2バイト固定命令長キューマシンアーキテクチャの性能評価

山﨑　淳一[†]　ABDERAZEK　BEN A.[†]　繁田　聡一[†]　吉永　努[†]　曽和　将溶[†]
[†]電気通信大学　大学院情報システム学研究科

1．はじめに

近年，プロセッサの性能向上には目を見張るものがあり，分岐予測や投機実行，並列処理等の技術により，年々複雑になってきている．

我々は，Java などでよく知られているスタック計算方式に代わり，キュー計算方式に注目し，基礎原理から考え直し，いくつかのプロセッサを提案してきた．本稿では，2 バイト固定命令長のキューマシンアーキテクチャの仕様と，ソフトウェアシミュレーション結果について述べる．

2．キューマシンアーキテクチャ

キューマシンアーキテクチャとは，命令の中間結果をキューと呼ばれる FIFO(First In First Out)データ構造に格納するという計算モデル，すなわちキュー計算モデルを使ったプロセッサアーキテクチャである．図 1 の命令実行の様子が示すように，キュー計算モデルでは，キューの先頭からデータを取り出して演算を行い，その結果をキューの末尾に挿入する．基本的にデータの取り出し位置と挿入位置は，キューの先頭と末尾に限定されている．従って，ランダムアクセスのレジスタ計算モデルのような，オペランドのレジスタ指定が必要ないため，命令がコンパクトでプログラムサイズを小さくすることが可能である．また，キュー内部のデータの並び方は前後の命令で使用するデータの位置が干渉しにくいため，データの依存性が少なく，並列処理が容易である特徴も持ち合わせている．

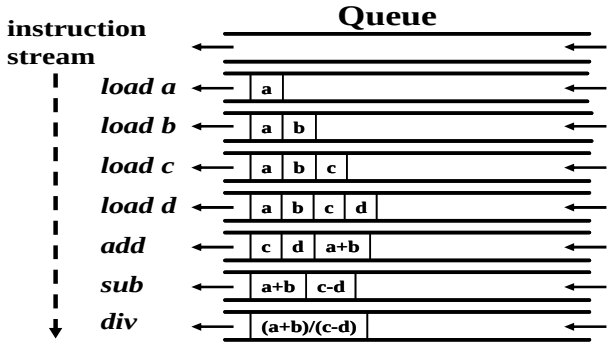

図 1.キュー計算モデルによる命令実行の様子

Paformance Evaluation of a Queue Machine with 2-Byte Fixed-Length Instruction Set Architecture

† Graduate School of Information System University of Electro-Communications.

3．2 バイト固定命令長キューマシンアーキテクチャ

3．1　命令セット

プログラム中の命令の実行やデータアクセスには，時間的に局所性が存在する．同じ値を何度も使用する場合，純粋なキューのみを用いると，その値を使う回数分だけキューへデータを格納しなければならなく，プログラムサイズの増加や処理時間の遅延が問題となっていた．そこで，キューの先頭以外へのアクセスを許し，それらの命令セットを全て 2 バイト固定長で表現した．可変長の命令フォーマットに比べて，2 バイトの固定命令長フォーマットにすることで内部構造をより簡素化する事が可能である．

メモリアドレスを伴う命令（ロード／ストア，分岐命令）では，インデックスレジスタやアドレスレジスタを用いて，それをオペコード内にて指定する事で 32 ビットの指定を実現している．また，オフセットの指定に，オペランドの 8 ビットを用いているが，拡張命令を用いる事で 2 命令により 16 ビットのオフセット幅を可能とした．

3．2　パイプライン構成

図 2 は，2 バイト固定命令長キューマシンアーキテクチャの基本ブロック構成を示している．それらは次の 6 ステージで構成され，パイプライン処理を行う．

（1）フェッチ：毎サイクル規定バイトを命令メモリからフェッチバッファにフェッチする．フェッチした命令に分岐系命令があった場合，分岐予測結果に従い，次クロックのフェッチカウンターに適用する．

（2）デコード：命令のタイプと使用する機能ユニットの種類の判別を行う．メモリアドレスを伴う命令のオフセット拡張命令を発見した場合は，対応する命令に結合させてからデコードを行う．

（3）キュー計算：各命令に対して，データの取り出し位置 QH と挿入位置 QT の計算を行う．また，QH，QT を操作する命令はこのユニット内にて実行し，以降のステージには送らない．

（4）バリア：　分岐系命令やインデックスレ

ジスタやアドレスレジスタの従属関係を調べ，ハザードを検知し，以前のステージの停止や処理の再開を行う．

（5） 発行：計算資源（キューレジスタデータや演算ユニット）から発行可能な命令を実行ユニットへ発行する．特に，アウトオブオーダー発行可能な場合は，バッファ内から発行可能な命令を探して発行する．

（6） 実行：各種演算ユニットに発行された命令を実行する

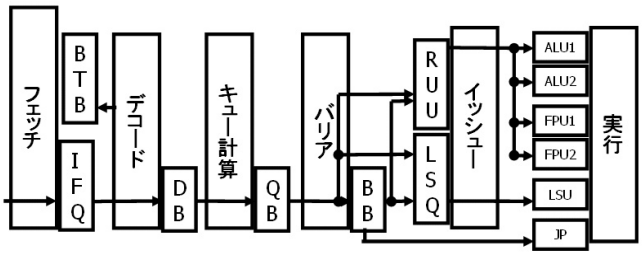

図 2.パイプライン基本ブロック構造

4． シミュレーション

4．1 ソフトウェアシミュレータ Q-sim

2 バイト固定命令長キューマシンアーキテクチャの性能評価にあたり，命令レベルで実行可能なソフトウェアシミュレータ Q-sim を作成した．ソフトウェアシミュレーションの特徴として，アーキテクチャの一部分や全体的な評価が容易である事，フェッチ幅やキューレジスタサイズ等のパラメータを変更して実験する事で，より適した実装環境の模索が容易である事が挙げられる．

Q-sim は，SimpleScalar Tool Set の Sim-Outorder を参考に作成した．これは，後々の評価結果を比較する為でもある．

4．2 評価方法

C 言語で書かれたベンチマークプログラムを本研究室で開発されたコンパイラを用いてコンパイルし、シミュレータと共に作成した簡易的なアセンブラを用いてアセンブルしシミュレーションを実行する。

5． おわりに

現在、作成したシミュレータを用いて動作の検証を行っています。

参考文献

[1]B. A. Abderazek, M. Sarem, M. Sowa, “Dynamic Fast Issue (DFI) Mechanism for Dynamic scheduled Processors”, IEICE Vol.E83-A No.12 pp.2417-2425 (2000)

[2]B. A. Abderazek, 繁田 聡一, K. Nikolova, 吉永 努, 曽和 将容: ”並列キュープロセッサの基本設計”, 電子情報通信学会, 技術研究報告(CPSY), Vol.102 No.478 pp.55-60(2002)

[3]Doug Burger and Todd M. Austin: The SimpleScalar Tool Set,Version 2.0, Technical ReportCS-TR-1997-1342, University of Wisconsin,Madison(1997)

[4]A.Markovskij, M.Sowa, S.Shigeta, and B.A.Abderazek, “Producer-order Parallel Queue Processor Architecture Design”信学技報(CPSY) (2004 年 1 月 発表予定)