同時マルチスレッディング（SMT）技術を用いたマルチスレッド並列キュープロセッサのハードウエア設計

佐々木 博敏† 奥村 義智†☆ Ben A. Abderazek†
繁田 聡一† 吉永 努† 曽和 将容†
†電気通信大学 大学院情報システム学研究科
☆現在，ソニー 株式会社

1 はじめに

現在，既存のプロセッサはランダムアクセスなレジスタを中間格納用レジスタとして用いているが，我々は中間格納用レジスタにキューを用いるキュープロセッサの研究[1) 2) 3)]を行っている．

同時マルチスレッディング（SMT）技術を用いたマルチスレッド並列キュープロセッサ（SMT-PQP）は，並列キュープロセッサ（PQP）をベースにした，前述のキューと呼ばれる FIFO（First In First Out）の記憶領域に計算の中間結果を格納するキュー計算モデルに基づくプロセッサである．本プロセッサは，レジスタリネーミングなどの特殊なハードウエアを用意することなく，スーパースカラープロセッサの持つ命令レベルの並列性（ILP）と，マルチスレッドプロセッサの持つスレッドレベルの並列性（TLP）の両方の並列性が利用可能である．本論文ではこのプロセッサのハードウエア設計について述べる．

2 特徴

2.1 キュー計算モデル

SMT-PQP の基本原理であるキュー計算モデルはキューと呼ばれる FIFO の記憶領域に計算の中間結果を格納する計算モデルである．全ての命令がキューの先頭（QH : Queue Head）からデータを取り出し，キューの末尾（QT : Queue Tail）に計算結果を格納する．この計算モデルはレジスタを明示的に指定する必要 がない．

図 1 に，キュー計算モデルで計算する様子を示す．例として，*a+b* の計算式を計算する．一番上のキューは初期状態の空のキューである．最初の命令（*load a*）でデータ *a* をキューに入れ，次の命令（*load b*）でデータ *b* をキューに入れる．加算命令（*add*）では２つのデータを必要とするのでキューの先頭から *a* と *b* が取り出され，加算が行われ，*a+b* という結果がキューの末尾に挿入される．

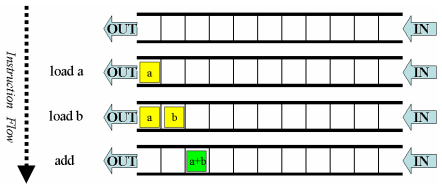

図 1 キュー計算モデル
Fig.1 Queue Calculation Model

例にように，キュー計算モデルはレジスタを明示的に指定する必要がないため，個々の命令長が短くなり，全体としてプログラムサイズが小さくなる特徴を持つ．その上，前後の命令間にデータ依存関係があることが少ないことから，SMT-PQP は，並列実行とアウトオブオーダー実行に適する．

2.2 ILP と TLP を併用するキューレジスタ

SMT-PQP は，SMT[4)]が計算資源を共有するという特徴を持ち，内部資源を物理的に増設することなく，ILP を有効に使えるアーキテクチャと TLP を有効に使えるアーキテクチャを組み合わせることで ILP と TLP の両方の並列性の抽出を可能にする．例えば，TLP が小さく，ILP が大きいプログラムには多くのレジスタリソースを ILP 用に用い，TLP が大きく ILP が小さいプログラムには，TLP 用にレジスタリソースを用いることができる．そのため，ILP，TLP に適応するプロセッサを得ることが実現可能である．

2.3 レジスタリソースの拡張

SMT-PQP は，並列キュープロセッサの持つ中間結果格納用レジスタのキューの共有という特徴に，SMT の持つ計算資源の共有という特徴が加わる．レジスタリソースの拡張により，レジスタにスレッド名を付けなくても，それぞれがプロセッサユニークになりレジスタ名がぶつかることは決してない．そのため，従来のプロセッサのようにレジスタリネーミングを行う必要はなくなり，レジスタリネーミング機構も必要なくなる．レジスタリソースの共有は，アプリケーションプログラムの ILP，TLP 含有特性に，レジスタレベルまで適用可能なことを示している．SMT-PQP は，キュー計算モデルのレジスタの明示的指定が不要であるという特徴を活かした設計を行うことで，ハードウエアが複雑になることはない．

“Hardware Designing of Multithread Parallel Queue Processor using Simultaneous Multithreading (SMT)”
Hirotoshi Sasaki†, Yoshitomo Okumura†☆, Ben A. Abderazek†, Soichi Shigeta†, Tsutomu Yoshinaga† and Masahiro Sowa†
†Graduate School of Information Systems, University of Electro-Communications
☆Presently with Sony Corporation

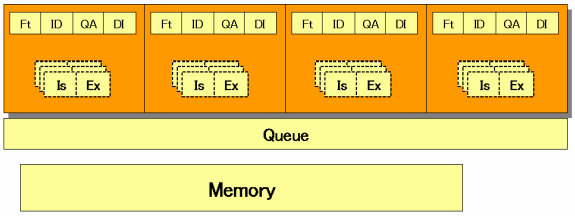

図 2　SMT-PQP アーキテクチャ
Fig. 2　SMT-PQP Architecture

3　ハードウエア設計

3.1　プロセッサ設計

SMT-PQP は，複数のプロセッサ要素（PE：Processor Element）で構成されている．PE は並列に動作し，スレッドを並列に実行する．今回の設計では 4 つの PE により 4 スレッドの並列実行としている．各 PE にはパイプラインを構成する各ステージが割り当てられ，それぞれの PE で 3 命令並列に処理する．そのため，最大 12 命令の並列実行が可能である．

SMT-PQP では，中間格納用の物理レジスタとして，一本のキューだけ用意する．マルチスレッド実行時には論理的に 4 つに区切られ，各 PE に対して独立したキューを与える．シングルスレッド実行時には論理的に区切らず，長い 1 本のキューとしてそのまま用いる．

図 2 に SMT-PQP アーキテクチャを示す．SMT-PQP は，図に示すように各 PE に対し，物理的にそれぞれ設けるステージ（■）と共有するステージ（●）がある．

- ■　フェッチ（Ft）
- ■　命令分割（ID）
- ■　キュー割付（QA）
- ■　命令分配（DI）
- ●　イシュー（Is）
- ●　実行（Ex）

SMT-PQP において，ここで問題となるのが実行ステージに対して十分な命令を供給するかという点である．この問題を解決するために，今回はアウトオブオーダー実行を用いた．アウトオブオーダー実行とは命令を出現順序に関係なく実行するものである．命令は消費するデータの生産が完了されていなければ実行できない．前後の命令の依存関係がない時は，ある命令の消費するデータの生産が未完了で，後続の命令のデータの生産が完了している場合には，後続の実行を先に行う． その後，命令のデータの生産が完了し，実行を行ったとしても支障はない．このとき，キュー内部の位置は互いに独立で，命令間の干渉は起こらない．

3.2　レジスタリソースの有効活用設計

SMT-PQP は，fission 命令（キュー分割命令）を実行し，キューを分割する．キューを PE 毎に区切り，プロセッサをマルチスレッド実行できる状態にする．スレッドは PE 毎に割り当てられたキューを使って並列に実行する．実行後は，シングルスレッド実行ができる状態に戻すために，fusion 命令（キュー結合命令）を行う．fusion 命令により，fission 命令で分割されたキューを再び 1 本のキューにする．このような実行方法であるために，マルチスレッド実行だけでなく，シングルスレッド実行にも対応できる．

また，シングルスレッド実行を行うとき，キューレジスタをスレッド数の乗数分扱うことが可能となる．分割していない状態の 1 本のキューを使用することで，従来よりもレジスタ資源を大きく得ることができる．

3.3　スレッド発行システムの設計

スレッドに発行は，親スレッド（発行するスレッド）が子スレッド（発行されるスレッド）に対し行い，リング状を形成する．今回の設計では 4 スレッドあるので，論理的なスレッド 0，1，2，3，4…に即して，物理的には PE0→PE1→PE2→PE3→PE0 とする．

4　おわりに

本論文では，並列キュープロセッサの高速化のアプローチとして，ILP だけでなく TLP にも着目した SMT-PQP のハードウエア設計を行った．

SMT-PQP は，CMP（Chip Multiprocessing）のようにプロセッサを増やさずとも，1 つのプロセッサ上に複数のスレッドを設けることで TLP を向上することができる．複数のスレッドが同時に実行され，計算資源は共有できるので稼働率の向上が望める．さらに，シングルスレッド実行とマルチスレッド実行の両方に対応しながら，シングルスレッド実行時では分割していない状態の 1 本のキューレジスタが利用可能であるので，従来の SMT プロセッサのようにレジスタリネーミングを行わずとも，レジスタリソースを大きく得ることができる．

今回，SMT-PQP のハードウエア設計において，ハードウエア記述言語である Verilog-HDL により記述し，設計を行った．今後，プログラムに対する処理能力等の性能評価を行う．

参考文献

1)　菊池遊，吉永努，曽和将容: “キュー計算モデルを用いた並列プロセッサの設計” 電子情報通信学会 情報・システムソサイエティ大会, p.42，(2001).

2)　奥村義智，吉永努，曽和将容: “キューマシン用並列化 C コンパイラ”，情報処理学会研究報告 2002-ARC-149，p.127, (2002).

3)　奥村義智: “ マルチスレッド並列キュープロセッサに関する研究 ”，2002 年度電気通信大学修士論文，(2003).

4)　Susan J Eggers, Joel S. Emer, Henry M. Levy, Jack L. Lo, Rebecca L. Stamm, Dean M. Tullsen: “Simultaneous Multithreading: A Platform for Next-Generation Processors”, IEEE Micro, (1997).