6G－5　細粒度並列計算機お茶の水1号 — メモリベースのデータ駆動同期機構の実現 —

戸塚 米太郎　松本 尚　平木 敬

東京大学理学部情報科学科

1　はじめに

細粒度並列処理ではプロセッサ間のデータ通信が頻繁に起こりうるため、データ通信とそれに伴う同期のオーバヘッドを小さく抑えることが最も重要な課題の一つである。従来、生産者消費者型の同期のために利用されてきた同期機構としてはHEPのfull/emptyビット[2]やデータ駆動計算機等で用いられてきたI-structureメモリ[1]があげられる。フォンノイマン型の要素プロセッサからなる並列計算機に対してはこれまでこのような機構は用いられてこなかったが、スヌープキャッシュ機構とfull/emptyビットによる同期機構を組み合わせた機構[4]が提案されており、その性能が期待される。このような機構を用いると同期がデータ駆動的に行なわれるため、データ通信と同期を統合的に処理でき、同期のための特別な手段を必要としなので、効率の良い処理が可能である。

お茶の水1号にはデータ駆動的な同期機構として、メインメモリのワード毎にデータのfull/emptyを示す同期ビットが付加されており、またメインメモリ上にFIFOを構成する機構が搭載されている。お茶の水1号は各プロセッサとバスの間に外部エージェント、メモリとバスの間にメモリコントローラが存在しており、これらが通常のバスアクセスのほか、上記同期機構の実現をしている。

本発表ではお茶の水1号における同期ビットによる生産者消費者型同期とメモリ上のFIFOを実現する機構の実装方式および同期性能（予定）を示す。

2　同期ビット

お茶の水1号のメモリはワード毎に同期ビットが付加されている。同期ビットは基本的に対応するワードのデータの存在を示すものであり、EMPTYあるいはFULLの値をもつ。現状では同期ビットはデータの存在を示す1ビットであるが、同期ビットの拡張により高機能メモリ[3]が可能になる。同期ビットをキャッシュ上にも付加し、同期ビットの処理をキャッシュ上で行なうようにすれば、スヌープキャッシュプロトコルとの組合せで効率の良い同期が実現できる[4]が、お茶の水1号のように既存の内蔵キャッシュをもつプロセッサを利用したマルチプロセッサではキャッシュ上での同期ビットの管理には困難な点が多い。そこで、お茶の水1号ではメインメモリのみに同期ビットが付加されており、キャッシュ上で同期を行なうような機構にはなっていない。このため本機構はキャッシュされない領域で利用可能であり、メモリコントローラと外部エージェントによりメインメモリを介した形で同期、データ通信が実現される。

本機構の動作を以下に述べる。

- リードリクエスト時
  1. プロセッサはリードリクエストを発行すると、データを受け取るまでブロックされる。
  2. そのプロセッサの外部エージェントはリクエストのアドレスを保持しつつ、メモリにリクエストを送り、バスを解放する。
  3. メモリコントローラは該当するワードの同期ビットを調べ、FULL状態であればデータのレスポンスを行なうが、EMPTY状態であれば何もしない。
  4. リードリクエストを発行したプロセッサの外部エージェントはメモリからのレスポンスデータが到着した時はプロセッサに送る。あるいは別のプロセッサからターゲットアドレスへのライトリクエストを保持しているアドレスとの比較によって検出し、そのライトリクエスト中のデータをバスから直接取り込み、プロセッサに送る。
- ライトリクエスト時
  プロセッサからライトリクエストが発行されるとメモリコントローラはデータのメモリへのライトとともに対応する同期ビットの値をFULL状態に書き換える。このとき上記のとおりに、先に同じアドレスへのリードリクエストによりブロックしているプロセッサがあれば、そのプロセッサの外部エージェントはデータをバスから取り込みプロセッサにデータを送る。

同期の一例を図1に示す。プロセッサBとプロセッサCはプロセッサAがアドレスXにデータを書く前にリード要求をし、ブロックされている。プロセッサBとプロセッサCの外部エージェントはバスを見張っており、プロセッサAがアドレスXにデータの書き込みをすると、そのデータをバスから取り込みプロセッサに送り、ブロックしていたプロセッサBおよびプロセッサCを再起動させる。

同期ビットの内容FULL/EMPTYは実際0/1のどちらかに対応することになるが、その対応を固定してはいない。アドレスの一部に同期極性ビット[4]をもたせ、極性ビットの値によってFULL/EMPTYと0/1の対応を決めている。これにより、データ通信に使い終った後のメモリの再使用が低コストに実現できる。また、アクセス手段として通常の同期とは無関係のメモリアクセスと区別する必要があるため、これもアドレスの一部のビットを使って区別している。

3　メモリ上のFIFO

お茶の水1号のメモリ上のもう一つの機能としてFIFOを構成する機能がある。FIFOは現在のところ全部で8本用意し

Memory-Based Data-Driven Synchronization Mechanism of a General Purpose Fine-Grained Parallel Processor Ochanomiz-1

Yonetaro Totsuka, Takashi Matsumoto, Kei Hiraki

Department of Information Science, Faculty of Science, the University of Tokyo

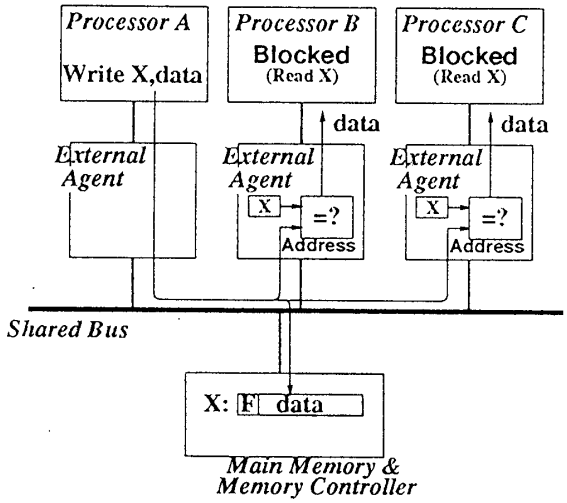

図 1. 同期ビットを使用した同期の例

てある。このFIFOはメモリアドレス（の一部）を識別子として指定される。プロセッサからのアクセスは通常のメモリアクセスと同様でよく、アクセスするアドレスは特定のFIFOの識別子であればよい。FIFOの管理はメモリコントローラによってプロセッサとは独立に行なわれ、プロセッサによるFIFOのポインタの管理は不要である。実際、メモリコントローラは各FIFOに対するポインタを持っており、FIFOの管理をリングバッファによる方式で行なっている。

プロセッサからFIFOへのリードリクエストが発行されると、メモリコントローラはFIFOにデータがあればデータのレスポンスを行ない、データがなければレスポンスはしない。このときリードリクエストを発行したプロセッサの外部エージェントはメモリからのレスポンスデータが到着するか、他のプロセッサから同じアドレスに対するライト（すなわち同じFIFOへのライト）が発行されたかの検出を行なう。外部エージェントはメモリからデータが返送されたときはそのデータをプロセッサに送るが、ライトの検出をしたときは再度FIFOへのリード要求を発行する。プロセッサからFIFOへのデータの書き込みがあると、メモリコントローラはデータを該当するFIFOに書き込むが、もしFIFOがデータで満たされていた時はそのライトをリトライさせるようにしている。

4 同期性能

ここではお茶の水1号の同期ビットを使用した生産者消費者同期の性能の見積もりをソフトウェアによる同期変数を用いた同期との比較によっておこなう。例としてプロセッサAとプロセッサBの間で1ワード分のデータ通信を行なう場合（プロセッサAがデータを書き、プロセッサBがそのデータを読む）を考える。性能の比較を表1に載せる。表1の値を求めるにあたっては、お茶の水1号の設計値として2次キャッシュアクセスに6クロック、メインメモリアクセスに24クロックかかるものとし、仮定としてバスは常に空いているものとした。

状況1はプロセッサAが、プロセッサBがリードリクエストを発行する前にデータの書き込みを行なった場合である。この場合、同期ビットを用いる機構ではデータ通信のための一回のメモリアクセスのみでよいが、同期変数を用いる場合ではデータ通信以外に同期変数のためにもメインメモリを読みにいかなければならない。状況2はプロセッサBが先にリードリクエストを発行し、プロセッサAからのライトリクエストが後から起こった場合である。ただし、この場合はプロセッサAがデータのライトを行なった時点を基準にし、それからのデータ通信完了までの時間の比較を行なったものである。同期ビットを用いる機構ではプロセッサAがメモリに書き込んでいるデータをバスから直接取り込むのに対し、同期変数を用いる場合では最初に同期変数をアクセスし、その後でメモリにアクセスしにいかなければならない。このとき同期変数はキャッシュされているものとし、スヌーププロトコルとしてアップデイト型を用いたものとしている。インバリデイト型のプロトコルを用いた場合は同期変数のアクセスのためにもメインメモリへのアクセスが必要になる。

表 1. 同期性能の比較

	同期ビット	同期変数
状況 1	24 クロック	48 クロック
状況 2	4 クロック	30 クロック

なお、ここでの比較はメモリへのアクセスのみに注目したものであったが、同期成立のチェックやバスのアービトレーションによる待ちなどを考慮すればさらに両者の差は広がるものと考えられる。

5 おわりに

細粒度並列マルチプロセッサお茶の水1号上のメモリベースのデータ駆動同期機構について、その実現方法および予備的な性能について述べた。今後の課題として、実際のアプリケーションをお茶の水1号上で並列実行させることにより本機構の有効性を評価していく予定である。

謝辞

お茶の水1号の作成にあたりチップを提供して頂いた日本電気株式会社に深く感謝致します。

参考文献

[1] Arvind, and R. A. Iannucci, "Critique of Multiprocessing von Newmann Style," in *Proc. 10th Int. Symp. on Computer Architecture*, pp. 426–436, June 1983.

[2] Jordan, H. F., "Performance Measurement on HEP—A Pipelined MIMD Computer," in *Proc. 10th Int. Symp. on Computer Architecture*, pp. 207–212, June 1983.

[3] 松本 尚, 平木 敬, "超並列計算機上の共有メモリアーキテクチャ," 電子情報通信学会コンピュータシステム研究会報告, pp. 47–55, Aug. 1992.

[4] 松本 尚ほか, "スヌープキャッシュを用いて通信と同期を統合する機構," 電子情報通信学会コンピュータシステム研究会報告, no. CPSY90-42, pp. 25–30, July 1990.