

## 2パス限定投機システムのハードウェア設計 — メモリアクセス機構 —

北直樹<sup>†</sup> 十鳥弘泰<sup>††</sup> 横田隆史<sup>††</sup> 大津金光<sup>††</sup> 馬場敬信<sup>††</sup>

<sup>†</sup>宇都宮大学工学部情報工学科 <sup>††</sup>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

近年の計算機開発で行われている、プログラム実行の高速化の方法として、プログラムを分割して並列処理を行う方法が挙げられる。しかし並列処理を行う際、プログラム中の依存関係によって、不要な待ち時間が発生する場合があった。そこで、依存関係による待ち時間を解消する方法の1つとして、並列実行される処理の内容を予測し、投機的に実行する方法が考えられた。

当研究室では、プログラム中のループイテレーションに含まれる実行経路(パス)のうち、実行頻度の高いパス(ホットパス)について、上位2つのホットパスが実行される経路がループ処理の大半を占めることに着目し、これら2つのパスに限定して投機的なマルチスレッド実行を行う、2パス限定投機方式が提案され、2パス限定投機方式による計算を実現する計算機システムとして、2パス限定投機システムが設計された。

本稿では、これまでに方式設計された2パス限定投機システムにおけるメモリシステムを実際にハードウェアとして設計することで、方式設計時に分からなかった問題を発見・解決する。

### 2 2パス限定投機システム PALS

2パス限定投機方式を実現するアーキテクチャ PALS の構成を図1に示す。

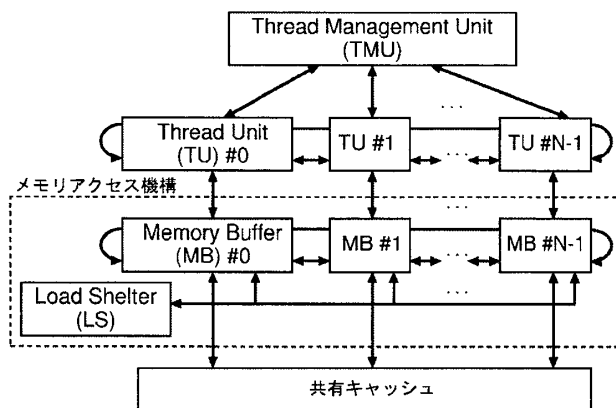


図1: PALSのハードウェア構成

本システムは以下の3つの機構から構成される。

- Thread Management Unit (TMU): 全体のスレッド制御を行う
- Thread Unit (TU): スレッドを実行する
- Memory Access Unit: 投機的メモリアクセスを処理する

TMUは#1パス、#2パスの予測を行い、その情報をTUに送ることでTUの動作を制御する。TUは演算を行う機構であり、TMUから送られたパス情報を元に投機的な計算を行う。並列計算中のメモリアクセスは共有キャッシュではなくメモリアクセス機構に対して行われる。メモリアクセス機構は、図1中の点線で囲まれた機構であり、TUのローカルメモリであるMemory Buffer (MB)、およびMBの空きエントリが無くなったときの退避場所であるLoad Shelter (LS)によって構成される。TUおよびMBは、スレッド間での通信を可能とするために、それぞれ隣接するTU、MBと接続され、リング構造を形成する。また、全てのMBは、LSおよび共有キャッシュに接続される。

### 3 メモリアクセス機構

本研究で設計するメモリアクセス機構は、PALSが行うメモリアクセスを実現するための機構である。

まず、PALSで行われる投機的マルチスレッドはパスの予測が失敗した際に、誤ったパス予測によって行われた処理を全て無効化するため、スレッドが実行される前の状態に戻す回復処理が必要となる。このとき、TUが行ったメモリアクセスを全て無効化する必要があるが、TUの行う投機的な計算の結果が、共有キャッシュに格納されるようにすると、無効化を行う際、共有キャッシュに格納されたデータを全て調べ、予測に失敗したスレッドが格納したデータのみを探して無効化しなければならないため、予測失敗時に行う回復処理が複雑になり、時間がかかる。このためPALSでは、回復処理を簡単にするため、各TUのローカルメモリとしてMBを用意する。プログラムの並列実行時、スレッドの投機実行が終了して結果が確定するまで、TUが行うメモリアクセスの内容を全てMBに格納することで、予測失敗時に回復処理を行う際は、MBの内容を全て無効化するだけで済むようにする。

しかし、MBに全てのメモリアクセスを記録する場合、MBの容量を超えるメモリアクセスが行われて投機的なストアデータが格納できなくなった際、MBのデータを共有キャッシュに退避させることができない。このため、メモリバッファの容量を超えてメモリアクセスが行われた場合、MB内のロード要求の記録をLS

Hardware Design of Two-Path Limited Speculation System -Memory Access Unit-

<sup>†</sup>Naoki Kita, <sup>††</sup>Hiroyoshi Jutori, <sup>††</sup>Takashi Yokota,

<sup>††</sup>Kanemitsu Ootsu and <sup>††</sup>Takanobu Baba

Department of Information Science, Faculty of Engineering, Utsunomiya University (<sup>†</sup>)

Department of Information System Science, Graduate School of Engineering, Utsunomiya University (<sup>††</sup>)

に退避させることで、MB 内に空きエントリを作り、新たなストアデータを格納することにする。

この他、マルチスレッド実行ではスレッド間での通信が何度も行われるため、隣接する MB 間で、投機的にストアされたデータの要求と送信が行えるようにすることで、高速なスレッド間通信を可能にする。また、投機的マルチスレッド実行の特徴として、前のスレッドが扱うデータを確定前に読み込むため、メモリ依存違反が発生する可能性があり、MB でメモリ依存違反を検出し、後続スレッドで行われた計算を全て破棄する機能が必要になる。

#### 4 ハードウェア設計

##### 4.1 MB の設計

設計した MB の内部モジュール構成を図 2 に示す。

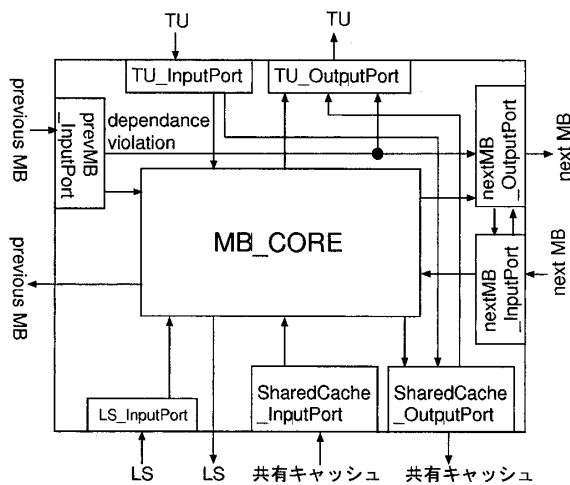


図 2: MB の内部モジュール構成

TU からのメモリアクセスを記憶するため、MB 内にメモリアクセスの種類、アドレス、データ記憶する機構が必要となるので、この機構を MB\_CORE モジュールとして設計する。また、PALS の他のハードウェア機構との通信は、TU が行うメモリアクセスを格納する処理の過程で行われるため、他のハードウェア機構に送信する信号の生成は全てこのモジュールで行うことにする。

MB は 5 つのハードウェア要素と通信を行うため、複数のハードウェア要素から同時に信号が入力された場合、またメモリアクセスの実行中に信号が入力された場合に備えて、MB に入力された信号を格納する入力ポートが必要となる。TU、先行 MB、後続 MB、LS、共有キャッシュから入力された信号を格納する FIFO を持つ入力ポートとして、TU\_InputPort、prevMB\_InputPort、nextMB\_InputPort、LS\_InputPort、SharedCache\_InputPort を用意し、MB\_CORE によって実行されるまで、MB に対する要求を格納する。

また、MB から出力される信号を制御するため、出力ポートを用意する。並列実行を行わない時、TU は MB にメモリアクセスの内容を格納する必要が無い

ため、共有キャッシュに直接メモリアクセスを行う。よって逐次実行時は TU から共有キャッシュ、共有キャッシュから TU に直接信号を送るため、逐次実行時と並列実行時で、TU および共有キャッシュへの信号の経路を変える必要があり、よって、信号を伝達する経路を逐次実行時と並列実行時で変更する出力ポート TU\_OutputPort と SharedCache\_OutputPort が必要となる。また、nextMB\_InputPort モジュールに格納されている後続 MB からの要求が実行される前に、TU の実行が終了する可能性があり、この場合後続 MB からのロード要求が正しく実行されず、後続スレッドの実行に必要なデータが送られない可能性がある。この問題を回避するため、スレッドの実行終了後に格納されている要求を後続 MB へと送り返す動作が必要となるため、nextMB\_InputPort と通信可能な nextMB\_OutputPort モジュールを用意し、後続 MB に送信する信号を選択する。

##### 4.2 LS の設計

LS の内部モジュール構成を図 3 に示す。

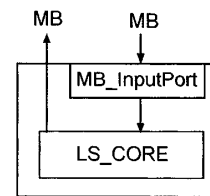


図 3: MB の内部モジュール構成

LS の記憶領域へのアクセスを行う機構を LS\_CORE として設計し、MB から退避したロード要求の格納、また MB での処理に必要なロード要求の情報の送信を行う。また、LS では内部の処理に複数クロックを要するため、入力ポート MB\_InputPort を用意して、MB から入力された信号を、LS 内の処理が終わるまで記憶する必要がある。

#### 5 おわりに

プログラムの実行経路を予測して計算を行う 2 パス限定投機システムと、投機的なメモリアクセスを実現するメモリアクセス機構について述べ、MB の設計を行った。今後は LS の設計を行い、メモリアクセス機構全体の動作検証を行う予定である。

##### 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, 大津 金光, 横田 隆史, 馬場 敬信, “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。