

C-010

CHAIN 手法を用いたプロセッサのアウトオブオーダー実行 Out-of-order Execution of a Processor with CHAIN Technique.

山根 章吾† 孟 林† 小柳 滋‡
Syogo Yamane Lin Meng Shigeru Oyanagi

1. はじめに

近年の情報機器の普及と高速化に伴う消費電力の増加により、プロセッサは単純な高速化だけでなく、消費電力を抑えた上での発展が求められている。

しかし、現在主流である Out-of-order 命令発行形式は、スケジューリングを行う際に、オペランドの比較に必要なハードウェア量が膨大になることが問題となっている。

我々は CHAIN 手法を用いて命令の依存関係を解決し、In-order 発行でも、従来の Out-of-order 発行に近い IPC を目指して来た。同様のアプローチは Cascading [3][4], ALU Inlining[5]でも行われている

しかし、インオーダー命令発行では対応ができないプログラムもあるため、アウトオブオーダー命令発行をインオーダー命令発行に近い時間でスケジューリングが行えないだろうか考えた。

本研究ではスケジューリングを簡略化、つまり比較回数の減少によりハードウェア規模を抑えるため、新たに3つの命令発行形式を提案し、その効果を従来の命令発行形式と比較することにより考察する。

2. CHAIN 手法とは

CHAIN 手法とは並列に実行しようとしている命令の間にデータ依存が生じるときに、これらの命令を同時に発行し、生成されたデータを他のユニットにバイパスすることによりデータハザードを解消するものである。図1にCHAIN手法の概要を示す。図に示すように、演算ユニット1の出力を演算ユニット2の入力に接続することにより、データ依存のある2つの命令を同一クロックで並列に実行することができる。

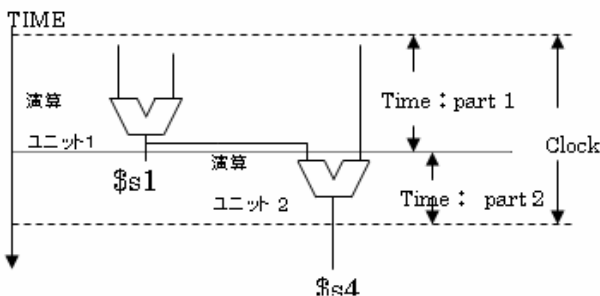


図 1 CHAIN 手法の概略図

3. 命令発行形式

今回 CHAIN 手法に適した命令発行形式を考察するため、従来の In-order 発行、Out-of-order 発行形式に加え、次の3つの命令発行形式を提案する。

1. Mixed-order
2. コンパイラ支援付き In-order
3. コンパイラ支援付き Out-of-order

3. 1 Mixed-order

Mixed-order 形式は、命令ウィンドウ内の命令のうち最大同時実行命令数の分だけ In-order で依存関係のチェックを行い、残りの命令を Out-of-order でチェックするものである。

In-order 部分で依存関係が存在した場合、Out-of-order 部分から In-order 部分との依存関係を含めて依存関係の無いものを発行する。これにより、全てを Out-of-order で発行した場合と比較して、IPC の減少を抑えた上で、依存関係チェックのために必要な比較器が減少し、ハードウェア規模の縮小を行える。

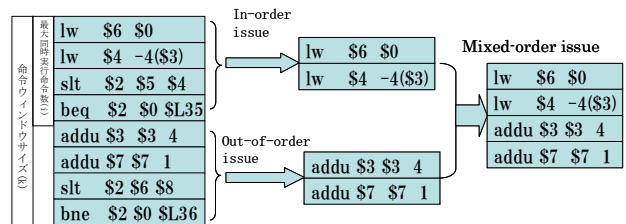


図 2 Mixed-order の解説図

3. 2 コンパイラ支援

コンパイラ支援とは、コンパイラによって予め依存関係を調べ、依存関係のある命令番号の一部をフラグとして命令に付加することで、スケジューリングを単純化し、命令の比較回数を大幅に減少させる。

命令形式を図3に示す。各命令には2つのソースオペランドに対応して2つのフラグをもつ。フラグの付け方としては、Addr には各ソースオペランドの最も近い依存関係をもつ命令アドレスの下位 5bit をフラグとして付加する。



図 3 命令形式

4. 命令発行方式の比較

4. 1 比較器の数の評価

最大同時実行命令数 i 、命令ウィンドウの大きさ k として、依存関係のチェックに必要な比較器の数について考察する。

In-order 発行の場合は、命令ウィンドウのサイズは i となり、各命令のディスティネーションレジスタと、後続の全てのレジスタを比較する必要があるため、比較器

† 立命館大学大学院, Graduate School of Science and Engineering, Ritsumeikan University

‡ 立命館大学, Ritsumeikan University

の数を表す数式としては次に示す形となる。

$$\sum_{n=1}^i 3(n-1) = \frac{3}{2}i(i-1)$$

Out-of-order 発行の場合は、In-order 発行に加えて、各命令のディスティネーションレジスタと、それに対する先行命令のソースオペランドを比較する作業が加わるため、数式としては次に示す形となる。

$$\sum_{n=1}^k \{3(n-1) + 2(n-1)\} = \frac{5}{2}k(k-1)$$

Mixed-order 方式の場合は、Out-of-order 発行に比べて、i の部分だけ各命令のディスティネーションレジスタと、先行命令のソースオペランドを比較する作業が必要となるため、以下の式となる。

$$\sum_{m=1}^k 5(m-1) - \sum_{n=1}^i (2n-1) = \frac{5}{2}k(k-1) - i(i-1)$$

コンパイラ支援付きの In-order 方式の場合、各命令アドレスと、後続命令のフラグとを比較する必要があるため、以下の式となる。

$$\sum_{n=1}^i 2(n-1) = i(i-1)$$

コンパイラ支援付きの Out-of-order 方式の場合、In-order に加えて、各命令の命令アドレスと、先行命令のフラグを比較する作業が必要となるため、以下の式となる。

$$\sum_{n=1}^k \{2(n-1) + 2(n-1)\} = 2k(k-1)$$

以上で示した、既存の 2 種と新しく提案する 3 種の比較器の数を計算すると表 1 のようになった。

表 1 各方式の比較器の数

最大実行数(i)	命令windowサイズ(k)	In-order	Out-of-order	Mixed-order	コンパイラ支援付きIn-order	コンパイラ支援付きOut-of-order
		$3/2*i(i-1)$	$5/2*k(k-1)$	$5/2*k(k-1) - 2*i(i-1)$	$i*(i-1)$	$2k*(k-1)$
4	6	18	75	63	12	60
4	7	18	105	93	12	84
4	8	18	140	128	12	112
8	12	84	330	274	56	264
8	16	84	600	544	56	480

各方式の比較器の数は少ない順にコンパイラ支援 In-order、In-order、コンパイラ支援 Out-of-order、Mixed-order、Out-of-order の順となった。

このことから、今回提案した方式は比較器の数の面では減少が確認できた。

4. 2 各方式の性能予測

命令発行方式の性能を予測するため、sort プログラムの一部を抜き出し机上実験を行った。机上実験では、分

岐予測を命令フェッチの時点で行うことが可能で、またその予測には間違いが無いものとして行っている。その結果、IPC はそれぞれ表 2 のようになった。

表 2 各方式での性能予測

命令発行形式	命令ウィンドウの大きさ	最大同時実行命令数	IPC
In-order発行	4	4	2.6
Out-of-order発行	6	4	3.4
Out-of-order発行	8	4	3.9
Mixed-order発行	6	4	3.4
Mixed-order発行	8	4	3.9
コンパイラ支援付きIn-order	4	4	2.6
コンパイラ支援付きOut-of-order	6	4	3.4
コンパイラ支援付きOut-of-order	8	4	3.9

4. 3 まとめ

それぞれの結果から、Mixed-order は Out-of-order よりも 10%少ない比較器の数で、ほぼ等しい IPC を出す事が確認できた。

コンパイラ支援付きの 2 種の命令発行形式は、従来の方式と比較とすると 20%少ない比較器の数で同一の IPC を出せることが確認できた。

以上のように、今回提案した方式は IPC と比較器の面で一定の効果が見られた。

5. おわりに

CHAIN 手法に適した命令発行方式として、Mixed-order およびコンパイラ支援方式を提案し、その効果を考察した。今後の課題としては、コンパイラ支援による発行形式については、理想的に動作するならば、最も少ない比較器の数で動作可能ではあるが、分岐先を内部で計算して決定する場合についての対応を考案し、また実験することが必要である。また両方の方式について言えることだが、実機での動作や、分岐ミスが増加した場合の動作の変化の確認などいくつかの課題が残されている。

参考文献

- [1] 安藤秀樹：命令レベル並列処理，コロナ社，2005
- [2] Mike Johnson, 村上和彰：スーパースカラ・プロセッサ，日経BP出版センター，1994.
- [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”，IPSJ，2005-ARC-164，pp 67-72，2005
- [4] 佐々木広，近藤正章，中村宏，“命令グルーピングにおける効率的な命令実行方式”，IPSJ，2006-ARC-170，pp 73-78，2006
- [5] 尾形 幸亮，嶋田創，中嶋康彦，森眞一郎，富田眞治，“パイプラインステージ統合における ALU Inlining”，平成 18 年度情報処理学会関西支部大会講演論文集，2006
- [6] 孟林，小柳滋，“スーパースカラプロセッサにおける動的 RENAME 手法と CHAIN 手法”，平成 18 年度情報処理学会関西支部大会講演論文集，pp. 207-210，2006 年