マルチプロセッサチップを想定した 並列計算機テストベット ATTEMPT-1 –キャッシュの実装– [*]

黒澤 飛斗矢[†]　寺澤 卓也[‡]　井上 敬介[†]　天野 英晴[†]
慶應義塾大学 理工学部[†]
東京工科大学 情報通信工学科[‡]

1 はじめに

我々はマルチプロセッサチップを念頭に置いた新 Keio プロトコルを提案している。マルチプロセッサチップではチップ内のバスの動作速度がはるかに高速になる反面、チップ内外の転送容量のギャップが問題になると考え、このプロトコルでは同期、メッセージ転送も含めて主記憶との転送を最小化することを可能としている。

また、新 Keio プロトコルではキャッシュに同期機構をサポートしており、Fetch-and-Decrement(F&D) とプロセッサ間割り込みを可能にしている。

本報告では、現在開発中である並列計算機テストベット ATTEMPT-1 での、このプロトコルの実装状況について述べる。

2 新 Keio プロトコルの実装

ATTEMPT-1 のキャッシュシステムはそのコントローラ部分に SRAM タイプ Complex PLD の Altera FLEX8000 を使用する。FLEX8000 は高速でなおかつチップの内部配線遅延がある程度予測でき、書き換え可能なことから、設計を変更して違うプロトコルの評価を取ることも可能である。キャッシュコントローラ部分をチップ化することで、キャッシュシステムの高速化を図り、実装上の電気的特性なども向上できる。

キャッシュコントローラの動作記述はアルテラ・ハードウェア記述言語 (AHDL) によって行う。ハードウェア記述言語を用いての設計は書きやすさ・読みやすさだけでなく、それを思考し理解する上でも重要である。

キャッシュシステム全体を大きく分けると、キャッシュコントローラチップ、データメモリ、情報を格納するメモリ、PLD を含むその他の論理回路から成る。情報を格納するメモリは、タグメモリ、ステータスメモリ、LRU メモリ、プロトコルメモリである。メモリには高速の SRAM を使用し、タグメモリやステータスメモリはプロセッサ側とバス側の両方から参照できるようにマルチポート化する。これらは全て各プロセッサボード上に実装する。

ATTEMPT-1 におけるキャッシュシステムの概略図を図 1に示す。

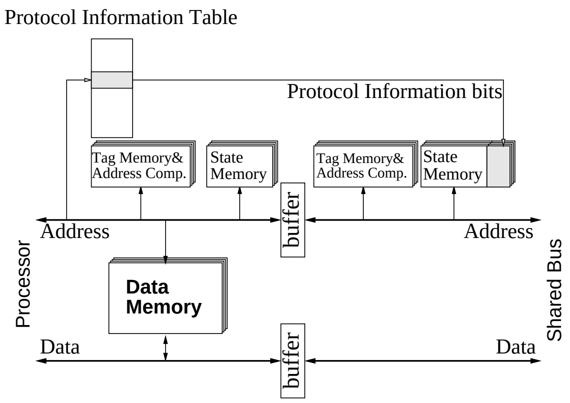

図 1: キャッシュシステムの概略図

2.1 キャッシュコントローラ

キャッシュコントローラはプロセッサ側とバス側のアクセスが独立に動作するため、それぞれ別の動作記述が必要である。そのため、プロセッサ側とバス側でタグメモリなどの参照は同時に行えるが、更新は同時には行えなず、その場合はアクセスを排他的に行う必要がある。また、プロセッサ側とバス側でそれぞれの状態にしたがって動作し、その動きの概要を図 2に示す。

タグの比較、キャッシュステータスの変換、LRU 制御、プロセッサ間割り込みなどはキャッシュコントローラチップの外部に配置し、ハードウェア的に行う。キャッシュコントローラはそれぞれの状態に応じてこれらの制御を行っている。

この様に、基本的に、キャッシュコントローラは外部の情報に基づいて自身の状態を変えそれに合わせた

---

[*] ATTEMPT-1: A test-bed for on-chip multiprocessors

[†] Hitoshi Kurosawa, Keisuke Inoue, Hideharu Amano, Keio University

[‡] Takuya Terasawa, Tokyo Engineering University

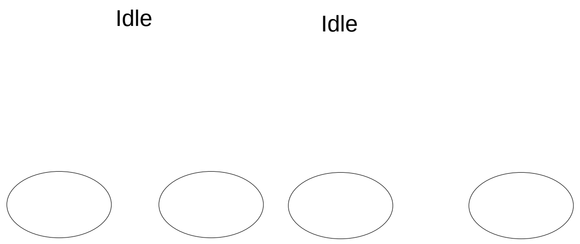

図 2: キャッシュコントローラの状態図

制御を行っているため、多少プロトコルが違っても、その記述に大きな変化はない。

2.2 キャッシュパラメータ

マルチプロセッサチップでは実装面積上の問題でキャッシュサイズなどが制限されることが考えられるので、キャッシュサイズによる性能の変化を評価するために、ATTEMPT-1 ではキャッシュ性能に影響を与える各種パラメータを設定可能にしている。それらの情報は各プロセッサ上に置かれ、ソフトウェアでっ動的に変更できる。

ATTEMPT-1 で設定可能なキャッシュパラメータを表 1に示す。

表 1: ATTEMPT-1 で設定可能なキャッシュパラメータ

パラメータ	設定範囲 (単位)
エントリ数	4K,8K,16K,32K(エントリ)
ブロックサイズ	1,2,4,8,16,32(ワード)
セットアソシアティビティ	1,2,4(セット)

2.3 同期機構

2.3.1 F&D

新 Keio プロトコルでは同期機構として F&D の機構をキャッシュに組み込んでいる。F&D を実現するために、キャッシュ内にラッチ付きの減算器を設ける。

F&D 操作はメモリ内容を更新するため、キャッシュコヒーレンス維持のため、無効化ブロードキャストで他のプロセッサのキャッシュに知らせる必要がある。従ってバス結合マルチプロセッサでは、F&D を行うプロセッサはバスマスタになる必要がある。そこで、これをシリアライズに利用して、まず、バスマスタになってからデータを読みだしプロセッサに返し (フェッチ)、その後バスに減算 (デクリメント) した値がブロードキャストされ、他のコピーを無効化または更新する。最後に、F&D を行ったプロセッサのデータメモリの値も 1 減算する。

これによって、F&D を行ったプロセッサがアクセスした後は、1 減算されたデータが全てのプロセッサからアクセスされることになる。さらに、バスのアービトレーションをオペレーションに含めているため、この操作は不可分に行うことが可能である。

2.3.2 プロセッサ間割り込み

ATTEMPT-1 ではプロセッサ間割り込み機構をキャッシュに実装し、プロセスの再開のシグナルやページフォールト時の TLB フラッシュを全プロセッサに連絡する場合などに対応できるようにする。

これを実現する手軽な方法は、各プロセッサにユニークなアドレスを割り当て、そのアドレスに対するアクセスを行えば対応するプロセッサに割り込みがかかるようにしておく方法である。このアドレスはバスに出されるので、バスを監視する必要がある。

割り込みは全プロセッサか、指定したプロセッサ 1 つにかけることが可能である。キャッシュコントローラでは、割り込みであることを高速に検出しなければならないので、通常のキャッシュメモリのためのアドレス比較とは別に、割り込みアドレスの検出を行う専用の比較器を用意する。

2.4 他のプロトコルとの比較

新 Keio プロトコルとその他のプロトコル (Symmetry、Berkeley、Illinois、Dragon、Firefly) のキャッシュシステムのハードウェア的な比較を行った。その結果、次の表 2に示す。

表 2: 新 Keio プロトコルとその他のプロトコルの比較

	新 Keio	その他
Tag Memory Status Memory LRU Memory Protocol Memory	392KB	320KB
Cache Memory	256KB	256K
FLEX8000	2800gates	2800gates
PLD etc.	4233gates	3456gates

2.5 キャッシュコントローラの開発環境

キャッシュコントローラの開発にはアルテラ社が用意している MAX+PLUS II 開発システム (PC/Windows) で行っている。このシステムを使っ

て、設計仕様の入力から論理合成、コンパイル、シミュレーション、デバイスのプログラムまで行えるが、ロジックレベルでのシミュレーションをサポートしていないため、VHDL、Verilog などの検証ツールを用いてシミュレーションを行っている。

3　まとめ

　現状では、新 Keio プロトコルの実装は進んでおり、キャッシュコントローラの設計が終わり、シミュレーションにより検証を行っている。現在、キャッシュコントローラ周辺のキャッシュメモリシステム部を ATTEMPT-1 に実装中である。