

スーパースカラプロセッサにおける CHAIN 手法とその効果

孟 林[†] 福原 潤一[‡] 小柳 滋[§]

立命館大学理工学研究科[†] 立命館大学[‡] 立命館大学[§]

1. はじめに

近年、VLSI 集積度の向上によって、プロセッサチップに集積されるトランジスタ数が、2年で3倍のペースで増加し続けてきている。また、プロセッサ高速化のため、クロック周波数も年率で30%向上している。しかし、プロセッサチップの微細化と高速化とともに、プロセッサの消費電力も年々増大している。消費電力の過密化による発熱困難もプロセッサチップの信頼性を低下させている。

本研究では、省電力とプロセッサ IPC の向上を考慮し、スーパースカラプロセッサの構築を行う。本プロセッサでは CHAIN 手法を用いる。CHAIN 手法は、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。この手法を用いて、ストールをさせずにハザードを解消することによって、電力を削減することができると思われる。それと共に、より単純なポリシーで命令を発行でき、プロセッサ IPC の向上によってより低いクロック周波数での動作もできるため、これらによってもプロセッサの省電力化をはかることができると思われる。

本論文では、まず第2章でスーパースカラプロセッサについて説明し、CHAIN 手法がプロセッサでしめる位置を説明し、さらにプロセッサの先見機構と発行ポリシーについても述べる。第3章では CHAIN 手法について詳細に説明する。第4章では SimpleScalar を用いて CHAIN 手法の効果について説明する。最後に第5章でまとめと今後の課題を述べる。

2. プロセッサの概要

2.1 スーパースカラプロセッサ

本スーパースカラプロセッサは、MIPS のパイプラインプロセッサをベースに構築されている。このプロセッサは、主に命令フェッチ (F)、命令の CHAIN (CH)、命令の Scheduling (SH)、命令デコードとレジスタファイル読み出し (D)、演算とアドレス生成 (E)、メモリアクセスとレジスタ書き込み (M)、レジスタ書き込み (W) の7部分で構成されている。演算とアドレス生成部 (E) は、CHAIN での連結を想定して、複数のロードストアユニット、分岐ユニット、シフトユニット、算術演算ユニットと比較演算ユニットで構成されている。本プロセッサは命令バッファをもち、命令バッファ内に4つの命令が含まれている。また、一つのレジスタファイル (Register File) も持っている。図1はスーパースカラプロセッサのブロック図を示している。

CHAIN モジュールでは、命令バッファの中の命令をデコードし、CHAIN を行うかどうかを判断し、CHAIN を行

う必要があるときに、関連の制御信号を作り、スケジューリングモジュールに渡す。演算部分では CHAIN モジュールで作られた信号を利用し、必要なユニットについて CHAIN を行う。

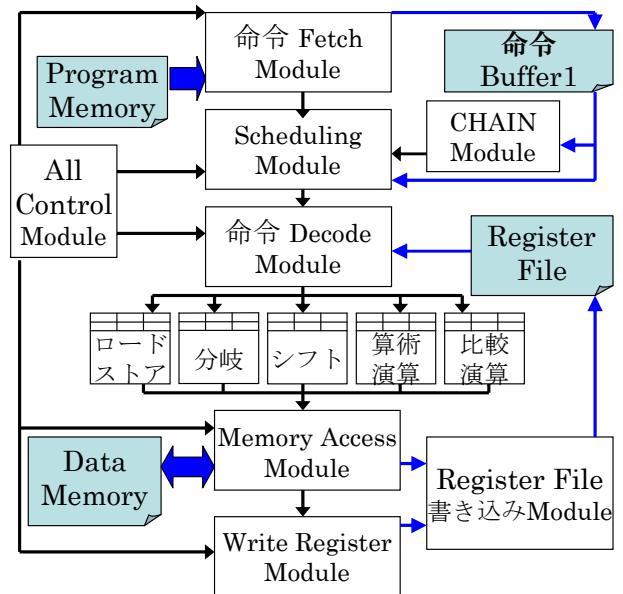


図1 プロセッサブロック図

2.2 先見機構と発行ポリシー

命令発行ポリシーで、プロセッサの先見能力が決まり、そして性能が左右される。先見機能とは、現在実行中の命令より、後続の命令が独立に実行可能かどうかを調べることである。プロセッサは先見機能を利用し、競合している命令が存在しても独立な命令を実行できる。このようなポリシーをより洗練すると、その先見機構が複雑になる。本研究では、CHAIN を用いて、データハザードを解消する。そのため、より単純な先見機構で、より単純なポリシーでの命令発行を行う。本プロセッサでは基本的に最も単純な命令発行ポリシーであるイン・オーダー発行 (in-order-issue) を利用している。

3. CHAIN 手法

3.1 CHAIN 手法の原理

CHAIN 手法は、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。次に、CHAIN を使える命令列を示す。

```
In1 Add $s1 $s4 $s5
In2 Sub $s2 $s1 $s6
```

この例の中で、In1, In2 を同時に実行するとき、二つの命令の間に \$s1 に関するデータ競合が発生する。このとき、CHAIN 手法による制御信号を作り、演算部分に

CHAIN Technique and the Effect in a SuperScalar Processor

[†]Lin Meng, Graduate School of Science and Engineering, Ritsumeikan University

[‡]Junichi Fukuhara, Ritsumeikan University

[§]Shigeru Oyanagi, Ritsumeikan University

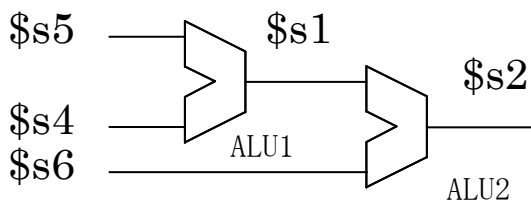


図2 CHAIN手法のイメージ図

渡す。そしてそれらの制御信号によって、図2のCHAIN手法のイメージ図に示すように、まず演算ユニット1を用いてIn1を演算させ、次に生成されたデータをユニット2にバイパスし、In2を実行する。その後、二つの演算が終わってから、結果をレジスタファイルに書き込む。この手法で、In1とIn2のように、データハザードを生じている命令列を並列に実行することができる。

3.2 CHAIN手法の効果

CHAIN手法を用いると、以下の理由により、省電力につながると思われる。

- CHAIN手法では、データハザードを生じている命令列を実行することができる。これによって無駄なストールを削減できる。
- 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って、汎用レジスタのデコードを省略できる。
- これまでストールで解決されたデータハザードがCHAIN手法で解決でき、プロセッサIPCの向上のため、より低いクロック周波数での動作も行える。
- これまで命令レベルの並列化を向上するために、複雑な発行ポリシーが開発されてきたが、CHAINでは、単純な先見機構と単純なポリシーでの発行で、より高い並列化を図ることが出来る。本プロセッサはIPCを下げずに、小さな回路での動作が行える。

4 CHAIN手法の評価

4.1 評価環境

CHAIN手法の効果について、SimpleScalar Tool Setを用いて、その中のsim-outorderを改造し、評価した。改造は、主にsim-outorder中のデータ依存をチェックする部分で、CHAINを使える命令に関するデータ依存のチェックを外すことである。ベンチマークプログラムとしてSPEC95の4本を使用した。CHAIN手法の効果をはかるために、最大実行命令数が2個、4個のそれぞれについてIPCの値を測定、評価した。命令の発行は、SimpleScalar Toolのオプションを利用し、in-order-issueで行った。

4.2 評価結果

図3では、Vortex, go, ijpg, compress95の4本のベンチマークを用いて、CHAIN手法を利用した効果を示している。図の横軸はベンチマークプログラムである。縦軸では各ベンチマーク実行後のIPCを示している。具体的には、CHAINを使用せずに、最大2命令と4命令を同時実行したときのIPC、CHAINを使用するときの最大2命令と4命令を同時実行したときのIPCである。

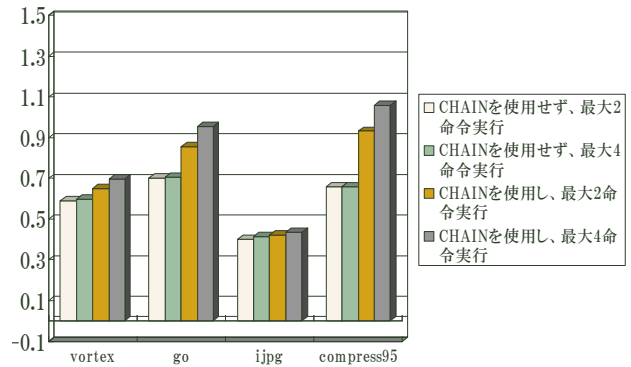


図3 CHAIN手法の効果

この図を見ると、CHAIN手法を用いることにより、IPCを向上させることができると言える。特に、goでは、2命令同時発行したときのIPCが1.223倍に向上し、4命令同時発行したときのIPCが1.35倍に向上した。compress95でも、2命令同時発行したときのIPCが1.42倍に向上し、4命令同時発行したときのIPCが1.6倍に向上した。CHAIN手法を用いると、発行ポリシーと先見機構が単純になり、IPCも向上させることができた。そのため、省電力につながると思われる。性能を十分に追求しない場合でも、IPCの向上によってプロセッサの周波数を下げることができ、それによって電力も節約できると考えられる。

5 まとめ

本研究ではCHAIN手法を提案した。これによって、同時に発行しようとする命令の間にデータハザードが存在するときに、ストールをさせずに、命令を並列に実行できた。また、IPCが向上し、単純な発行ポリシーと単純な先見機構によるハードウェアの簡素化をはかることができる。これは省電力につながると思われる。今後の課題としては、同時に実行できる命令数を考慮し、SimpleScalar Toolをさらに改造することで、IPC精度を向上させることである。

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] David A. Patterson/John L. Hennessy 著 成田光彰 訳：コンピュータの構成と設計（上）（下）、日経BP社，1999
- [4] 五島正裕、他：スーパースカラのための高速な動的命令スケジューリング方式，ARC研究報告 Vol2001 No. 20, 2001
- [5] <http://www.simplescalar.com/>
- [6] 佐藤寿倫：新世代マイクロプロセッサアーキテクチャ（後編）：新しいデザインバランス1省電力とプロセッサ Vol.46 No.11 2005