

スーパースカラ型マイクロプロセッサの命令発行機構の解析

2L-3

山口龍一 山田晃弘 山本崇夫 井上雅夫 三宅二郎 枝松寿一
松下電器産業(株) 半導体研究センター

1. はじめに

スーパースカラ技術では、単位サイクルあたり複数の命令を、マイクロプロセッサ内部の複数の演算器に発行して高性能化を達成する。命令発行の際、以下の依存関係により同時発行が阻害される。

- ・データ依存 後続命令が先行命令の結果を参照
- ・逆依存 先行命令の読込元と後続命令の書込先が同じ
- ・出力依存 先行命令と後続命令の書込先が同じ
- ・制御依存 分岐条件が先行命令の結果に依存

上記依存を解消し、より多くの命令を発行するため、種々の命令発行方式が提案されている[1]。従来の解析[2]では、純粋に命令の並列度を評価することを目的としていたので、命令は理想的に供給されるとしていた。しかしながら、現実のマイクロプロセッサに应用する場合、命令/データ供給能力は分岐命令とキャッシュメモリにより左右されるという問題がある。

我々は、仮想的にマイクロプロセッサモデルを構築し、分岐機構とメモリ階層による性能阻害要因を組み込んで、種々の命令発行方式を実現した場合の性能を解析した。本稿では、仮想マイクロプロセッサでの命令供給能力による命令発行機構の性能変化について報告する。

2. 命令発行機構

以下の命令発行方式を比較した。

【インターロック制御】 発行対象命令が先行命令に対していずれかの依存関係にある場合、当該命令と後続命令の演算器への発行を止める。したがって、命令は in-order に発行される。

【スコアボード】 発行対象命令が先行命令に対してデータ依存あるいは出力依存関係にある場合、当該命令の演算器への発行を止める。一方、後続命令は先行命令に対して依存がない場合、out-of-order に命令発行される。ここでは文献[1]に従った。

【レジスタリネーミング】 発行対象命令が先行命令に対してデータ依存にある場合のみ、当該命令の演算器への発行が止められる。後続命令は out-of-order に命令発行される。

【投機実行】 分岐が確定する前に、分岐予測にしたがって分岐先命令の実行を開始する。予測する分岐数は1で、簡単化のため taken 予測とした。

3. 解析モデル

図1に想定したモデル構成図を示す。命令は、命令フェッチキューによりメモリ階層から読み出され、命令発行部で一旦保持される。メモリから命令発行部へは1サイクル当り最大4命令が送られ、整数命令と浮動小数点命令それぞれ最大16命令が保持される。命令発行部では、命令間の依存関係を調べ、実行可能命令を演算器に発行する。演算器は、整数演算用にALUと、メモリアクセスのアドレス計算用の加算器と、両方の機能を兼ねた演算器の計3個を設定した。浮動小数点演算用には、加算器と乗算器の計2個を設定した。以上の演算器を使用することで、最大5命令が同時に演算器に発行される。命令は、SPARC 命令セット[3]を使用する。

命令発行部は前記4種のアルゴリズムをモデル化した。また状態保存部は、命令間の依存関係の解消と、投機実行での分岐前の状態保存に使用した。

解析には上記構成をモデル化し、SPARC 実行形式のオブジェクトを実行できる専用ソフトウェアシミュレータを開発した。

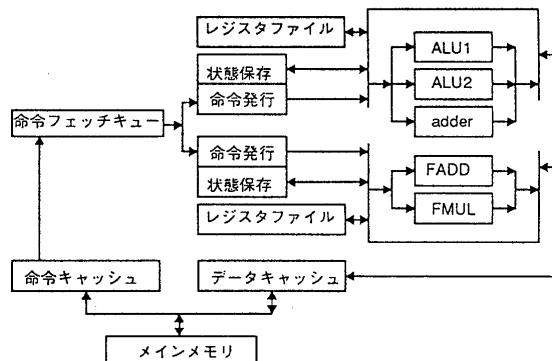


図1 モデル構成図

表1 シミュレーションによる実行命令数及び分岐命令頻度

ベンチマーク	実行命令数	分岐命令頻度
gcc	24614231	0.183
espresso	32571560	0.178
doduc	61348604	0.064
fpppp	10750376	0.015

4. シミュレーション結果

ベンチマークプログラムには、SPEC ベンチマークの一部 (整数演算系: gcc と espresso、浮動小数点系: doduc と fpppp) を使用した。実行命令数と分岐頻度を表 1 に示す。

図2、図3、図4、図5にシミュレーション結果を示す。図2はgccとespresso、図3はdoducとfppppそれぞれを実行した場合に命令発行部で保持される平均命令数を示す。横軸は命令発行方式、縦軸は保持される命令数を示す。図4はgccとespresso、図5はdoducとfppppそれぞれを実行した場合の命令発行機構と単位時間あたりの命令実行数 (IPC) の関係を示す。横軸は命令発行方式、縦軸はIPCを示す。キャッシュサイズは8/16/32/∞ Kbyte と変化させた。

図2と図3から、整数系のプログラムでは浮動小数点系プログラムに比較して、命令発行機構に保持される命令数が極めて少ない。これは表1から分岐命令の出現頻度によると予想される。

図4と図5から、キャッシュが大きくなるにしたがって命令発行機構の違いによる同時に実行できる命令数の差が大きくなる。また、浮動小数点系の方が分岐間隔が長

く、スコアボード/レジスタリネーミングにより命令実行数が最大2倍程度高められる。投機実行では分岐間隔が短い整数系の方が多くの命令を供給され命令実行数が改善されている。

5. おわりに

分岐命令とメモリ階層による性能阻害要因を組み込んで命令発行機構の評価を行なった。シミュレーション結果から、命令/データの供給能力を高めることで命令発行方式の改善による命令実行数の向上が期待できる。しかし、整数系プログラムでは、分岐による命令供給阻害により、命令発行方式の違いによる性能差が小さくなる。

【参考文献】

- [1] S. Weiss et al.: "Instruction Issue Logic in Pipelined Supercomputers", IEEE Trans. Computers, Vol.c33, No.11, p.1013, 1984.
- [2] M. Butler et al.: "Single Instruction Stream Parallelism Is Greater Than Two", 18th Annual Int. symp. on Computer Architecture, p.276, 1991.
- [3] "The SPARC Architecture Manual", Prentice-Hall, 1992.

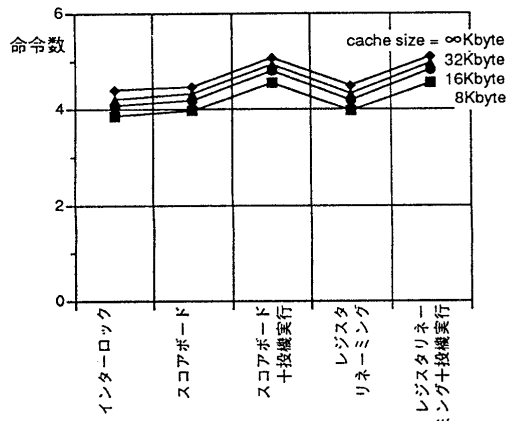


図2 整数演算系(gcc, espresso)での命令発行機構による命令発行部で保持される命令数の変化

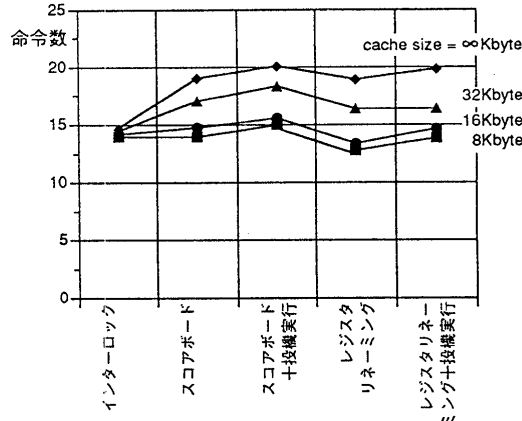


図3 浮動小数点系(doduc, fpppp)での命令発行機構による命令発行部で保持される命令数の変化

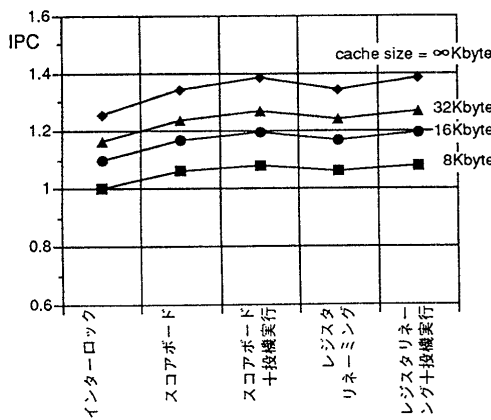


図4 整数演算系(gcc, espresso)での命令発行機構によるIPCの変化

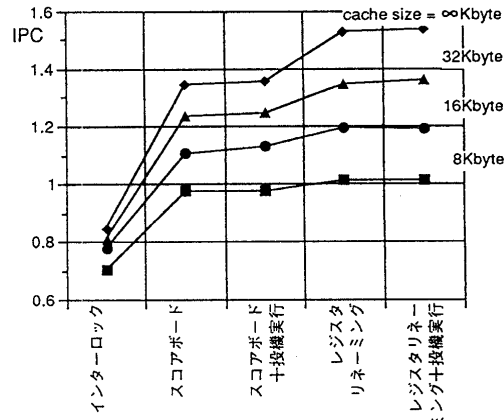


図5 浮動小数点系(doduc, fpppp)での命令発行機構によるIPCの変化