

# バイパス専用 ALU を用いる事による 小面積高スループットプロセッサ

齋藤 和明<sup>1</sup> 三輪 忍<sup>2</sup> 中條 拓伯<sup>1</sup>

概要：命令レベルの並列性 (ILP) を有効に利用するため、発行幅を大きくすることは有効である。しかし、発行幅を大きくするにあたって、非演算回路の回路面積や消費電力の増加といった問題が引き起こされる。我々は特にレジスタ・ファイルの回路面積増加を抑え、その上でスループットを向上させるために、直近の演算結果を提供するバイパス機構に着目し、レジスタ・ファイル・アクセスの制限された ALU であるバイパス専用 ALU を提案する。提案手法であるバイパス専用 ALU を適用した場合、ALU 2 個の構成に対し、SPEC CINT2006 ベンチマークセットにおいて、平均 8.0% のスループット向上、SPEC CFP2006 ベンチマークセットにおいて平均 5.7% のスループット向上であった。

## 1. はじめに

命令レベルの並列性 (ILP) を有効に利用するため、発行幅を大きくすることは有効である。しかし、発行幅を大きくするにあたって、非演算回路の回路面積や消費電力の増加といった問題が引き起こされる。特に、回路面積の増加によって、そのユニットのレイテンシが増加してしまうため、元々と同じ周波数で動作させるには、パイプライン化せざるを得ない。しかし、命令スケジューラのパイプライン化は一般的にプロセッサの性能低下を引き起こす。また、レジスタ・ファイルのパイプライン化は、分岐予測ミス・ペナルティの増加や、バイパス機構の複雑化を招くこととなる。レジスタ・ファイルは、多ポートの RAM で構成されている。レジスタ・ファイルの回路面積は、レジスタ・ファイル・ポート数の二乗に比例して増加することがわかっている。また、パイプライン化されているため、レジスタ・ファイル経由でのオペランド・アクセスが不可能な期間が存在する。通常、レジスタ・ファイル経由でのオペランド・アクセスが不可能な期間にも直近の演算結果を利用し、後続の命令を連続して発行することによって命令のスループットを向上させるために、バイパス (フォワードリング) 機構が設けられ、高速化がはかられている。スループットを向上させる手法として、ALU Cascading [1]

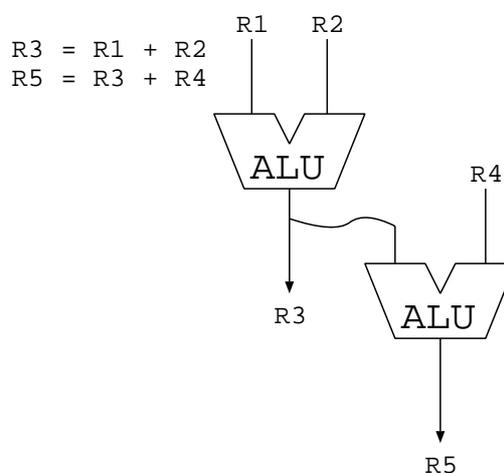


図 1 ALU Cascading

がある。

これは、図1のように、ある ALU の出力を別の ALU の入力へ繋ぎ、オペランドを後続の命令に受け渡すことによって 1 クロック・サイクル中に複数の命令を実行しスループットを向上させるものである。この ALU Cascading では、スーパーカラ・プロセッサに適用する際に、ALU Cascading 可能な命令の組の wakeup を同時に行う必要がある。その上、依存関係にある命令を同サイクル中に実行するため、ALU Cascading ではプロセッサの 1 サイクルを ALU 2 つ分の遅延が許容できるレベルまで落とす必要がある。本論文で

<sup>1</sup> 東京農工大学大学院 情報工学専攻  
<sup>2</sup> 東京大学大学院情報理工学系研究科

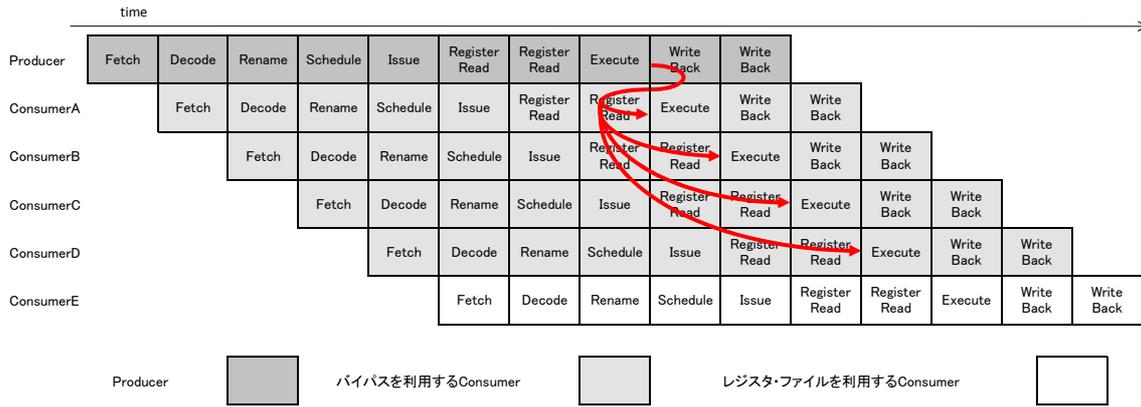


図 2 バイプライン・チャート

は、バイパス機構に着目したバイパス専用 ALU について提案を行う。バイパス専用 ALU (Bypass-Dedicated ALUs : BD-ALU) では、プロセッサの動作周波数を落とさず、レジスタ・ファイル・アクセスの制限された ALU を用い、バイパス機構からオペランドの取得ができる命令だけを対象として演算を行う。これによって、ALU 数の増加によるレジスタ・ファイルの回路規模増加を抑えつつスループットの向上を狙っている。INT ユニットにバイパス専用 ALU を適用した結果、INT-ALU 3 個のプロセッサと同等のポート数で、INT-ALU 2 個のモデルに対し SPEC CINT2006 において最大で 26.2%、平均で 7.3%、CFP2006 において、最大で 22.9%、平均で 5.3% スループットが向上する結果となった。本論文では、次章で関連研究について述べる。続く 3 章で提案手法であるバイパス専用 ALU を説明する。4 章では、サイクル・アキュレート・シミュレータにバイパス専用 ALU を実装した場合のスループットの評価を行う。5 章はまとめとする。

## 2. 関連研究

これまで、レジスタ・ファイルの回路面積削減を目的とした研究は多くある。レジスタ・ファイルの回路面積削減のため、物理レジスタの割当てを遅らせる方法 [2], [3], 分割したレジスタ・ファイルを階層的に接続する階層型レジスタ・ファイル [4] や、ビット分割レジスタ・ファイル [5] が提案されている。特に、レジスタ・ファイルのポート数削減の為に、マルチバンク化 [6], [7], [8] やレジスタ・ファイル・キャッシュ [4], [9], [10], などが提案されている。本研究における主たる目的は、レジスタ・ファイル・ポート数の削減ではなく、レジスタ・ファイル・ポート数増加によるレジスタ・ファイルの回路面積増大をできるだけ抑えながら高いスループットを実現することとしている。

## 3. バイパス専用 ALU

バイパスからオペランド取得可能な期間は、レジスタ・ファイル・アクセスに要するサイクル数によって異なる。

図 2 に、レジスタ・リード、及び、ライトに 2 サイクルをあてた場合のバイプラインチャートを示す。レジスタ・ファイルを通じて依存元命令 (Producer) から、依存先命令 (Consumer) へとオペランドを受け渡すには、少なくとも 4 サイクル以上あいている必要がある (Producer と ConsumerE の関係)。Producer の実行完了から Consumer の実行開始までが 3 サイクル以下の場合、バイパス機構を用いてオペランドを受け渡す必要がある。バイパス機構を利用して、オペランドを取得して実行する命令の割合を確認するため、SPEC CINT2006 ベンチマークを用いて予備評価を行った。SPEC CINT2006 ベンチマークにおいて、

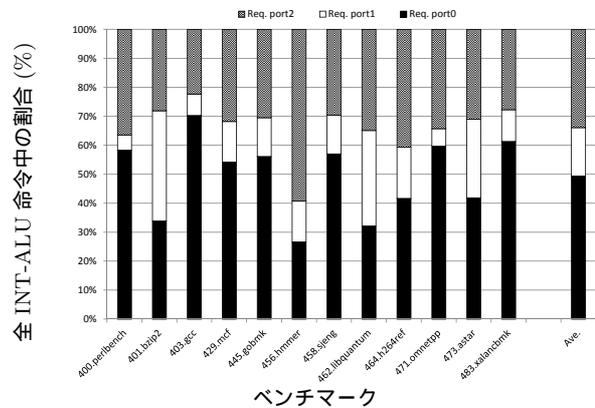


図 3 全 INT-ALU 命令の必要ポート数の内訳

予備評価を行った結果を図 3 に示す。ALU 8 個、レジスタ・リード・ポート 16 のレジスタ・ファイルに完全に接続された構成のプロセッサにおいて、SPEC CINT2006 から 12 本を用い、入力 train0、1G 命令スキップ後、100M 命令を実行した。全 INT-ALU 命令中、バイパスからオペランドを取得して実行される命令のうち、必要とされるリード・ポート数の割合を示している。横軸がベンチマーク名、縦軸が割合を示している。黒で示されている部分がバイパスだけからオペランドを取得して実行することが可能なもの、白で示されている部分がレジスタ・ファイルのリード・ポートへ 1 つ接続されている必要があるもの、網掛けが実行時に両方のオペランドをレジスタ・ファイルが

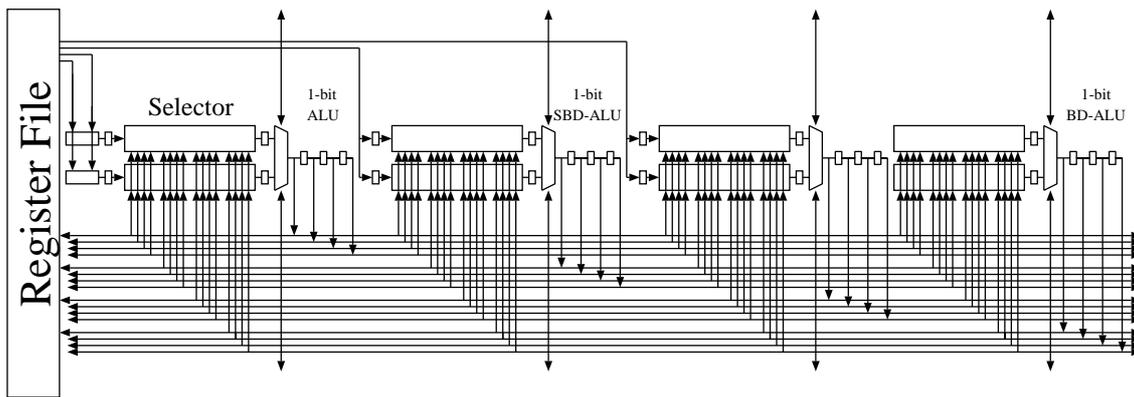


図 4 構成図 (1 ビット・スライス)

ら取得する必要があるものである。リード・ポートへ接続されている必要があるものに関しては、右オペランド・左オペランドの区別無しに必要なレジスタ・ファイル・ポート数を 1 としている。最大で約 80% の命令がオペランドをバイパスから取得して実行されており、平均で約 68% の命令がオペランドをバイパスから取得し実行されていることがわかった。この結果から、少なくともオペランドをバイパスから取得して実行される命令中の 7 割はバイパスだけからオペランドを取得しており、全命令中の 5 割となることがわかった。そこで、バイパス専用 ALU では、オペランドの依存関係が解決され、実行時点においてオペランドをバイパスから取得出来る命令の演算にレジスタ・ファイル・アクセスの制限された ALU を用いる。

バイパス機構を用いてオペランドを受け渡すことが可能な場合、レジスタ・ファイルにアクセスする必要は減るため、バイパス専用 ALU を用いて命令の実行を行うことが可能である。片方のオペランドのみをレジスタ・ファイルから取得出来るものを SBD-ALU、両方のオペランドをバイパスからだけ取得するものを BD-ALU、これらの総称としてバイパス専用 ALU と表記する。既存のプロセッサに本手法を用いて ALU 数を増やした場合、ALU 数の増加に伴うリード・ポートの増加が抑えられるため、レジスタ・ファイルの回路面積増加を抑えられることとなる。SBD-ALU を追加した場合には、リード・ポート、ライト・ポートそれぞれ 1 の増加、BD-ALU を追加した場合は、ライト・ポート 1 の増加となる。レジスタ・ファイルのリード・ポート数を 4 以上とした場合、これらの ALU を用意する組み合わせが複数個考えられる。図 4 に、発行幅 4 のプロセッサにおいて、1ALU+2SBD-ALU+1BD-ALU の構成 (1 ビット・スライス分) とした場合を示す。この図では、左端 1 つの ALU の入力にはレジスタ・ファイルのリード・ポートが完全に接続されているが、中央 2 つの SBD-ALU の入力にはレジスタ・ファイルのリード・ポートが 1 つだけ接続されており、右端 1 つの BD-ALU の入力にはバイパスだけが接続されている。さらに、全ての ALU において、ラ

イト・ポートが接続されている為、レジスタ・ファイルは合計 8 ポートの構成となる。また、SBD-ALU への命令の発行を行う際に左右のオペランドを区別していないが、レジスタ・ファイルから取得するオペランドを供給する機構は既存のセレクタを利用できるため、回路の複雑化は起こらない。

### 3.1 動作

命令スケジューラは、命令キュー内のすべてのオペランドがレディになったものの中から、命令キューに入ってきた順が早いものから発行していく。命令を発行する際に、命令スケジューラは発行対象命令が実行時点で必要とするリード・ポート数を把握しているものとする。発行しようとする命令の必要とするリード・ポート数に応じて、各 ALU の利用状況を確認し、発行可能と判断された場合、次サイクルで発行を開始する為に ALU の予約を行う。この動作を、そのサイクルでの発行の上限である発行幅一杯まで行うか、命令キュー内に発行可能な命令が存在しなくなるまで繰り返し行う。命令スケジューラは、すべてのオペランドがレディとなった命令のうち、古いものから順に発行していくが、完全にレジスタ・ファイルと ALU が接続されている場合と比較すると、リード・ポート数が制限されているために、ポート数不足によって発行が滞ってしまうケースがある。また、ポート不足以外にも、命令を発行できる ALU が不足しているために遅延させられてしまうケースがある。これは、必要ポート数が多い命令によるバイパス専用 ALU の予約は不可能である為である。例えば、SBD-ALU が 2 つ空いている状態でも、1 つの SBD-ALU に接続されているリード・ポートは 1 つに限られるため、動的に接続を変更し、両方のオペランドをレジスタ・ファイルから取得する必要がある命令を実行することは出来ない。これらのいずれかの条件に該当する場合、そのサイクルにおいて命令の発行は行われぬ。ALU が予約不可能な場合に起こる遅延によって、本来バイパスから取得できたはずのオペランドがレジスタ・ファイルに書き戻されて

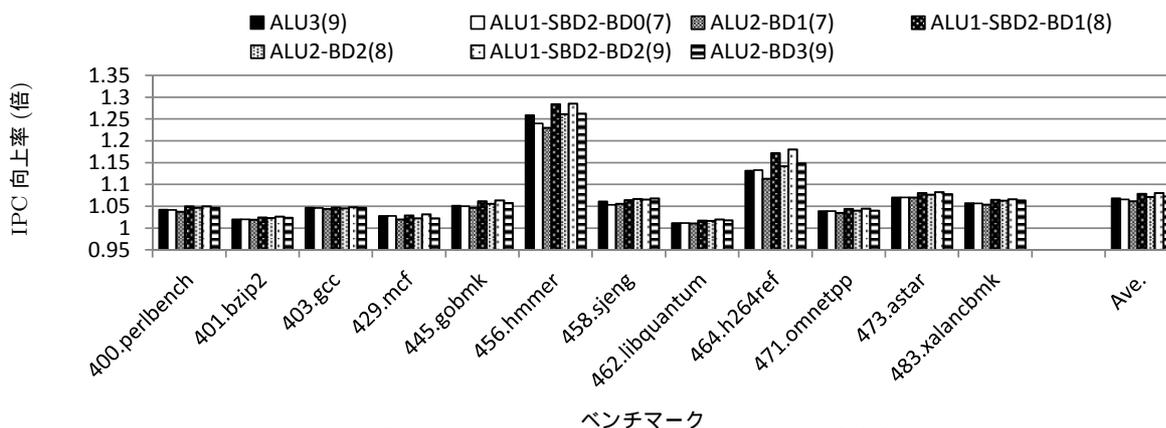


図 5 SPEC CINT2006 におけるスループット向上率

しまった場合、オペランドの取得をレジスタ・ファイルから行う必要があるため、本来 BD-ALU で処理できたものを SBD-ALU へ割り当てる必要が生じるか、パイパス専用 ALU 自体が機能しないこととなる。逆に、BD-ALU や、SBD-ALU を予約したい場合、それらの ALU が予約されてしまっている状況において、命令が要求しているポート数以上にレジスタ・ファイルに接続されている ALU に空きがあれば、そちらをを予約し、命令発行を行うことは可能である。その性質上、BD-ALU を利用すべき命令が遅延させられてしまうことは希であるが、SBD-ALU や通常の ALU を利用すべき命令は遅延させられてしまう可能性が高いことになる。

## 4. 性能評価

### 4.1 評価環境

評価には、サイクル・アキュレート・シミュレータ鬼斬式 [11] を用い、Alpha64-ISA、アウト・オブ・オーダーでのシミュレーションを行った。シミュレーションで仮定するプロセッサ環境を 1 に示す。パイパス専用 ALU は、INT-ALU に追加する形となっているため、INT 発行幅は構成に応じて変化する。次節にて都度表記する。シミュレーションは、SPEC CINT2006 ベンチマークセットから 12 本、SPEC CFP2006 ベンチマークセットから 16 本を用い、入力は train0、1G 命令スキップ後、100M 命令を実行してスループット (Instructions per cycle : IPC) の測定を行った。回路規模の評価には、Hewlett-Packard 社製 McPAT[12] を用いた。

### 4.2 スループットの評価

図 5 に、ALU 数 2 個の場合を 1 としたときの SPEC CINT2006 における IPC の向上率を示す。横軸はベンチマーク、縦軸はスループット向上率を示している。

凡例は上段左から比較のための ALU 数が 3 個の場合 (レジスタ・ファイル・ポート数 9)、レジスタ・ファイル・ポート数が 7 の ALU1 個+SBD-ALU2 個+BD-ALU0 個、ALU2

表 1 評価パラメータ

parameter	remarks
inst. queue(INT/FP/LS)	256/16/32
register file(INT/FP)	256/128, 2-cycle
fetch width	4
issue width(FP)	2
commit width	6
INT mul/div	1
FP ALU	1
FP mul/div	1
L1 I-cache	64KB 2-way, 64B-line, 2-cycle
L1 D-cache	64KB 2-way, 64B-line, 2-cycle
L2 unified cache	2048KB 2-way, 64B-line, 12-cycle
memory access latency	80cycle

個+BD-ALU1 個、レジスタ・ファイル・ポート数が 8 の ALU1 個+SBD-ALU2 個+BD-ALU1 個、ALU2 個+BD-ALU2 個の場合、そして、レジスタ・ファイル・ポート数が 9 の ALU2 個+BD-ALU3 個、ALU1 個+SBD-ALU2 個+BD-ALU2 個の場合を表している。INT 発行幅は基準となる ALU 数 2 個の場合において 2、その他については図の左から順に 3, 3, 3, 4, 4, 5, 5 である。

ALU2 個+BD-ALU1 個の構成では、全体的に ALU3 個の構成に対し性能が劣るが、レジスタ・ファイル・ポートの合計数が ALU3 個の場合と同等となる ALU2 個+BD-ALU3 個の構成では、高いスループットを示す事がわかる。ALU2 個+BD-ALU3 個の構成では特に hammer において、ALU2 個の構成に対し 26.2%の IPC の向上が見られた。一方、bzip2 や libquantum など、メモリアンテンシブなベンチマークにおいては、IPC の向上は 2%程度となった。平均では、ALU2 個+BD-ALU3 個の構成において、7.3%の IPC 向上であった。構成を ALU1 個+SBD-ALU2 個とした場合、矢張り全体的に ALU3 個の構成に対し性能が劣るが、そこに BD-ALU を追加し、レジスタ・ファイル・ポートの合計数が ALU3 個の場合と同等となる ALU1 個+SBD-ALU2 個+BD-ALU2 個の構成では、ALU2

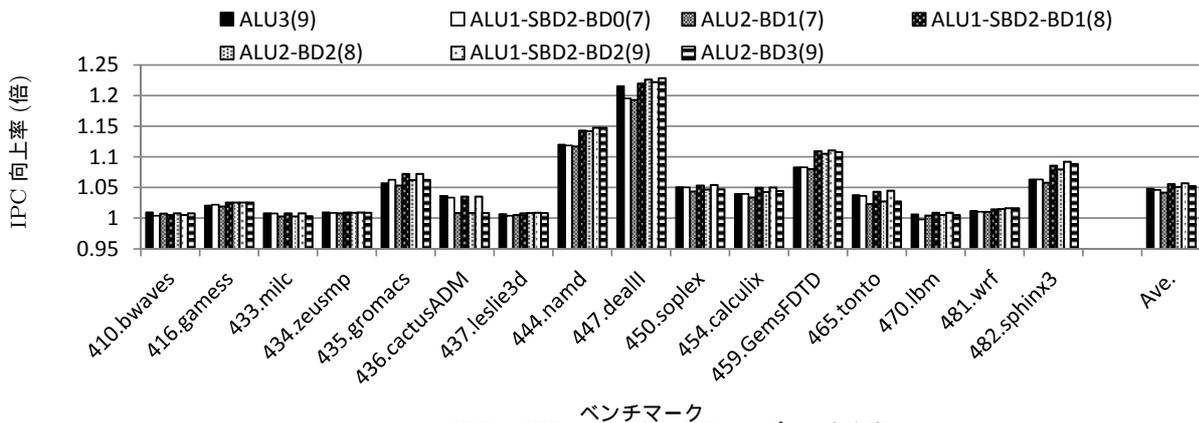


図 6 SPEC CFP2006 におけるスループット向上率

個+BD-ALU3 個の構成よりも高いスループットを示す事がわかる。ALU2 個に BD-ALU を追加した場合に対しても、両方のオペランドをレジスタ・ファイルから取得する ALU を減らし、SBD-ALU を用いた上で BD-ALU を追加した場合、より高いスループットを示している。ALU1 個+SBD-ALU2 個+BD-ALU2 個の構成では、特に hammer において、ALU2 個の構成に対し 28.6%の IPC の向上が見られた。平均では、8.0%の IPC 向上であった。

この結果から、基本的には ALU 数の多い構成の方が高いスループットが得られ、特に SBD-ALU を用いた場合に平均では良い結果が得られたが、ベンチマークによっては、両方のオペランドをレジスタ・ファイルから得ることの出来る ALU が多く存在する方が有利であることがわかった。

図 6 に ALU 数 2 個の場合を 1 としたときの SPEC CFP2006 における IPC の向上率を示す。横軸はベンチマーク、縦軸はスループット向上率を示している。

凡例は上段左から比較のための ALU 数が 3 個の場合 (レジスタ・ファイル・ポート数 9)、レジスタ・ファイル・ポート数が 7 の ALU1 個+SBD-ALU2 個+BD-ALU0 個、ALU2 個+BD-ALU1 個、レジスタ・ファイル・ポート数が 8 の ALU1 個+SBD-ALU2 個+BD-ALU1 個、ALU2 個+BD-ALU2 個の場合、そして、レジスタ・ファイル・ポート数が 9 の ALU2 個+BD-ALU3 個、ALU1 個+SBD-ALU2 個+BD-ALU2 個の場合を表している。INT 発行幅は基準となる ALU 数 2 個の場合において 2、その他については図の左から順に 3, 3, 3, 4, 4, 5, 5 である。

CFP2006 においては、INT-ALU 命令の絶対数が少ないためか、ALU の構成がスループットへ及ぼす影響が顕著に出ている。CINT2006 の場合と同様に、ALU2 個+BD-ALU1 個の構成では、全体的に ALU3 個の構成に対し性能が劣るが、レジスタ・ファイル・ポートの合計数が ALU3 個の場合と同等となる ALU2 個+BD-ALU3 個の構成では、高いスループットを示す事がわかる。しかし、一部のベンチマークにおいて、BD-ALU を追加しても IPC が頭打ちになってしまい、ALU3 個の構成のプロセッサに届かないこ

とがあった。これは、INT-ALU 命令間の依存関係が殆ど無く、BD-ALU を追加するよりも、単純にレジスタ・ファイルのリード・ポートに接続された ALU が多くある方が有利であるためだと考えられる。その他のベンチマークにおいては、ALU2 個+BD-ALU3 個の構成では特に dealII において、ALU2 個の構成に対し 22.9%の IPC の向上が見られた。平均では、ALU2 個+BD-ALU3 個の構成において、5.3%のスループット向上であった。構成を ALU1 個+SBD-ALU2 個とした場合、矢張り全体的に ALU3 個の構成に対し性能が劣るが、そこに BD-ALU を追加し、レジスタ・ファイル・ポートの合計数が ALU3 個の場合と同等となる ALU1 個+SBD-ALU2 個+BD-ALU2 個の構成では、ALU2 個+BD-ALU3 個の構成よりも高いスループットを示す事がわかる。最大 22.3%、平均では、5.7%の IPC 向上であった。

#### 4.3 回路規模の評価

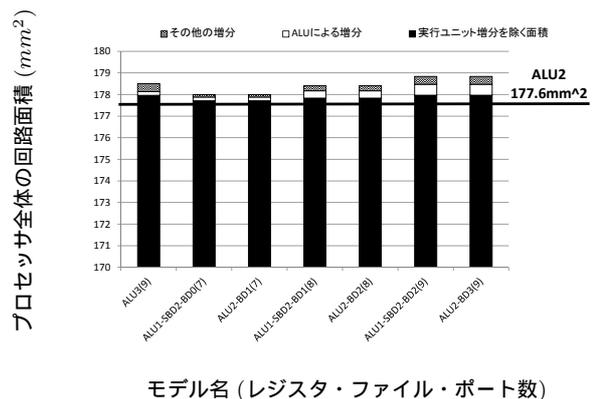


図 7 各モデルにおけるプロセッサ全体の面積比率

McPAT を用い回路規模の評価を行った。90nm プロセッサの DEC Alpha 21464 モデルにおける回路面積の比率を図 7 に示す。横軸がモデル、縦軸が面積を表している。黒線で表されているのが ALU が 2 個の場合のプロセッサ全体の回路面積であり、177.6mm<sup>2</sup> である。凡例は網がけで表されているのが実行ユニットにおける ALU 以外の回路

面積の増分であり、白で表されているのが ALU による回路面積の増分、黒が実行ユニットによる回路面積の増分を除くプロセッサの面積である。全てのプロセッサにおいて、INT レジスタ・ファイルのエントリ数は等しいため、レジスタ・ファイル・ポート数が 9 で等しくなる ALU3 と ALU1+SBD-ALU2+BD-ALU2, ALU2+BD-ALU3 の構成のレジスタ・ファイルの回路面積は等しい。同様に、レジスタ・ファイル・ポート数が 8 で等しくなる ALU1-SBD2-BD1 と ALU2+BD-ALU2 の組、レジスタ・ファイル・ポート数が 7 で等しくなる ALU1+SBD-ALU2+BD-ALU0 と ALU2+BD-ALU1 の組のレジスタ・ファイルの回路面積は等しい。ALU による回路面積の増加は、総 ALU 数が 5 個となる ALU2+BD-ALU3 や、ALU1+SBD-ALU2+BD-ALU2 の構成において  $0.51\text{mm}^2$ 、ALU 2 個のプロセッサ全体の回路面積に対し、0.28%増と、非常に小さくなっている。また、同モデルにおいて、ALU 2 個のプロセッサ全体の回路面積に対し、プロセッサ全体の回路面積の増加は 0.5%となっている。

## 5. まとめ

本論文では、バイパス機構に着目したバイパス専用 ALU の提案を行った。バイパス専用 ALU を適用したプロセッサを SPEC CINT2006 を用いて評価した結果、2 個の ALU を搭載したプロセッサに対し、ALU を 1 つに減らし、2 個の SBD-ALU、3 個の BD-ALU を追加することにより、最大で 28.6%、平均で 8.0%スループットが向上する結果となった。同様の構成において、SPEC CFP2006 を用いて評価した結果、最大で 22.3%、平均で 5.7%スループットが向上する結果となった。また、同モデルにおいて、ALU 2 個のプロセッサ全体の回路面積に対し、プロセッサ全体の回路面積の増加は 0.5%となった。ALU を追加する事による回路面積の増加は非常に小さく、且つ高スループットを達成できた。今後、バイパス専用 ALU を適用する際に、ライト・ポートを削減する手法を適用することにより、よりレジスタ・ファイル面積の増加を抑えた場合のスループット及び回路面積の評価を行う予定である。

## 参考文献

- [1] Motokazu Ozawa, Hiroshi Nakamura, and Takashi Nanya. Instruction execution mechanism based on cascade alu. *IPSI SIG Notes*, Vol. 99, No. 41, pp. 7–12, may 1999.
- [2] Teresa Monreal, Antonio Gonzalez, Mateo Valero, Jose Gonzalez, and Victor Vinals. Delaying physical register allocation through virtual-physical registers. In *In International Symposium on Microarchitecture*, pp. 186–192, 1999.
- [3] Antonio Gonzalez, Jose Gonzalez, Mateo Valero, C Jordi Girona, and Mdul D. Virtual-physical registers. In *In Proceedings of the 4th International Symposium on High-Performance Computer Architecture*, pp. 175–184,

- 1997.
- [4] Jose lorenzo Cruz, Antonio Gonzalez, Mateo Valero, and Nigel P. Topham. Multiple-banked register file architectures. In *In Proceedings of the 27th Annual International Symposium on Computer Architecture*, pp. 316–325, 2000.
- [5] KONDO MASAOKI and NAKAMURA HIROSHI. Reducing register port and size requirements by bit-partitioning(microarchitecture). *情報処理学会論文誌. コンピューティングシステム*, Vol. 46, No. 12, pp. 62–72, 2005-08-15.
- [6] Jessica H. Tseng and Krste Asanovic. Banked multiported register files for high-frequency superscalar microprocessors. In *In International Symposium on Computer Architecture*, pp. 62–71, 2003.
- [7] Steven Wallace, Steven Wallace, Nader Bagherzadeh, and Nader Bagherzadeh. A scalable register file architecture for dynamically scheduled processors. In *In International Conference on Parallel Architectures and Compilation Techniques*, pp. 179–184, 1996.
- [8] Rajeev Balasubramonian, Sandhya Dwarkadas, and David H. Albonesi. Reducing the complexity of the register file in dynamic superscalar processors. In *Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture*, MICRO 34, pp. 237–248, Washington, DC, USA, 2001. IEEE Computer Society.
- [9] Hui Zeng and Kanad Ghose. Register file caching for energy efficiency. In *Proceedings of the 2006 international symposium on Low power electronics and design*, ISLPED '06, pp. 244–249, New York, NY, USA, 2006. ACM.
- [10] Rajeev Balasubramonian, Sandhya Dwarkadas, and David H. Albonesi. Reducing the complexity of the register file in dynamic superscalar processors. *Microarchitecture, IEEE/ACM International Symposium on*, Vol. 0, p. 037, 2001.
- [11] 塩谷亮太, 五島正裕, 坂井修一. プロセッサ・シミュレータ「鬼斬式」の設計と実装. *先進的計算基盤システムシンポジウム SACSIS2009*, Vol. 2009, No. 4, pp. 120–121, 2009.
- [12] Sheng Li, Jung Ho Ahn, Richard D. Strong, Jay B. Brockman, Dean M. Tullsen, and Norman P. Jouppi. Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 42, pp. 469–480, New York, NY, USA, 2009. ACM.