# 2パス限定投機システムの提案 – メモリアクセス機構 –

十鳥 弘泰† 福田 明宏† 佐藤 和史†† 米田 淳一†† 大津 金光†† 横田 隆史†† 馬場 敬信††
†宇都宮大学工学部情報工学科 ††宇都宮大学大学院工学研究科情報工学専攻

## 1 はじめに

近年の半導体集積技術の向上により，プロセッサコアやキャッシュといった豊富なハードウェア資源を1個のチップ上に搭載したマルチコアプロセッサの普及が進んでいる．このマルチコアプロセッサの性能を最大限引き出すためには，実行するプログラムコードの並列化 (マルチスレッド化) が必要不可欠である．

マルチスレッド化を効果的に行うために，我々はプログラム中に存在するループの内，実行頻度が高い経路 (以下，ホットパス) を抽出し，上位2つのホットパスに特化したコードを投機的にマルチスレッド実行する**2パス限定投機方式**を提案し，トレースベースのシミュレーションにより方式の有効性を示した [1]．これを背景に，我々は2パス限定投機方式を実現するシステムアーキテクチャ (**2パス限定投機システム**) の設計を行い，同システムをシミュレータとして構築することで，システムの妥当性の検証および，2パス限定投機方式のクロックレベルでの詳細な評価を行うことが必要であると考えた．そこで本稿では，2パス限定投機システムにおいて投機的なロード/ストアを実現する記憶装置群 (以下，メモリアクセス機構) について設計を行う．

## 2 2パス限定投機方式

多くのプログラムでは，ループイテレーション中に複数の条件分岐を含み，実行される可能性のあるパスは条件分岐の数に応じて指数関数的に増大する．しかし，実際に実行されるパスには偏りがあり，多くの場合ごく少数のパスが支配的であった．2パス限定投機方式では，実行頻度の高い上位2つのホットパス (以下，実行頻度の高い順に**#1パス**，**#2パス**と呼ぶ) を投機の対象とすることで，効果的なマルチスレッド実行を実現する．#1パスおよび#2パスに対して，それぞれのパスに特化したコード (以下，投機スレッドコード) を生成し，ループの1イテレーション毎にどちらのパスが実行されるか予測し，投機スレッドコードを並列実行させることで高速化を達成する．

## 3 2パス限定投機システム PALS

### 3.1 構成

2パス限定投機システム (以下，本システムをPALSと呼ぶ) は，以下の3つの機構により構成される．

- マルチスレッド制御機構: Thread Management Unit (TMU)
- スレッド実行機構: Thread Unit (TU)
- メモリアクセス機構: Memory Access Unit

TMUはパス予測およびスレッド制御を全てのTUに対して行う．TUはプロセッサコアに相当する機構であり，TMUから受け取ったパスの予測結果をもとに投機スレッドコードの実行を行う．TMUが複数のTUに対して順次スレッドを割り当てることにより，マルチスレッド実行を実現する．メモリアクセス機構は，TUのローカルメモリである Memory Buffer (MB) と，MBの補助的役割を担う Load Shelter (LS) により構成される．TUおよびMBはそれぞれ隣接するTU，MBと接続され，リング構造となる．また，投機スレッドコードはスレッドコード生成処理系が生成する．

### 3.2 メモリアクセス機構の設計方針

PALSでは，パスの予測が失敗した場合や，同一アドレスに対するデータの依存違反が発生した場合に，当該スレッドおよびその後続のスレッドの実行を破棄し，レジスタやメモリの内容をスレッド実行前の状態へ戻す必要がある．TUの記憶装置として共有メモリを利用した場合，データの回復処理が複雑化し，大きなオーバヘッドとなることが考えられる．そこでPALSでは，各TUのローカルメモリとしてMBを用意し，スレッドの実行が確定した際に，MBの内容を共有キャッシュへ書き込む．スレッドの実行を破棄する場合は，当該TUのMBのエントリを全て無効にすればよいため，回復処理にかかるオーバヘッドを低減できる．

またPALSでは，ループの1イテレーションを投機スレッドの対象としているため，多数のスレッドが起動されることとなり，スレッド間での通信回数も増大する．TU間で実現するレジスタ間通信は，1個のレジスタにつき一度の送受信しか行えないため，大量のデータの通信には不向きであった．そこで，メモリアクセス機構では隣接するMB間での通信 (以下，MB間通信) を実現し，大量のデータの送受信を可能にすることにより，スレッド間通信の高速化を図る．

## 4 メモリアクセス機構

### 4.1 構成

図1にメモリアクセス機構周辺の構成を示す．MBはTUのローカルメモリとして使用され，全てのメモリアクセスはMBを介して行われる．LSおよび共有キャッシュは全てのMBと接続されるが，通信を行うことができるのは先頭スレッドのMBのみである．

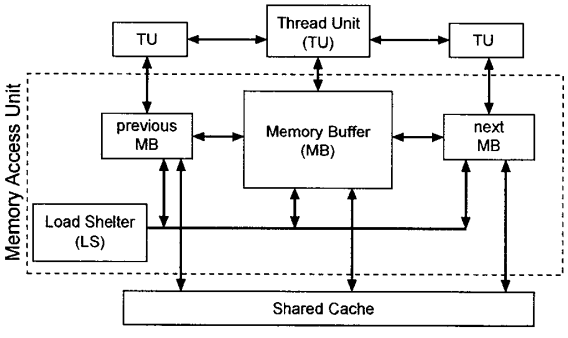


図 1: メモリアクセス機構周辺の構成図

Proposal of Two-Path Limited Speculation System – Memory Access Unit –
†Hiroyoshi Jutori and Akihiro Fukuda
††Kazufumi Sato, Junichi Yoneda, Kanemitsu Ootsu, Takeshi Yokota and Takanobu Baba
Department of Information Science, Faculty of Engineering, Utsunomiya University (†)
Department of Information Science, Graduate School of Engineering, Utsunomiya University (††)

### 4.2 MB 間通信

MB 間通信では，隣接する MB 間でのロード/ストアを実現する．値の受信を行う MB は，先行スレッドの MB に対してロード要求を送り，要求を受け取った MB は，そのアドレスのデータを保持していなければ，更に先行するスレッドの MB に同アドレスのロード要求を送る．また，先頭スレッドの MB は共有キャッシュへロード要求を送る．ロード要求を送る際に，MB は元々の要求が自 TU または後続の MB のどちらから送られたかを示すエントリを作成する．そして，TU からのストアが行われた場合や，先行スレッドの MB または共有キャッシュからのデータ返答があった際に，MB はそのデータのアドレスに対するロード要求のエントリの有無を調べる．エントリが存在した場合は，記憶していた情報に従いデータを送信する．

#### メモリ依存違反

MB 間通信では同期制御を行わず，ARB[2] と同様の方法でメモリ依存違反を検出することで，データの整合性を保つ．MB は後続スレッドの MB から読み込まれたアドレスを記憶し，当該アドレスへの新たなストアが行われた場合に後続スレッドを破棄する．

図 2 にメモリ依存違反を検出した際の制御の様子を示す．先頭スレッドはアドレス X のデータを共有キャッシュからロードし，後続スレッドはアドレス X に対するロード要求を先行スレッドへ送る．その後，先頭スレッドがアドレス X に対してストアを行った場合，メモリ依存違反となるので後続スレッドを破棄する．後続スレッドは回復処理を行った後，それまで実行していたパスを再度実行する．また，メモリ依存違反を検出したスレッドは実行を継続する．

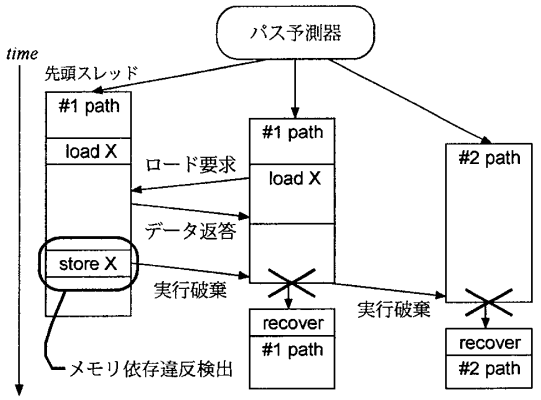


図 2: メモリ依存違反の制御

### 4.3 制御ビット

MB のエントリを図 3 (a) に示す．エントリに付加する制御ビットは以下の通りである．

- valid bit: エントリが有効であることを示す．
- speculative read bit: エントリが後続の MB から読み出されたことを示す．
- speculative write bit: エントリが自 TU から書きこまれたことを示す．
- self query bit: 自 TU のロードによって先行スレッドの MB または共有キャッシュに問い合わせ中のエントリであることを示す．
- other query bit: 後続スレッドの MB の要求によって先行スレッドの MB または共有キャッシュに問い合わせ中のエントリであることを示す．

ロード要求に対して返答されたデータをどこに送信するかは，self query bit および other query bit により判断する．また，メモリ依存違反の検出は，speculative read bit により行う．当該ビットが真であるエントリに対してストアが行われた場合，後続スレッドの実行を破棄する．スレッドの実行が確定した際には，valid bit および speculative write bit が真であるエントリを共有キャッシュに書き込む．


(a) MBのエントリ　(b) LSのエントリ

図 3: エントリの構成

### 4.4 エントリの枯渇問題

MB は後続の MB から要求されたアドレスを記憶しておく必要があるが，そのエントリによって MB が枯渇するという問題があった．特に，先頭の MB が自 TU からのストアデータを格納できなくなった場合，プログラムの実行が停止してしまう．この問題に対して，後続の MB により作成されたエントリは先頭スレッドの実行には必要ないということから，これらのエントリの待避領域を用意することで，複雑な制御を行わずに実行停止を回避できると考えた．そこで PALS では，後続の MB からのロード要求のエントリを格納する記憶装置として LS を用意する．LS のエントリはアドレスと制御ビットにより構成される (図 3 (b))．LS は，制御ビットとして speculative write bit を持たない．

先頭スレッドの MB のエントリが全て消費された場合，先頭の MB は後続の MB から送られたロード要求のアドレスを LS に格納した後，共有キャッシュへ要求を送る．共有キャッシュから返されたデータのアドレスを格納したエントリは MB 内に存在しないため，MB は LS にアドレスを問い合わせる．そして，LS は問い合わせのあったアドレスのエントリの制御ビットの内容から，MB へデータの送信先を伝える．また，LS において依存違反が発生した場合，先頭スレッド以外の全てのスレッドを破棄する．

## 5 おわりに

投機的マルチスレッド実行を行う 2 パス限定投機システムと，システムにおいて投機的ロード/ストアを実現するメモリアクセス機構について述べた．現在は，PALS のシミュレータの動作テストを行っており，メモリアクセス機構の単体テストでは正常な動作が行われることを確認した．今後は，シミュレータ全体のテストを行い，実際のプログラムによる性能評価を行う予定である．

#### 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 (B)18300014，同 (C)19500037，同 (C)20500047）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] 横田隆史, 斎藤盛幸, 大津金光, 古川文人, 馬場敬信, “2 パス限定投機方式の提案”, 情報処理学会論文誌:コンピューティングシステム, Vol.46, No.SIG 16(ACS-12), pp.1-13, 2005.

[2] M. Franklin, G. S. Sohi, “ARB: A Hardware Mechanism for Dynamic Memory Disambiguation”, IEEE Transactions on Computers, Vol.45, No.5, pp.552-571, 1996.