同期通信用メモリにおけるカウンタとブロッキングの効果

山脇　彰†（正員）　岩根　雅彦†（正員）

The Effect of Counter and Blocking for a Communication and Synchronization Memory

Akira YAMAWAKI† and Masahiko IWANE†, *Regular Members*

† 九州工業大学工学部，北九州市
Faculty of Engineering, Kyushu Institute of Technology, Kitakyushu-shi, 804–8550 Japan

あらまし　同期通信用メモリ TCSM（Tagged Communication and Synchronization Memory）のブロッキング機構について述べ，通信回数を書き込むカウンタの効果と，バスバックオフ機能を適用したブロッキングの効果について明らかにする．実験から，通常のメモリと比較して TCSM はバックオフとカウンタにより実行時間を平均で 39%に抑えた．その速度向上に対して，カウンタによる効果が平均で 58%，バックオフによる効果が平均で 42%となった．

キーワード　マルチプロセッサ，同期通信用メモリ，ブロッキング，カウンタ，バスバックオフ

1. まえがき

マルチプロセッサにおける文や部分文レベルでの並列処理においては，頻繁に発生する同期と通信のオーバヘッドの削減が重要となる．オーバヘッドを削減するために，同期と通信を同時に扱う構造化メモリとして，共有メモリにフル・エンプティビットを付加した I-structure [1] やそれを拡張してキュー構造をもたせた Q-structure [2]，ロックディレクトリを付加したキャッシュ [4]，共有メモリにカウンタを付加し CAM によってエントリの動的な割当てを可能とした TCSM [3] が提案されている．

ここでは，バスバックオフ機能 [5] を適用した TCSM のブロッキング機構について述べる．そして，カウンタとバックオフの効果を明らかにするためにマルチプロセッサ MTA/TCSM II（MultiThread Architecture/TCSM II）[6] で 1 対多通信と相互排除のシミュレーションと実測を行う．

2. 同期通信用メモリのブロッキング

2.1 TCSM の概要 [3]

TCSM は，マルチプロセッサオンチップでのマルチスレッド環境において，相互排除，1 対多通信，条件同期，及びバリア同期を統一的に表現できる同期通信用メモリであり，概念図を図 1 に示す．TCSM は

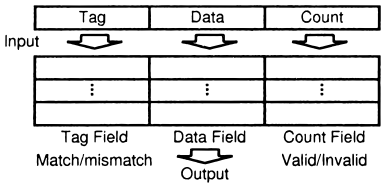

図 1　TCSM の概念図
Fig. 1　Concept of TCSM.

CAM（Content Addressable Memory）で構成され，1 エントリはタグ，データ，カウンタからなる．タグはエントリを識別するためのものであり，カウンタは通信回数を示し，非ゼロ/ゼロによってデータの有効/無効を表す．TCSM は無効なエントリからの読出しと有効なエントリへの書込みをブロックし生産者–消費者間の同期を保証する．カウンタは有効なエントリの読出し時にデクリメントされる．full/empty ビットとは異なり，TCSM ではカウンタに消費者数を指定できるため，1 対多の同期通信を 1 エントリで実現でき，そのエントリは全消費者による読出し完了後に暗黙的に解放される．

マルチスレッド環境でのタスクはリソースの保護単位であることから，タグをタスク ID と変数名の連結とし，TCSM をタスクごとに保護する．TCSM は同一タスクに属するスレッドまたはスレッドに内在する文や部分文レベルの並列性を抽出したマイクロスレッド間の並列処理に用いられる．

TCSM はマルチプロセッサオンチップ内部で全プロセッサコアから共有され，以下に示す命令で TCSM に対する書込み，読出し，リセット動作が実行される．STCSM（Store TCSM）命令は即値のタグと通信回数，レジスタのデータを TCSM に書き込む．LTCSM（Load TCSM）命令はタグを即値で指定し，レジスタに読み出したデータを格納する．RTCSM（Reset TCSM）命令はレジスタでマスクパターンを指定し，即値指定のタグで一致したエントリのカウンタをリセットする．

2.2 バスバックオフ機能の利用

生産者と消費者による書込みと読出しのブロックをバスバックオフ機能（BOFF:Bus back-OFF）により実現する．BOFF とは，プロセッサの現在のバス転送を中断させ再実行させる機能であり，BOFF がプロセッサに対してアサートされている間，プロセッサはバスを駆動できない．図 2 に BOFF を適用した TCSM アクセスに関するプロセッサの状態遷移図を

示す．各状態は S_0（ノンブロック），S_1（TCSM アクセス完了待ち），S_2（書込みブロック），S_3（読出しブロック）からなる．RM（Read Miss）は TCSM に対する読出し失敗，WM（Write Miss）は書込み失敗を表す．RH（Read Hit）は，TCSM に対する読出しが成功しかつそのエントリが解放されたことを，WH（Write Hit）は，TCSM に対する書込みが成功したことを表す．このブロッキング機構を実現するために，図 3 の BTT（Blocked Tag Table）を組み込んだ．BTT は，ブロック時の状態とそのときのタグを格納するためのテーブルで，タグフィールドは一致検索の CAM である．BTT のエントリがプロセッサに対応しており，BTT 内に格納されるタグは TCSM 内のタグと同一のものである．WB（Write Block flag）は，プロセッサが書込みブロック状態であることを，RB（Read Block flag）は読出しブロック状態であることを表す．これらの初期値は 0 であり，同時に 1 になることはない．

S_0 のプロセッサが TCSM へアクセスすると S_1 に遷移する．プロセッサが TCSM アクセス中に，WM が発生した場合，BTT のエントリにタグ，WB に 1 がセットされ，RM が発生した場合，BTT のエントリにタグ，RB に 1 がセットされる．このとき BOFF のアサートにより，バス転送が強制的に中断され，TCSM の読出し若しくは書込みに失敗したプロセッサのみがバスを駆動できない状態 (S_2,S_3) になる．BTT に対して，TCSM が RH を発生した場合，タグにヒットしたエントリの WB がリセットされ，TCSM が WH を発生した場合，タグにヒットしたエントリの RB がリセットされる．これにより BOFF がネゲートされ，プロセッサは S_2 若しくは S_3 から S_0 へと遷移する．

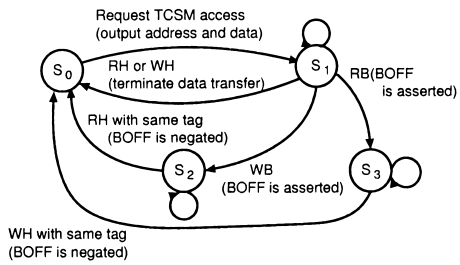

図 2　TCSM アクセスに関するプロセッサの状態遷移図
Fig. 2　State transition diagram of processor for TCSM access with bus back off.

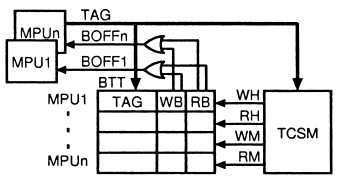

図 3　ブロッキング機構
Fig. 3　Blocking mechanism.

3. 実験の準備

3.1 実験環境

対象マシンのマルチプロセッサ MTA/TCSM II [6] は，MTA/TCSM [3] に対して，L1 キャッシュの無効化信号を書き込んだプロセッサ自身には送らないように変更し，TCSM をメモリマップド I/O にした改良機である．また，高速バリア同期機構への拡張も行っているがここではふれない．MTA/TCSM II は 8 台の 486DX2 [5] からなる単一バス結合共有メモリ型マルチプロセッサであり，486DX2 内にライトスルーの L1 キャッシュ，マザーボード上にライトバックの L2 キャッシュをもち，TCSM 及びバスアービタは単一バスに接続されている．バスアービタは集中型で，バスの優先度は回転式である．

実験では，プロセッサとスレッドを 1 対 1 に対応させる．シミュレーションでは初期値としてバスの優先順位を ID の最も小さいスレッドが最高で，その後順次回転すると仮定し，実行時間として，バスサイクル時間の総和とバスサイクルにオーバラップできない命令の実行時間の総和 (T_{no}) を加算し算出した．MTA/TCSM でのシミュレーションによる実行時間 [3] では TCSM アクセスに関して I/O 命令を用いていたが，MTA/TCSM II では mov 命令に変更して算出した．シミュレーションで用いた MTA/TCSM II の基礎データを表 1 に示す．メモリアクセスは最小値である．数値にはバス調停にかかる 2 クロック分を含んでいる．実測では，L1 と L2 ともにオンとし，プログラムのコード及びデータが L1 キャッシュに存在することを確認してから測定した．

3.2 評価プログラム

プログラムの説明で表 2 の記述を用いる．tcsm_w

表 1　MTA/TCSM II の基礎データ
Table 1　Basic data of MTA/TCSM II.

略称	説明	CLK
W_t	tcsm_w のバスサイクル時間	4
R_t	tcsm_r のバスサイクル時間	4
RB_t	TCSM 読出し失敗のバスサイクル時間	3
W_m	メモリ書込みのバスサイクル時間	4
R_m	メモリ読込みのバスサイクル時間	4(+3)
Bts	Test&Set 命令のバスサイクル時間	9

():バースト転送時の追加分

は第1引き数のタグ，第2引き数のデータ，第3引き数の通信回数を用いて TCSM へ書き込む．TCSM のブロッキングに BOFF を適用する場合，TCSM の検索と書込みが同時に行われ，WM 時にスレッドは BOFF でブロックされる．BOFF を適用しない場合，tcsm_w は TCSM の検索と TCSM へのデータ書込みで 2 回のバスアクセスを発生する．まず，タグを用いて TCSM を検索し WH/WM を調べ，WH 時は TCSM にデータを書き込み，WM 時は何もしない．検索と書込みは不可分動作であり，検索の結果は戻り値として返され WH 時は 1 を，WM 時は 0 を返す．

tcsm_r は第1引き数のタグを用いて TCSM を読み出し，その結果を第2引き数に返す．BOFF をブロッキングに適用する場合，TCSM の検索と読出しが同時に行われ，RM 時にスレッドは BOFF でブロックされる．BOFF を適用しない場合の動作は，tcsm_w と同様であり，RH 時に 1 を，RM 時に 0 を返す．

評価で使用したプログラムを図 4，図 5 に示す．両図において，TCSM_B，TCSM_NB は TCSM を用いて実装したプログラムであり，前者は BOFF によるブロックを行い，後者は BOFF によるブロックを行わない．MEMORY はメモリを用いて実装したプログラムである．図 4 のプログラムにおいて，TCSM_B は生成データに対する消費者の読出しブロック，MEMORY はデータ生成フラグ，TCSM_NB は tcsm_r の戻り値チェックによって通信データの正しさを保証する．図 5 のプログラムにおけるロックに関して，TCSM_B は TCSM の読出しブロック，TCSM_NB は tcsm_r を用いたビジーウェイト，MEMORY は TEST&SET 命令によるスピンロックで実現している．

表 2　TCSM アクセスに関する操作の記述
Table 2　Descriptions for TCSM access.

表記	説明
WH=tcsm_w(tag,data,count)	TCSM 書込み動作を実行
RH=tcsm_r(tag,&data)	TCSM 読出し動作を実行

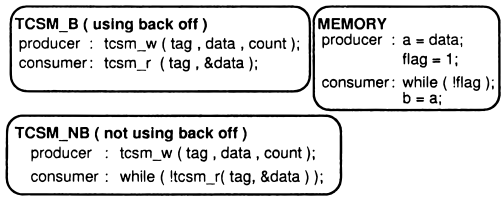

図 4　1 対多通信のプログラム
Fig. 4　Programs of multicast.

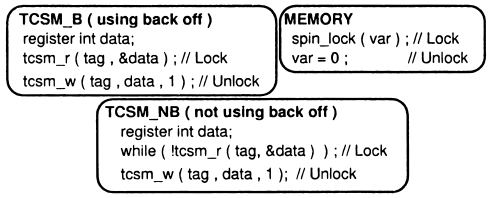

図 5　相互排除のプログラム
Fig. 5　Programs of mutual exclusion.

4. シミュレーション [3]

4.1　1 対多通信

図 4 のプログラムをスレッド ($p = 2$〜8) とし，生産者のスレッド ID が最も小さい状態でそれぞれをプロセッサに割り当てた．TCSM_B の実行時間 (T_{tb})，MEMORY の実行時間 (T_m) は次式となる，

$$T_b = 4 \cdot p \tag{1}$$

$$T_m = 21 \cdot p - 13 \tag{2}$$

TCSM_NB は，TCSM_B に対し消費者の tcsm_r による戻り値チェック分 ($p - 1$ 回)，実行時間が増加する．したがって，実行時間 (T_{nb}) は次式となる．

$$\begin{aligned} T_{nb} &= W_t + (p-1)(R_t + R_t) + T_{no} \\ &= 8 \cdot p - 4 \end{aligned} \tag{3}$$

シミュレーション結果を表 3 に示す．

MEMORY はフラグとデータが別であるため，データとフラグとしてのカウンタを同時に読み込める TCSM_NB と比較し，$p - 1$ 回バスアクセスが多い．TCSM_B は，ブロッキングにより tcsm_r の戻り値を調べる必要がないため，TCSM_NB と比べ $p - 1$ 回の TCSM アクセスを削減できる．MEMORY と比較して，TCSM_B は実行時間を 22%に，TCSM_NB は 56%に削減している．TCSM_B の MEMORY に対する速度向上の内訳はカウンタで 78%，バックオフで 22%である．

4.2　相互排除のシミュレーション

図 5 のプログラムをスレッド ($p = 2$〜8) とし，それ

表 3　シミュレーション結果（1 対多通信）
Table 3　Result of the simulation. (multi cast)

	$p = 2$	3	4	5	6	7	8
T_b	8	12	16	20	24	28	32
T_{nb}	12	20	28	36	44	52	60
T_m	29	50	71	92	113	134	155

表 4　シミュレーション結果（相互排除）
Table 4　Result of the simulation.（mutual exclusion）

	$p=2$	3	4	5	6	7	8
T_b	19	33	50	70	93	119	148
T_{nb}	24	48	80	120	168	224	288
T_m	35	66	106	155	213	280	356

ぞれをプロセッサに割り当てた．TCSM_B と MEMORY の実行時間は次式となる．

$$T_{tb} = \frac{3}{2} \cdot p \cdot (p-1) + 8 \cdot p \tag{4}$$

$$T_m = \frac{1}{2} \cdot p \cdot (9p + 17) \tag{5}$$

TCSM_NB は TCSM_B に対し，tcsm_r による戻り値チェック分（$p-1$ 回）だけバスアクセス回数が増加する．したがって，実行時間は次式となる．

$$\begin{aligned} T_{nb} &= p \cdot R_t + p \cdot (p-1) R_t + p \cdot W_t + T_{no} \\ &= 4 \cdot p^2 + 4 \cdot p \end{aligned} \tag{6}$$

シミュレーション結果を表 4 に示す．

TCSM_B は TCSM_NB での tcsm_r の戻り値チェック分だけ実行時間を抑えることができた．MEMORY が最も悪い結果となったのは式 (4)，(5)，(6) よりメモリの lock 付き TEST&SET 命令がバスを 9 クロック使用するためである．MEMORY と比較して TCSM_B は実行時間を 56%に，TCSM_NB は 22%に抑えた．TCSM_B の MEMORY に対する速度向上の内訳は，カウンタで 38%，バックオフで 62%である．

5. MTA/TCSM での実測

前章での結果を得て，BOFF によるブロッキングを適用した TCSM を MTA/TCSM で評価した．各プログラムの実行に関する条件はシミュレーションと同じである．1 対多通信の結果を表 5 に，相互排除の結果を表 6 に示す．表 5 の p は消費者数であり，数値は生産者がデータを生成してから最後の消費者がデータを読み出すまでにかかったクロック数である．表 6 の p は全スレッド数を表しており，2〜8 台までスレッドを変化させている．数値は全スレッドがプログラムの実行を開始してから完了するまでにかかったクロック数である．相互排除において，MEMORY の値がシミュレーションの結果と異なっている．lock 付き TEST&SET 命令でのメモリアクセスは L2 キャッシュにキャッシングされず主メモリ（DRAM）アクセスとなる [5]．したがって，DRAM のリフレッシュによるレイテンシの影響と考える．双方とも TCSM_B の方が良い結果となっており，1 対多通信及び相互排除におけるカウンタとブロッキングの効果を実機で確認できた．

表 5　1 対多通信の実測結果
Table 5　Actual survey result of multi cast.

	$p=1$	2	3	4	5	6	7
TCSM_B	8	12	16	20	24	28	32
MEMORY	29	50	71	92	113	134	155

表 6　相互排除の実測結果
Table 6　Actual survey result of mutual exclusion.

	$p=2$	3	4	5	6	7	8
TCSM_B	19	33	50	70	93	119	148
MEMORY	35	66	115	166	228	310	389

6. むすび

同期通信用メモリ TCSM のブロッキング機構について述べ，その機構を搭載したマルチプロセッサ MTA/TCSM で 1 対多通信と相互排除の実験を行った．通常のメモリと比較して，TCSM はバックオフとカウンタにより実行時間を 22〜56%に抑えた．その速度向上に対して，カウンタによる効果が 38〜78%，バックオフによる効果が 22〜62%であり，それぞれの効果についても確認できた．今後は，MTA/TCSM II での高速バリア同期機構に対する評価と一般的なプログラムによる評価を行う．

謝辞　日ごろからお世話になっている春日工作所（株）に深謝します．

文　献

[1] A. Nikhil and R.S. Nikhil, “I–structure: Data structres for parallel computing,” Trans. Prog. Lang. and Sys. ACM, vol.11, no.4, pp.598–639, Oct. 1989.
[2] 佐藤三久，児玉祐悦，坂井修一，山口喜教，“並列計算機 EM–4 における分散データ構造を用いたマルチスレッドプログラミング，” 情処学 ARC 研報，ARC–92, 1992.
[3] 岩根雅彦，山脇　彰，田中　誠，“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 (D-I)，vol.J83–D–I, no.3, pp.317–328, March 2000.
[4] T. Tarui, T. Nakagawa, N. Ido, M. Asaie, and M. Sugie, “Evaluation of the lock mechanism in a snooping cache,” International Conference on Supercomputing 92, pp.53–62, July 1992.
[5] Intel Corp,“インテル 486 マイクロプロセッサデータブック，” Intel Corp, 1992.
[6] 山脇　彰，岩根雅彦，“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.

（平成 12 年 12 月 25 日受付，13 年 3 月 29 日再受付）