった．ただし，統合度 2 の PSU 適用時の go のように ALU Inlining 適用率が高いにも関わらず IPC があまり向上しない場合や，ijpeg と m88ksim のように IPC 向上率と ALU Inlining 適用率との高低が逆転している場合もある．この理由には以下のようなことが考えられる．プログラムを実行した際の分岐予測率が低い場合，分岐予測ミス・ペナルティによって演算の行われないサイクルが多くなるため，総実行サイクル数が多くなる．そのため，総実行サイクル中に占める演算命令の実行サイクル数の割合は相対的に低くなる．したがって，ALU Inlining を適用して演算命令の実行サイクル数を削減しても，その削減率を総実行サイクル中の割合として見ると小さくなってしまうため，IPC があまり向上しないものと考えられる．図 7 は統合度 2 の PSU 適用時における分岐方向予測率を示しているが，この図から分かるように，ijpeg の分岐方向予測率は 91.8%なのに対して，m88ksim の分岐方向予測率は 98.2%となっている．すなわち，ijpeg は m88ksim よりも分岐予測ミスを頻繁に起こしていることになる．また，go の分岐方向予測率は全ベンチマーク・プログラム中最低となっている．このように，分岐予測率が低い場合は ALU Inlining による IPC 向上が達成しにくくなると言える．

5. ま と め

本稿では，PSU の実行ステージにおける，何も処理が行われない空き時間を活用するために ALU Inlining を適用し，これによって IPC が向上することを示した．ALU Inlining により，8 本全てのベンチマークで IPC が向上し，最大 IPC 向上率は 2 段の場合 11.0%，4 段の場合 14.9%であった．また，平均 IPC 向上率は 2 段の ALU Inlining で 6.77%，4 段の ALU Inlining で 10.3%となった．ALU Inlining の適用率については，平均で 2 段のとき 17.1%，4 段のとき 21.9%であった．ただし，適用率と IPC 向上率は必ずしも一致するとは限らなかった．

参 考 文 献

1) 嶋田創, 安藤秀樹, 島田俊夫, ”パイプラインステージ統合によるプロセッサの消費エネルギーの削減,” 情報処理学会論文誌, コンピューティングシステム, Vol. 45, No. SIG 1 (ACS 4), pp.18-30, 2004 年 1 月.
2) 嶋田創, 安藤秀樹, 島田俊夫, ”パイプラインステージ統合とダイナミック・ボルテージ・スケーリングを併用したハイブリッド消費電力削減機構,” 2004 年先進的計算基盤システムシンポジウム SACSIS 2004, pp.11-18, 2004 年 5 月.
3) 佐々木広, 近藤正章, 中村宏, ”GALS 型プロセッサにおける動的命令カスケーディング,” 情報処理学会研究報告, 2005-ARC-164, pp. 67-72. 2005 年 8 月.
4) D. Burger and T.M. Austin, ”The SimpleScalar Tool Set, Version 2.0,” Technical Report CS-TR-97-1342, University of Wisconsin-Madison Computer Sciences Dept., July 1997.