同期通信用メモリを用いた高速バリア同期機構

4P－5

山脇 彰　田中 誠　岩根 雅彦
九州工業大学 工学部

1. はじめに

共有メモリ型のマルチプロセッサにおける並列処理では，プロセッサ間の同期と通信のオーバヘッドが問題となる．同期の高速化を目的としてハードウェアバリア同期機構が提案されている[2)3)]．一方，マルチスレッド環境下において同期と通信を同時に扱い，それらのオーバヘッドの削減を目的とした同期通信用メモリ TCSM(Tagged Communication and Synchronization Memory)[1)] がある．しかし，TCSM を用いたバリア同期はメモリのロック付き AND 命令を用いたソフトウェアバリア同期に対して 0.91 の速度向上であった．そこで，TCSM を TCSMII として任意参加型の高速バリア同期機構への拡張を行った．ここでは TCSMII の概要を述べ，それを用いたバリア同期の概念と実現方法を示す．そして，シミュレーションと実測によりメモリおよび TCSM によるバリア同期と TCSMII によるバリア同期を比較し TCSMII の有効性を示す．

2. 同期通信用メモリ TCSMII の概要

TCSM はマルチスレッド環境下において条件同期，相互排除，バリア同期を統一的に扱える同期通信用メモリである．TCSM に対して高速なバリア同期機構への拡張を行ったものが TCSMII であり，概念図を図 1に示す．TCSM からの拡張は各エントリでの WF(Wait Flag)1 ビットである．

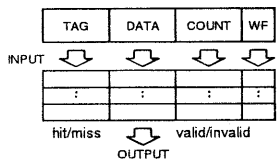

図 1　TCSMII の概念図

TCSMII の基本動作であるが，書込み動作に関して，WF をエントリに書込む以外は TCSM と同様である[1)]．読出し動作は，タグを入力しヒットしたエントリのカウントが非ゼロであった場合データを出力しカウントをデクリメントする．このとき，カウントが非ゼロで WF が 1 の場合，読出しを行ったスレッドのバスをブロックする．この状態をスレッドの待ち合わせ状態とよぶ．カウントがゼロになった場合，WF をリセットし，同一タグによる書込みブロックと待ち合わせ状態のスレッドを起動する．TCSMII に関する命令は STCSM 命令を拡張し，以下のようになる．

```
STCSM2 Data, Tag, Count, WF
Data : Registor, others : Immediate
```

3. 同期通信用メモリを用いたバリア同期

マルチスレッド実行モデルにおけるタスクはプログラムの実行環境であり，タスクに属するスレッドは一連の命令実行で一般的に粗粒度の並列性をもつ．スレッドはプロセッサの割当て単位となる．スレッド内部をより小さなレベルで並列化した部分をマイクロスレッドと呼ぶ．以降ふたつを総称してスレッドとよぶ．タグはタスク ID と変数名を連結したものとし，タスク毎に TCSMII のグループ化を行う．

ハードウェアバリア同期は同期に参加するプロセッサに対して同期情報を対応させるが，TCSMII によるバリア同期では同期に参加するスレッドに対し，タグで識別するバリア変数を対応させる．このことにより，プロセッサを仮想化できマルチプログラミング環境に容易に適応できる．タグで識別されるバリア変数を用いるため，任意の同期点において同期に参加するスレッドのみが同期をとればよい．タグの決定と同期命令のプログラムコードへの挿入は静的に行われる．

TCSM を用いたバリア同期の実行例を図 2に示す．バリア変数として同期点 i でタグを 1，同期点 j でタグを 2 とし，バリア同期が必要なスレッド間でのみバリア同期を実行している．また，T_1, T_4 は同期点 i，T_2, T_3 は同期点 j のバリア同期の完了を待たない．

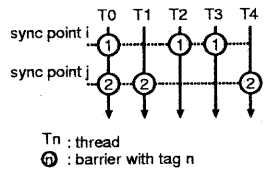

図 2　バリア同期実行の様子

4. バリア同期の実現法

スレッドに対する WF を用いた TCSMII 読出し後のバスブロックと，TCSMII の同一タグによる上書きブロックを利用し，任意参加型の高速バリア同期を実現する．プログラムを以下に示す．

```
control thread                    other threads
  STCSM2 Rn, Tag, nt - 1, 1         LTCSM R1,Tag
  STCSM2 Rn, Tag, nt - 1, 0         LTCSM R1,Tag
  ( Rn : registor, nt : number of thread )
```

バリアグループに属するタグを管理するスレッドを制御スレッド，その他のスレッドを一般スレッドと呼ぶ．制御スレッドは同期情報をタグとし WF を 1 にして TCSMII に書き込み，次に，WF を 0 として同一タグによる書き込みを行う．このとき，通信回数はバリアグループの要素数-1 とする．一般スレッドは同期情報を示すタグを用いて TCSMII 読出しを 2 回連続して実行する．

スレッドのブロックにバスバックオフ機能 (BOFF)[4)] を用いているため，WF=1 のエントリに対する読出し自体をブロックすると，ブロックの解除後にスレッドは同一エントリを再度読出してしまう．この再読出しを防ぐために，WF=1 のエントリに対する読出しは完了させ，その後 BOFF によってバス動作をブロックするようにした．そのため上記の実現方法になった．

The High Speed Barrier Mechanism Using a Communication and Syhchronization Memory
Akira Yamawaki, Makoto Tanaka, Masahiko Iwane
Faculty of Engineering, Kyushu Instisute of Technology

3つのスレッドによるバリア同期の例を図3に示す．トラッ

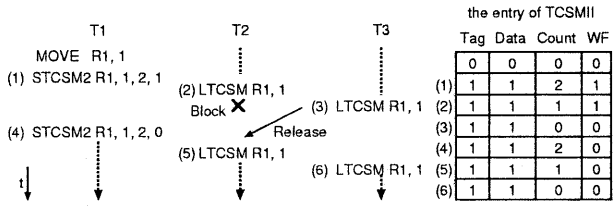

図3 TCSMIIを用いたバリア同期の例

ピングフェーズ ((1)〜(3)) で，制御スレッド (T_1) は一般スレッド (T_2, T_3) に対してタグを1，通信回数を2，WFを1にしてTCSMIIに書き込む．T_2 がタグを1としTCSMIIを読み出し，TCSMIIのカウンタは2から1となる．カウンタが非ゼロでWF=1より，T_2 のバス動作がブロックされ次のLTCSM2命令が実行できない．T_3 のTCSMIIの読み出しによりカウンタが0になり，WFが0にリセットされ T_2 が起動される．T_2, T_3 のTCSMII読み出しが完了する前の，T_1 のTCSMIIへの2回目の書き込みはブロックされる．リリースフェーズ ((4)〜(5)) では，T_1 が T_2, T_3 への1対多通信を行い，各スレッドはバリア同期から抜けていく．

5. 実験 [1)]

対象マシンのマルチプロセッサMTA/TCSMII[4)]は，MTA/TCSMに対して任意参加型バリア同期機構への拡張とともに，TCSMアクセスをIO命令からmov命令に変更し，キャッシュ無効化信号をメモリに書き込んだプロセッサには送らないようにした改良機である．MTA/TCSMIIは単一バス結合の共有メモリ型マルチプロセッサであり486DX2を8台搭載し，バス優先度の制御は回転式である．

シミュレーションにおいて実行時間を，バスサイクルにオーバラップできない命令にかかるクロック数の総和 (T_{nove}) とバスサイクルにかかるクロック数の総和 (T_{bus}) の和として求めた．使用するMTA/TCSMIIの基礎データを表1に示す．シミュレーションの条件として，プログラムをスレッド (p=2〜8) とし，すべてを同時にプロセッサへ割り当てる．バス優先度の初期値は制御スレッドが最も高く順次回転していくとする．TCSMIIアクセス命令はmov命令で実装されており，タグ，カウント，WFはアドレスに埋め込まれている．

表1 基礎データ

略称	説明	CLK
W_{tcsm}	TCSM(II) 書込みのバスサイクル時間	4
R_{tcsm}	TCSM(II) の読出しのバスサイクル時間	4
RB_{tcsm}	TCSM(II) 読出し失敗のバスサイクル時間	3
And	lock付きAND命令のバスサイクル時間	8

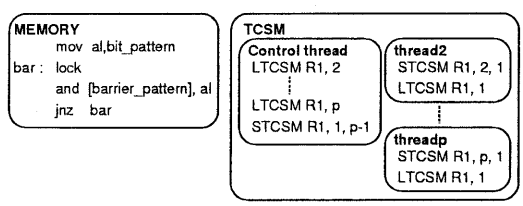

図4 メモリとTCSMを用いたバリア同期

MEMORYでの実行時間は次式となる．

$$T_{mem} = p + (p-1) \cdot And + T_{nove} = 16 \cdot p - 8 \quad (1)$$

TCSMIIでは次式となる．

$$T_{tcsm2} = [W_{tcsm} + (p-1) \cdot R_{tcsm}] + [W_{tcsm} + (p-1) \cdot R_{tcsm}] + T_{nove} = 8 \cdot p \quad (3)$$

第1項はトラッピング・フェーズのバスサイクル時間で第2項はリリース・フェーズのバスサイクル時間である．式(1)〜(2)の結果を表2に示す．MTA/TCSMにおけるTCSMの評価結果[1)]も表に入れておく．TCSMIIはMEMORYに対して平均1.80，TCSMに対して平均2.03の速度向上を得た．TCSMはバリア同期の実行中に一般スレッドによるTCSM読出し失敗が p 回発生するが，TCSMIIには存在しないため，TCSMIIの方が良い結果となった．また，IO命令のレイテンシも影響していると考える．MEMORYはlock付きAND命令が8クロックのバスサイクルを必要とするためTCSMIIと比べて悪い結果となった．

表2 シミュレーション結果

	p=2	3	4	5	6	7	8
TCSMII	16	24	32	40	48	56	64
TCSM[1)]	27	45	63	81	99	117	135
MEMORY	24	40	56	72	88	104	120

シミュレーション結果から，MTA/TCSMIIを開発し実測を行った．実測結果を表3に示す．TCSMに対してMTA/TCSMにおける実測結果[1)]も表に入れている．使用したプログラムと条件はシミュレーションと同じである．TCSMII，TCSM，MEMORYを用いたバリア同期の実測結果はシミュレーションと一致し，実機でTCSMIIの有効性が確認できた．

表3 実測結果

	p=2	3	4	5	6	7	8
TCSMII	16	24	32	40	48	56	64
TCSM[1)]	27	45	63	81	99	117	135
MEMORY	24	40	56	72	88	104	120

6. 結び

TCSMに対してTCSMIIとして高速なバリア同期機構への拡張を行った．シミュレーションと実測で，TCSMIIを用いたバリア同期とメモリおよびTCSMを用いたバリア同期との比較を行った．結果から，TCSMIIはメモリと比較し平均1.80，TCSMと比較し平均2.03の速度向上を達成し，TCSMIIの有効性が確認できた．今後は，TCSMIIによるバリア同期機構を用いた並列処理の評価を行っていく．

参考文献

1) 岩根 他, "マルチプロセッサオンチップにおけるCAMを用いた同期通信用メモリ," 信学論, J83-D-I, No.3, pp.317–328, mar.2000.

2) O'Keefe, M.T .et, "Hardware Barrier Synchronization:Static Barrier MIMD(SBM)," 1990 Int. Conf. on Parallel Processing, Vol.1,pp.35–42, 1990.

3) Gupta, R., "The Fuzzy Barrier:A Mechanism for High Speed Synchronization of Processors, " Proc.3rd Int.Conf.on ASPLOS,pp.54–63, Apr.1989.n

4) 山脇 他, "バスバックオフ機能を用いた同期通信用メモリの制御方式," 情処学第61回全国大会予稿集, 6D-7, pp.77–78, 2000.