4N-3 SIMP（単一命令流／多重命令パイプライン）方式のシミュレーションによる評価

入江直彦　久我守弘　村上和彰　富田眞治
（九州大学）

1　はじめに

現在我々は，高性能計算機システムの心臓部となりうる超高速プロセッサのアーキテクチャとしてSIMP (Single Instruction stream／Multiple instruction Pipelinig：単一命令流／多重命令パイプライン）方式を提案している．本稿では，SIMP方式の妥当性を検証するために作成したソフトウェアシミュレータについて述べる．

2　シミュレータの目的

SIMP方式とは，命令パイプラインを多重化することにより時間並列処理に加えて空間並列処理を目指すものである．Tステージのパイプラインを S本装備した場合，動的命令数Nのプログラムを（N／S－1＋T）τで実行できる．ここでτはパイプライン・ピッチを表わす．しかし実際は，命令間の依存関係に起因するパイプラインの乱れにより性能低下を招く．特にSIMP方式においては，時間的および空間的という2次元の命令間依存が生じるため，これにどう対処するかが鍵となる．

シミュレータは，こういったパイプラインの乱れについて検証し，SIMP方式実現への妥当性を与えることを目的とする．

3　シミュレータの概要

3.1　要件

本シミュレータは「SIMP方式評価のための汎用シミュレータ」を目指す．そのための要件として，（1）種々の段数・機能を持つパイプラインの評価，（2）パイプラインの多重化による性能向上の測定，（3）各種高速化技法の効果の検討，（4）種々の命令セットとパイプラインとの適応関係の調査，があげられる．

3.2 環境

本シミュレータは，実行環境であるMC68020から得たトレースデータ（動的命令流）によりシミュレータに投入する命令流を生成する．多重命令パイプラインの場合，命令列は命令ブロックの形で供給する．

3.3　構成（モデル）

上記の要件（1）を満たすには任意のステージ構成を許す必要がある．本シミュレータでは，パイプラインの機能を最小単位に分割し，それらの集合によってステージおよびパイプラインを表現する．この最小分割された機能要素をここではサブステージと呼ぶ．シミュレータにおいてパイプラインは，資源－サブステージ－ステージ－パイプラインといった階層構成をとり，各ステージにどのサブステージを用意するかで任意のパイプラインを構成できる．（図1）

3.4　命令セット

上記の要件（4）を満たすため，オペコードテーブルおよびアドレッシングモードテーブルによって命令の機能／粒度やアドレッシングモードの種類／コストを規定する．このテーブルを変えることにより種々の命令セットを表現できる．

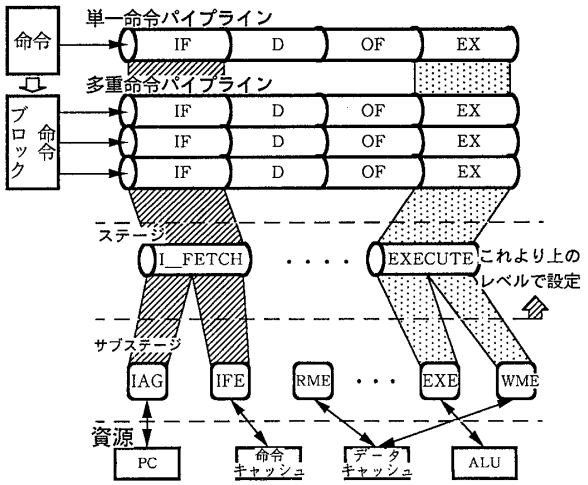

図1 シミュレータの概念図

Evaluation of SIMP (Single Instruction stream／Multiple instruction Pipelinig) through Computer Simulation
Naohiko IRIE, Morihiro KUGA, Kazuaki MURAKAMI, and Shinji TOMITA
Kyusyu University

3.5 依存関係

命令パイプラインの乱れの要因として命令間の依存関係があげられる. 命令間の依存関係にはデータ依存と制御依存があるが, シミュレータにはこれらの依存関係によるパイプラインの乱れを表現することが要求される.

(1) データ依存関係

データ依存関係には①フロー依存②逆依存③出力依存がある. 命令パイプライン処理においてはこの三者とも起こり得るが, ②③については簡単なハードウェアにより削除が可能であるため, ここでは①のみを取り扱う. またSIMP方式において各命令は時間的および空間的という2次元の依存関係を持つため, シミュレータにはこれらの依存関係の表現が要求される. シミュレータでは命令間にデータ依存関係がある場合, それらの命令のリンクをとり先行命令のデータ書き込みが終了するまでオペランドは供給されない.

(2) 制御依存関係

制御依存は条件分岐により生じる. 本シミュレータでは後続命令は次の命令ブロックから投入し, 分岐命令の実行が終了するまで次の命令ブロックはフェッチされない.

3.6 高速化技法

(1) バッファリング

バッファリングは各ステージ間の処理速度の違いを緩衝し, ステージの利用効率を高める手法である. シミュレータにおいては各ステージ間にFIFOのバッファを設けることでこれを実現している. バッファのエントリー数はパラメータとして与えられる.

(2) バイパス機構

命令間にフロー依存が存在する場合, 依存関係のある命令間でできるだけ早くデータの授受を行う必要がある. そのための機構としてバイパス機構がある. シミュレータにおいてフロー依存は予め解析されており, 先行命令の実行完了によってオペランドが供給されるという形でバイパス機構を表現している.

(3) 分岐予測

制御依存に対する対策として分岐予測があげられる. シミュレータでは常に分岐が起こると予測するTaken予測と, 常に分岐しないと予測するNot Taken予測について検討している.

4 シミュレーション結果

対象とするパイプラインのステージ数を3, 6, 9段とし, パイプラインの多重度を1〜6とする. また高速化技法については3.6節で述べた技法を単独で用いた場合および全てを用いた場合について検討した. 命令セットについては今回は固定している.

評価の基準として, 応答時間を考える. これはある動的命令流を処理するのにかかったステージ数にパイプラインピッチを掛けたものである.

テストプログラムは汎用ベンチマークプログラムであるDhrystone (C言語版) を用いた.

図2に3段単一命令パイプラインを基準とした応答時間の向上比を示す. これより, パイプラインを多重化した場合, 4〜6多重で単一の1.7〜2.3倍の性能向上が見られた. この結果は, 単一命令パイプライン用の命令流を想定しているため, コンパイラによる最適化によりさらに性能向上が期待できる.

5 おわりに

詳細なシミュレーション結果はまた別の機会に譲りたい. また現在試作中のSIMP型プロセッサ『新風』のシミュレーションも併せて進めていくつもりである.

参考文献

1) 村上 他 : SIMP (単一命令流／多重命令パイプライン) アーキテクチャについて, 情処第36回全国大会論文集, 3C−1 (1988)

2) 五島 他 : SIMPアーキテクチャに基づくハードウェア・システム構成, 情処第36回論文集, 3C−2 (1988)

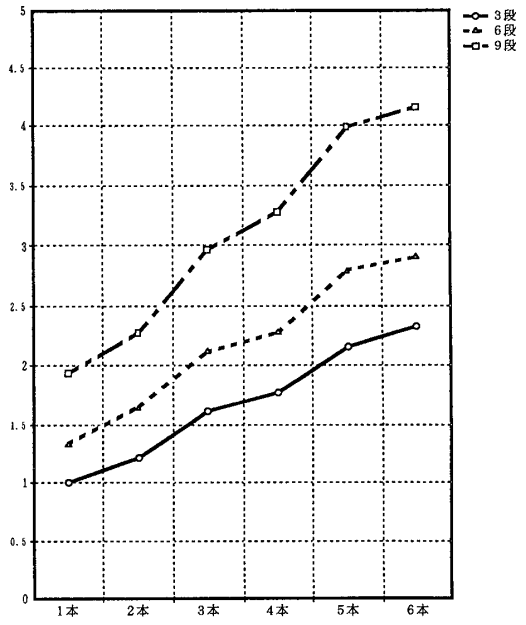

図2 応答時間の向上率