

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。

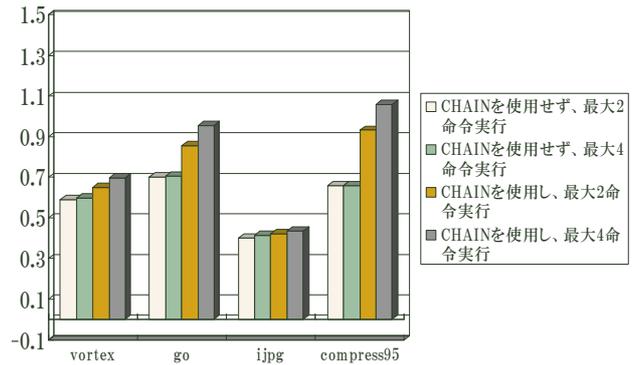


図3 CHAIN手法の効果

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