2パス限定投機システムのハードウェア設計 ーマルチスレッド制御機構ー

金井 新一† 十鳥 弘泰† 横田 隆史† 大津 金光† 馬場 敬信†
†宇都宮大学大学院工学研究科情報システム科学専攻

1 はじめに

近年、1つのチップ上に複数のプロセッサコアを搭載したマルチコアプロセッサの研究・開発が盛んに行われている。マルチコアプロセッサ上で性能向上を達成するために、複数のプロセッサコアで異なるスレッドの並列して実行を行うことでスレッドレベル並列性を向上させる手法がある。我々の研究室では、スレッドレベル並列性を抽出する手法として、プログラムの実行頻度の高いループ内の実行頻度の高い上位2つの実行経路（パス）を投機実行の対象とした、投機的マルチスレッド実行を行う2パス限定投機方式を提案し、この方式を実現する2パス限定投機システムPALSの構築を行っている[1]。

投機的にマルチスレッド実行する方式ではスレッドの制御を効率よく実現することがシステムの性能に影響を与えるが、ハードウェアで実装を行った場合にスレッドの制御にかかるクロックサイクル数や使用する資源量について検討が行われていない。

本研究では、2パス限定投機システムをハードウェアで実現することを目的として、2パス限定投機システムにおけるスレッド制御部の設計を行う。

2 2パス限定投機方式

2パス限定投機方式は、まずプログラム内の実行頻度の高いループ中で実行頻度の高いパスのプロファイルを行い、その情報を元に実行頻度の高い2つのパス（#1、#2パス）の2つのパスについて、それぞれ分岐命令や不要な命令を削除して最適化を行った投機スレッドコードと、元のコードである非投機スレッドコードを生成しておく。#1、#2パスの投機スレッドコードについてはプログラム中の分岐命令をassert命令に置き換える。この命令により、投機実行に成功したか失敗したかを判断する。投機実行に失敗したスレッド実行部は回復処理を行い、もう一方のパスの投機スレッドコードを実行する。また、後続するスレッドを実行する実行部では回復処理を行った後に新たにスレッドの実行を開始する。

3 2パス限定投機システム

PALSでは、プログラム中の投機実行の対象となるループ部分に対して、#1、#2パスのうちのどちらのパスの投機スレッドコードを実行するかを予測することでスレッドの生成を行う。また、マルチスレッド実行開始するための命令とマルチスレッド実行終了のための命令を備えている。逐次実行時に対象ループの実行を開始したら、パスの予測を行いスレッドコードを割り当てることでスレッドの生成を行う。対象ループの実行が終了したら逐次実行を再開する。2パス限定投機方式を実現するPALSの構成を図1に示す。PALSではスレッドの制御を行うThread Management Unit（TMU）、実行処理部であるThread Unit（TU）、Memory Buffer（MB）、Load Shelter（LS）により構成される。

TMUとTUは、TMUを中心に1対1で接続される。TU、TMU間はスレッド生成や制御のための通知のみを行い、データ等の広い帯域を用いる通信は行わない。また、各TUはMBを1つずつ持ち、MBはLSとバスで接続され、TU同士、MB同士はそれぞれリング状に接続される。スレッドをリングに沿ってTUに連続して割り振ることで、スレッドの生成やスレッド実行終了などの処理を容易にする。

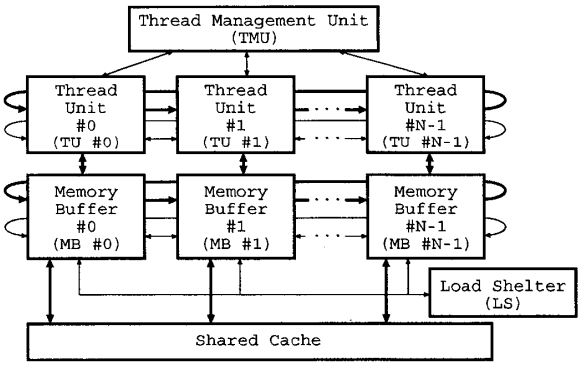

図1: 2パス限定投機システムの構成図

4 スレッド制御部の設計

4.1 スレッド制御部の構成

本研究で設計したスレッド制御部の構成図を図2に示す。各モジュールの機能は以下の通りである。

- Input Port（入力ポート）:
  TUから通知された内容をTMU　Controllerへ送る。TUから同時に通知が送られてきた場合を考慮し、TUからの通知内容を保持するためのバッファを用意する。このバッファによりTUからの通知を順序付けを行い、スレッドに沿って起動したスレッドを処理することで、データの整合性を保つ。
- TMU Controller：
  TMU、TUの状態レジスタと、先頭、末尾スレッドを実行するTUの番号、投機実行に失敗したTUの番号を保持するレジスタを持ち、この値とTUからの通知を基にTMUの動作を制御する。

Hardware Design of Two-Path Limited Speculation System -Multithread Control Unit-
†Shinichi Kanai, Hiroyoshi Jutori, Takashi Yokota, Kanemitsu Ootsu and Takanobu Baba
Department of Information Information Systems Science, Graduate School of Engineering, Utsunomiya University (†)

- 2-Level Path Predictor（パス予測器）:
  2 レベル分岐予測 [?] と類似した構造を持ち、パスの実行履歴、予測履歴を保持するレジスタと実行頻度を格納するカウンタテーブルから成る。

  履歴を保持するレジスタをカウンタテーブルのインデックスとして、パスの予測、実行結果の更新を行う。投機実行に失敗した場合には、予測履歴を保持するレジスタを予測に失敗する前の状態まで巻き戻してスレッドの生成を再開する。
- Output Port（出力ポート）:
  パス予測器からのパスの予測結果と TMU、TU の状態を基に TU へスレッドの割り振りを行う。また、投機実行に失敗した場合に回復処理を行う TU を決定して回復処理の許可を TU へ通知する。

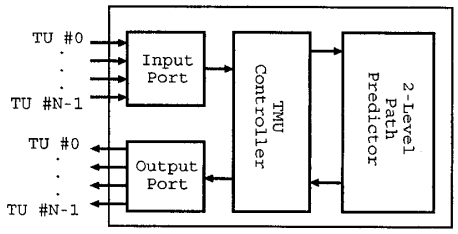

図 2: スレッド制御部の構成

4.2 スレッド制御部の動作

スレッド制御部で行う動作と動作に必要なクロックは以下の通りである。後述する処理に関して、投機実行失敗時に回復処理を行う必要のある TU を選択する際に、ハードウェア記述言語で条件付きの繰り返し処理を行なうと、この処理に数クロックかかってしまう場合がある。そこで、繰り返し処理を行なう必要がある部分に関してはマスクを生成して処理が必要な部分を判断することで、処理にかかるクロック数を軽減する。

- マルチスレッド実行開始
  TU からマルチスレッド実行開始の通知を受け取り、パスの予測を開始する。TU に空きがなくなるまでパスの予測を行い、スレッドの生成を行う。通知を受け取ってからパスの生成までは 4 クロック必要である。
- マルチスレッド実行終了
  TU からマルチスレッド実行終了の通知を受け取り、次のマルチスレッド実行のために初期化を行い、シングルスレッド実行を行う。通知を受けてから 3 クロック後にスレッド制御部の初期化が完了しマルチスレッド実行開始まで逐次実行を行う。
- スレッド実行終了
  TU からのスレッド実行終了の通知と実行履歴を元にパスの実行履歴の更新を行う。実行履歴の更新は通知を受けて 2 クロック後に行いその 3 クロック後に新たにスレッドの割り振りを行う。
- 投機実行失敗
  投機実行失敗の通知を TU から受け取った場合、TMU の回復処理を行い、回復処理が必要な TU へ回復処理の許可を通知する。同時に複数の TU からこの通知を受け取った場合には、先頭に近い TU からの通知を実行し、他の TU からの通知は無視する。また、投機実行失敗に失敗した TU が末尾スレッドを実行する TU であった場合には、スレッド制御部のみ回復処理を行い、マルチスレッド実行モードへと遷移する。通知を受け取ってから回復処理の許可を通知するまでには、最大で 4 クロックが必要である。
- 回復処理
  回復処理中は、回復処理の許可を通知した TU からの回復処理終了の通知を待つ。すべての TU で回復処理が終了したら、2 クロック後にマルチスレッド実行を再開し、スレッドの生成を再開する。回復処理中に他の TU から投機実行失敗の通知が送られてきた場合、投機実行に失敗した TU よりも先頭スレッドを実行する TU に近ければ、入力ポート内のバッファにこの通知内容を保持しておき回復処理後に再び回復処理を行う。また、この TU よりも後続する TU からの通知であれば無視する。すべての TU で回復処理が終了してからスレッドが新たに割り振られるまでは 5 クロック必要である。

5 動作検証

設計したスレッド制御部は波形シミュレーションおよび、アサーション検証を行い正常に動作していることを確認した。

5.1 論理合成結果

設計を行ったスレッド制御部に対して、ターゲットデバイスを ViretexII XC2V6000 FPGA とした場合の論理合成結果を表 1 に示す。表中の括弧はマルチ VLIW プロセッサの試作評価基盤として我々の研究室で構築した PISA ベース VLIW プロセッサ（PVP）のコアで使用するハードウェア資源との比較結果である。

表 1: スレッド制御部の論理合成結果

資源	スレッド制御部	PVP コア	資源量
F/F	45（1%）	3680	67584
LUT	113（1%）	15109	67584
Slice	62（1%）	8506	33792

6 おわりに

本稿では、2 パス限定投機方式におけるスレッド制御部のハードウェア実装を行い、シミュレーションにより動作を確認した。動作に掛るクロックはすべての動作において 5 クロック以内であることを設計結果から示し、論理合成結果から低コストでスレッド制御部が実現可能であることがわかった。

謝辞

本研究は，一部日本学術振興会科学研究費補助金 (基盤研究 (C)20500047，同 (C)21500049，同 (C)21500050) および宇都宮大学重点推進研究プロジェクトの援助による．

参考文献

[1] 十鳥 弘泰 ほか: “2 パス限定投機方式を実現するマルチコアプロセッサ PALS の提案”，信学技報，Vol.109, No.319, pp.19-24, (CPSY2009-46), 2009 年 12 月 3 日.