

CAMを用いた同期機能付き共有レジスタ構成

1N-1

山脇 彰 東 由美 松田 孝史 岩根雅彦
九州工業大学 工学部

1 はじめに

近年、半導体技術の進歩によりマルチプロセッサオンチップシステムが可能になってきている¹⁾²⁾。このようなシステムでは同期機能を備えた共有レジスタ構成が、2項演算レベルでの細粒度並列処理において有効であり、いくつかの共有レジスタ構成が提案されている²⁾。ここに、マルチプログラミング環境を対象とした、細粒度並列処理のための同期/通信用共有レジスタを提案し、FPGAを用いて実装した結果を報告する。

2 CAMを用いた同期機能付き共有レジスタ

2.1 共有レジスタの構成

CAMを用いた同期機能付き共有レジスタ(単に共有レジスタと記す)の構成を図1に示す。

Shared Register		
TAG	DATA	COUNT
CAM	⋮	COUNTER

図1 共有レジスタの構成

共有レジスタのフォーマットは図1のように、タグ、データ、カウント回数の各フィールドからなっている。タグは共有レジスタを識別するためのものであり、カウント回数は通信の対象となるプロセッサ数である。タグフィールドは一致検索のCAMで構成されている。カウントフィールドはカウンタで構成されており、それ自身の非ゼロ/ゼロでもってデータの有効/無効を表す。

2.2 共有レジスタの動作

2.2.1 読み出し時の動作

タグが一致し、カウントフィールドが非ゼロのエントリが存在する場合、そのエントリの内容を読み出し、カウンタを1だけカウントダウンする。タグが一致し、カウントフィールドがゼロのエントリが存在する場合は、読み出しはブロックされる。タグの一致するエントリが存在しない場合も、読み出しはブロックされる。

2.2.2 書き込み時の動作

カウントフィールドがゼロのエントリが存在する場合、そのエントリにタグ、データ、カウント回数を書き込む。タグが一致し、カウントフィールドが非ゼロのエントリが存在する場合は、書き込みはブロックされる。共有レジスタが満杯の場合も、書き込みはブロックされる。

2.4 共有レジスタ指定

命令形式は共有レジスタをLOAD, STORE, MOVE, その他の命令に関して、基本的にはローカルレジスタと同様に扱う。

MOVE SR1, R1, 1 R1の内容をSR1に格納(カウント1)

LAOD SR1, A, 1 アドレスAの内容をSR1に格納(カウント1)

ADD SR1, R1, 2 SR1とR1の内容を加算しSR1に格納(カウント2)

上に示した例においてR_iはローカルレジスタ, SR_iは共有レジスタ, 第3オペランドはカウント回数である。また、命令セットおよび機械命令のフォーマットは個々のインプリメンテーションに依存する。

3 共有レジスタによる同期および通信

3.1 相互排除

ある共有レジスタに対する書き込みと読み出しを組み合わせることによってクリティカルセクションを図2に示すように実現できる。

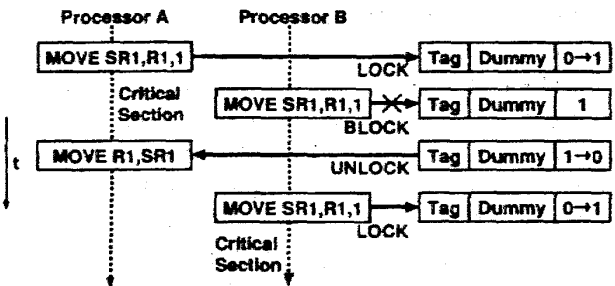


図2 相互排除の実現例

まず、プロセッサAがTagで識別されるSR₁への書き込みをカウント回数を1として行っている。このことによって、SR₁をロックしたことになる。その後、プロセッサBが同じSR₁に対しロックを行おうとしているが、カウンタが非ゼロであるためブロックされクリティカルセクションに入ることができない。プロセッサAがクリティカルセクションを抜ける際に、SR₁を読み出し、カウントフィールドが0になってはじめてプロセッサBはクリティカルセクションに入ることができる。

3.2 条件同期

プロセッサによる共有レジスタの読み出しは、その共有レジスタへの書き込みが発生しない限りブロックされることを利用して、生産者消費者間の同期を保証できる。図3において、消費者1がTagで識別されるSR₁の読み出しを行っているがカウントフィールドがゼロ、つまりデータが生産されていないのでブロックされる。その後、生産者によってSR₁にデータが書き込まれているが、この場合、通信を行うプロセッサ数が2である

A Shared Register with Synchronization Using CAM
Akira YAMAWAKI, Yumi HIGASHI, Takashi MATSUDA,
Masahiko IWANE
Faculty of Engineering, Kyushu Institute of Technology

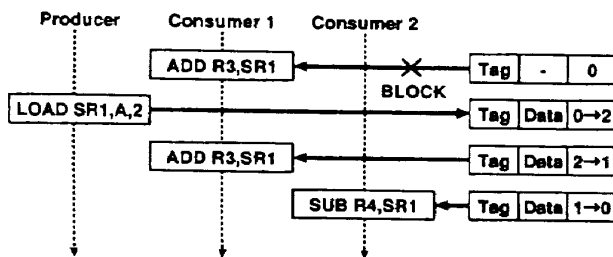


図3 条件同期の実現例

のでカウントフィールドに2を書き込んでいる。カウントフィールドが非ゼロになったので、消費者1、消費者2ともにデータを読み出すことができる。例からわかるように、カウンタを設けたことにより1対多の通信と同期を容易に実現できる。

4 スレッドの並列実行モデル

4.1 マイクロスレッド

マルチスレッドにおいて、タスクはそれに属する複数のスレッドの実行環境であり、スレッドはタスク内の一連の命令実行である。そのスレッドに対し、2項演算レベルの細粒度並列性を考慮してより細かく並列化を行う。この並列化された部分をマイクロスレッドと呼ぶ。このように階層化を行うことによって、スレッド中に、粗粒度並列処理を、マイクロスレッドに細粒度並列処理を対応させることができる。

4.2 タグの生成

タグとして共有レジスタにタスクIDを付加することによって、各タスクに属したスレッド→スレッド、スレッド→マイクロスレッド、マイクロスレッド→マイクロスレッド間で共有レジスタを介した通信と同期を行うことができる。タスクIDは、プログラムのローディング時にOSによって、機械命令の共有レジスタ指定フィールドに付加される。このようにすることによって、共有レジスタをあらかじめ物理的にグループ化する必要がなく、さらに、空きエントリの有効利用が可能である。

5 評価システムの開発

5.1 マルチプロセッサシステムの概要

マルチプロセッサオンチップの理想としては、単純なスカラプロセッサを半導体のリソースが許す限り多数搭載することだが、実現することは困難であるので、共有レジスタの有効性を検証するためのテストベッドとなる個別マイクロプロセッサによるマルチプロセッサシ

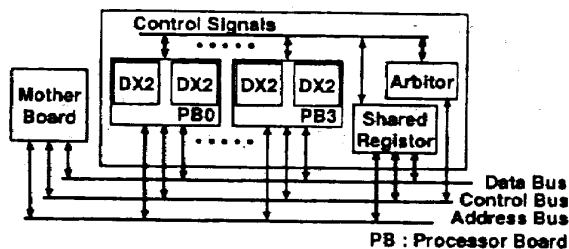


図4 マルチプロセッサシステムの構成

テムを開発している。プロセッサとして486DX2を使用し共有レジスタをFPGAで実現して、早期実現の達

成のために既存のDOS/Vマシンを改造する。開発しているマシンは対称かつ単一バス共有メモリ型のマルチプロセッサであり、プロセッサボードにはそれぞれ2台の486DX2プロセッサを搭載し、合計で8プロセッサシステムとなる。

市販のプロセッサを用いるために、実際に共有レジスタを操作する命令を実装することができない。そのため共有レジスタに対する操作は、IOポートにアクセスすることによって実現する。アクセスを1回の命令で終了できるようにタグおよびカウント回数はIOアドレス部に埋め込んでいる。主メモリアクセスよりも高速なIOアクセスにより、主メモリに対して優位性を持つと考えられる。

5.2 FPGAによる共有レジスタの実現

共有レジスタのフォーマットは、8プロセッサであるのでカウンタは3ビット、タグは7ビット、データは32ビットであり、エントリ数は64である。

設計においては、共有レジスタを機能的に複数のブロックに分け、それぞれをFPGAに割り当てた(表1)。全体としてコントローラ部とレジスタ部の各ブロックに大きく分け、レジスタ部はさらに、タグフィールドとカウンタ、データフィールドの2つのブロックに分割した。使用したFPGAはXilinx社のXC4005EとXC4013Eである。コントローラ部にXC4005Eを1個、タグフィールド部とカウンタ、データフィールドにXC4013Eをそれぞれ2個使用した。内部リソース使用状況はFPGA1個当たりのものを記している。また、内部リソ

表1 FPGAの割り当てと内部リソース使用状況

Block	FPGA	CLBs	IO Pins	F,G Function Generators	H Function Generators	CLB Flip Flops	
Controller	XC4005E	73%	63%	53%	21%	23%	
Register	Tag	XC4013Ex2	63%	32%	48%	12%	19%
	Data Counter	XC4013Ex2	61%	47%	46%	28%	8%

スは仕様変更に対応できるように余裕を持たせている。以上により、5個のFPGAにより実現可能であることがわかった。作成した共有レジスタへのアクセスは、タグフィールドの一致信号とカウントフィールドの非ゼロを示す信号とのANDによってヒットしたかミスしたかを判定している。そこで、タグが入力されてから判定結果が出力されるまでの遅延をタイミングシミュレータを用いて調べた結果、106nsであることがわかった。

6 結び

現在、マルチプロセッサシステムにおいて、プロセッサボードおよび共有レジスタボードの設計が完了し、実装段階である。今後は、プロトタイプが完成次第、目的に沿った評価を行っていく予定である。

参考文献

- [1] Kunle Olukotun: "The Case for a Single-Chip Multiprocessor", Proc of ASPLOS-7, October 1996.
- [2] 岩下, 宮鳴, 村上: 次々世代汎用マイクロプロセッサ・アーキテクチャ PPRAM の概要, 情報研報, ARC-113-1, 1995.