

クロスバススイッチ結合マルチプロセッサに適した 排他制御機構の提案

Proposal of the Basic Mutual Exclusion Mechanism for the Multi-Processor
Connecting Processors and Main Storages with the Crossbar Switch

山本 登[†]

1. はじめに

密結合マルチプロセッサシステムの論理的構成法の検討を長く続けている。p組の処理装置 (IPU) 群とm組の主記憶装置 (MSU) 群をp×mのクロスポイントスイッチを内蔵した記憶スイッチ装置 (SWU) で接続する。SWUにはIPU対応にIPU接続部 (PIFC)、MSU対応にMSU接続部 (MIFC) を設けそれぞれIPU・SWU間の接続路 (P接続路) とSWU・MSU間の接続路 (M接続路) による情報転送動作を担わせる。

SWUには共有キャッシュ (CC) のほか、先取りを目的とした後置キャッシュ (BC)、キャッシュの内容一貫性維持機構、FIFO式参照要求受付機構、スイッチ閉閉制御機構、メモリ参照準備機構、参照終結機構、メモリ参照制御機構 (MAC)、BCとMSUの参照制御を担う主記憶参照制御機構などをM接続路を単位としてm組分設ける。

本システムのように共有主記憶をもつマルチプロセッサでは、同時に参照することを許さない情報資源のために長期・短期の相互排除機構を必要とする。このうち、短期相互排除機構はTest & Set命令と呼ぶ機械語命令で実現することが多い。排他制御の対象となる全ての情報資源に各1バイトの情報を割り当てて占有の可否を制御する。本稿では検討中のシステムに適した短期排他制御機構の原理と実装を示すと共に、それをを用いた長期占有型相互排除資源のためのセマフォ機構の試作内容について述べる。

2. 基本排他制御機構の設計

2.1 論理仕様の設計

(1) Test & Set命令の問題点

短期の相互排除に用いられるTest & Set命令では、命令で指定した主記憶位置のバイトが1か否かを検査すると共にそのバイトを1に書き換える。当該バイトの読み出しから書き換え迄は不可分の動作で、その間は他のメモリ参照は禁止される。資源占有の可否は読み出した内容が0か否かで判断する。占有できない場合はTest & Set命令を繰り返し占有可能になるのを待つ (スピンロック)。この命令の繰り返しでプロセッサが占有されることも問題だが、共有主記憶への参照が時間的に集中する点こそ問題である。

(2) Test & Set命令の改善案

密結合マルチプロセッサシステムに生起する短期の資源占有の必要を満たしつつ、上述のTest & Set命令の欠点を解消する排他制御機構の実現を目指す。実現に際し従来のTest & Set命令の概念を大きく変えることは避ける。

- ◆ 占有対象資源を表す情報の実体を何にするにせよ全てのプロセッサから参照可能でなければならない。
- ◆ Test & Set命令は書き込み動作を伴うため問題が多い。しかし排他制御資源を表す情報を主記憶域にデータとして置く論理的必然性はなく、他の方法を考える余地がある。
- ◆ 占有許可を得る迄待たせる場合は占有資源毎に待機プロセスの属性を記録した待行列用データ構造が必要となる。

また、物理的に待機する機構と再開する機構についても検討する必要がある。

(3) 排他制御の仕様

資源占有を拒否された場合も含め命令の発行は1回で済む方式とし、資源占有命令のバースト的発行を防ぐ。このため占有を認められない時はハードウェア的に待ち状態とし、占有可能になったら再開させる方式を用いる。

(a) 占有にはTest & Set、解除にはTest & Reset命令を発する。資源番号は命令語の実効アドレスで指定する。

(b) 占有できないときIPUはハードウェア的に命令の実行を凍結し待ち行列にIPU番号を記録する。

(c) 占有が終わったIPUはTest & Reset命令を発する。待機中のIPUがあれば待ち行列の先頭にあるIPUの (凍結中の) Test & Set命令を再開する。無ければ当該資源を利用可能状態に表示してTest & Reset指令の処理を終わる。

2.2 基本排他制御の実装

実効アドレスで指定する占有資源番号は主記憶空間ではなく資源アドレス空間である。しかし排他制御機構は全てのIPUから共通に参照できる場所になければならない。

(1) 排他制御機構の実装箇所

排他制御機構はm組のMAC部に分散して実装する。PIFC部は資源番号下位部を解読してm組あるMAC部の一つを選び、Test & Set/Test & Reset指令の処理を要求する。

(2) 資源管理記憶

MAC内の排他制御機構は資源番号毎に占有状態情報と占有解除を待つIPU番号リストを記憶した図1の待ち行列をもつ。待ち行列の実体は小容量高速メモリである。

占有状況	!	待ち IPU リスト
フラグ	:		

管理表は資源番号毎のエントリをもち、資源の占有状況フラグと占有解除を待つスレッド・プロセスの番号を先着順に記憶する。

図1. 相互排除資源管理表

(3) Test & Set命令の動作

IPUは実効アドレスを計算するとSWUにTest & Setの処理要求を発して応答を待つ。応答の受領は占有許可を意味する。応答が来る迄待つ事の可能な制御系を設計する。

(4) メモリ参照指令処理部の動作

MACはTest & Set指令を受けると指定された番号の資源の占有状態を調べる。非占有状態なら占有状態に変えると共に占有許可を当該IPUに返し、占有中なら待行列にIPU番

[†] 日本大学工学部情報工学科

号を登録して処理を終える。このためIPUやPIFC内のTest & Set指令処理部は応答待のままとなる。この状態にあっても当該IPUへのキャッシュの一貫性保証に必要な機能は動作できるように設計する。

Test & Reset指令を受けると待行列を調べる。空なら当該資源を非占有状態に変更すると共に指令を発したIPUに応答し処理を終える。待行列が空でないなら待行列の先頭に登録されたIPUに占有許可信号を返すと共に待行列の更新とTest & Reset命令を発したIPUへ応答を返送する。

3. 長期排他制御機構：セマフォの設計

Test & Set/Test & Reset命令による短期排他制御機構を用いてブロックを伴う比較的長期の排他使用のための機構を試作した。

3.1 論理仕様の設計

研究室で開発した『密結合マルチプロセッサシステムシミュレータ』¹⁾に5種のセマフォ系命令を含むセマフォ機構を実装した。シミュレータはメインフレーム上で動くGPSS言語で記述されている。この関係で新設のセマフォ系命令はメインフレームの命令体系に準じた形式をもつ。

◆**Create Semaphore** 主記憶上のセマフォ変数のアドレスによりセマフォ変数を登録する。セマフォ変数に初期値を設定し、汎用レジスタ：GROにセマフォ番号を返す。

◆**Delete Semaphore** 実効アドレスで指定した番号のセマフォを削除する。

◆**Reserve Semaphore** 実効アドレスで指定したセマフォ番号で表す資源の占有を要求する。占有できればセマフォ変数の値を0にする。条件コードは占有できた場合は0、できない場合は1にする。プログラムは条件コードが1であることから占有不可を認識し、Wait Semaphore命令を実行して待ち状態になる。

◆**Release Semaphore** 実効アドレスで指定したセマフォの占有を解除する。当該セマフォの解放を待つプロセスがあれば待ち行列の先頭位置のプロセスを起動し、なければセマフォ変数の値を1にして処理を終える。

◆**Wait Semaphore** 実効アドレスで指定したセマフォの割り当てを待って休止する。Reserve Semaphore命令で条件コードが1の場合に発行しなければならない。

3.2 セマフォ機構の実現

(1) 命令語の実装

シミュレータの命令処理部にセマフォ関係の命令の機能処理部をGPSSで記述して追加した。

(2) 関連データ構造

(a) セマフォ管理表

排他制御の対象とする資源単位に1行を割り当てるセマフォ管理表を設ける。行の内容はセマフォ番号で参照するが、番号の有効性と対応するセマフォ変数のアドレスをこの表で管理する。この管理表の各行はCreate Semaphore命令で登録され、Delete Semaphore命令で削除される。

(b) セマフォ待行列

図3の待ち行列表はセマフォ番号を指標にして参照する。セマフォ番号を単位に待機中のタスク数と待機しているタスクのタスク管理表へのポインタを登録する。

Wait Semaphore命令でカウンタが加算されると共に、ポインタが追加登録される。Release Semaphore命令が実行されると待機中のタスクが再開されるが、同時にカウンタを減じると共に待行列から再開タスクのデータを除く。

有効フラグ	セマフォ変数のアドレス	セマフォ番号
-------	-------------	--------

使用可能なセマフォの数だけのエントリをもつ。

図2 セマフォ管理表

行有効フラグ	セマフォID	待機タスク数	タスク管理表へのポインタ (19エントリ)
--------	--------	--------	-----------------------

使用可能なセマフォの数だけのエントリをもつ。

図3 セマフォ待行列

4. 応用プログラムによる動作確認

4,000個の整数を格納した配列を20等分して200個ずつのデータをもつ部分域を作る。その領域を20台のIPUに割り当て、部分和を求めさせては総和用変数に足し込ませる。この並列プログラムを密結合マルチプロセッサシステムシミュレータ上で実行させ排他制御機構の動作を確認した。

図4は短期相互排除を適用した場合のタイミング図である。主タスクは4,000個の被加算データを生成した後、20組の部分計算タスクを生成・起動する。実行時引数は加算データの先頭指標とデータ数、総和用変数へのポインタである。20組の部分計算タスクは同一プログラムコードを共用し部分和を計算し総和用変数に加え込む。すなわち総和用変数を読み込み・求めた部分和を加え・書き戻す。総和用変数は同時に複数のタスクから参照される可能性があるため相互排除の対象にしなければならない。

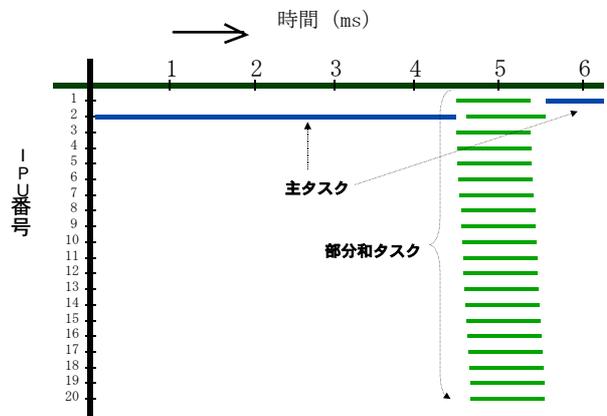


図4 動作確認プログラムのタイミング図

5. おわりに

クロスバススイッチ結合式マルチプロセッサの短期相互排除機構として、占有できるまでハードウェア的にTest & Set命令の実行を凍結する方式を考えその実装を試みた。更にそれを用いてより長期の占有に用いるセマフォの論理仕様と実装を考え密結合マルチプロセッサシステムシミュレータ上でプログラムを走行させて確認した。

参考文献

1) 山本：密結合マルチプロセッサシステムシミュレータの開発、情学論文誌、vol.30、No7、pp.878-894 (1989)。