

1B-7

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ド 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主記憶との転送を最小化するキャッシュプロトコルを提案し、並列計算機テストベッド ATTEMPT-1 上で実装を行なっている [1, 2]。様々なプロトコルについて評価を取るため、キャッシュの制御は、ハードウェア記述言語を用いて記述され、書き替え可能な FPGA 上に実装されている。本報告では、このキャッシュおよび同期機構の実装の詳細について述べる。

2 キャッシュの実装

キャッシュシステムは、図 1 に示すように、キャッシュコントローラチップ、データメモリ、キャッシュ制御情報メモリ、PLD を含むその他の論理回路から構成される。キャッシュ制御情報メモリには、タグ、ステータス、LRU、プロトコル情報が格納される。メモリには高速の SRAM を使用し、キャッシュ制御情報メモリのタグやステータスについては、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。

2.1 キャッシュコントローラ

キャッシュコントローラチップは 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ができる。

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。記述を書き換えることにより、容易にプロトコルの変更が可能である。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は独立に行なわれる。これに対応し、アクセス制御は図 2 に示すように、それぞれ独立に記述

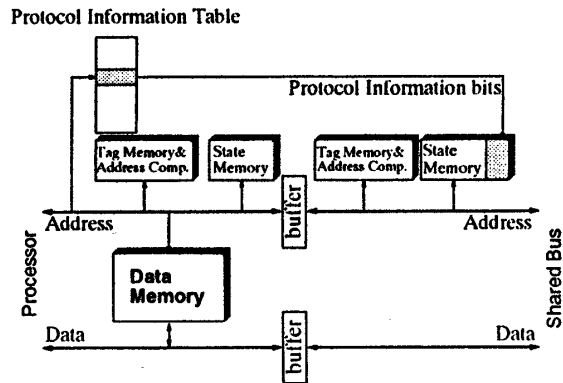


図 1: キャッシュシステムの概略図

している。更新についてのアクセスは排他的に行う必要があるため、二つの状態移行の間に排他制御用の信号を設けている。このことにより、プロトコルの可読性が向上し、変更も容易である。

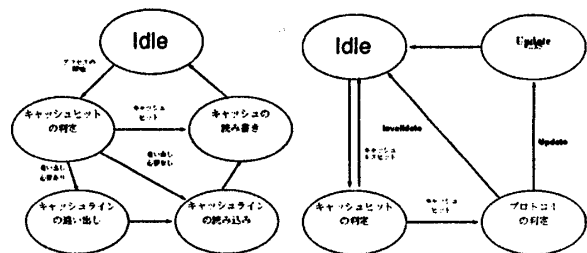


図 2: キャッシュコントローラの状態図

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に必要な回路は、キャッシュコントローラチップの外部に設ける。キャッシュコントローラは状態に応じてこれらの制御を行う。

2.2 マルチプロトコル

プロトコルはプロセッサ (IDT R3081) に内蔵されている MMU のページサイズ単位で、書き込み無効化

*ATTEMPT-1: A test-bed for on-chip multiprocessors - Implementation of cache system -

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

†Takuya Terasawa, Tokyo Engineering University

型と書き込み更新型を選択できる。

プロトコル情報は各プロセッサボード上に置かれ、ソフトウェアで動的に変更が可能である [3]。そのため、同じキャッシュラインで書き込み無効化型プロトコルと書き込み更新型プロトコルが混在でき、これを利用して受信者選択型のマルチキャスト転送も行うことが可能である。

2.3 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ことは重要である。このため、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。それらの情報はキャッシュ制御情報メモリ上に置かれ、ソフトウェアで動的に変更することができる。

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)

2.4 同期機構

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。この操作は書き込み更新型プロトコルにおけるデータの更新操作とほぼ同様の操作で行なう。

各キャッシュ内にはラッチ付きの減算器を設ける。F&Dを行うプロセッサは、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)、その後バスに減算器によって減算(デクリメント)した値をブロードキャストし、他のキャッシュ上の値を更新する。

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、バスのアービトラク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。

同期用のデータに対して書き込みを行なった場合、あるいは減算の結果が0になった場合に、あらかじめ

登録しておけば割り込みを発生することができる。

2.5 他のプロトコルとの比較

現在、既にキャッシュコントローラチップの設計は終了しており、様々なプロトコルの記述を行なっている。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量の比較結果を表2に示す。新 Keio プロトコルは他のプロトコルと比較し、制御はやや複雑であるが、制御回路のゲート数はほとんど増加していないことがわかる。

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	392KB	320KB
Protocol Memory		
Cache Memory	256KB	256K
FLEX8000	2800gates	2800gates
PLD etc.	4233gates	3456gates

3 まとめ

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。ロジックレベルでのシミュレーションは、VHDL、Verilog に変換して行っている。現在、コントローラチップの設計は終了し、周辺を含めたキャッシュを ATTEMPT-1 ブレッドボード上に実装中である。

参考文献

- [1] 小椋, 寺澤, 井上, 天野, “マルチプロトコルをサポートしたキャッシュシステムの実装と評価.” 信学技報 CPSY93-39, Nov. 1993.
- [2] 小椋, 寺澤, 井上, 天野, “WSI を意識したマルチプロセッサ用のキャッシュプロトコル”
- [3] 松本 尚, “細粒度並列実行支援機構”, 計測機アーキテクチャ研究会報告 No.77-12, 情報処理学会, pp.91-98, July 1989