

ショート・ノート

マルチプロセッサシステム用相互排斥モジュールの一設計*

古谷立美** 飯塚肇** 大表良一**

Abstract

This paper describes a Mutual Exclusion Control Module (MECM) which is designed to solve the mutual exclusion problem of a bus-connected multiprocessor system particularly for our ACE system. MECM is connected to a bus as an independent module, and other modules on the bus can realize mutual exclusion by using primitives to the MECM. The idea presented is easily applicable to other multiprocessor system.

1. ま え が き

共有資源を使用するマルチプロセッサシステムで、mutual exclusion (相互排斥) は、重要な問題である。従来これを解決する方法としては、IBM 360 などに使われた Test & Set 命令や文献 1) に示すものがある。Test & Set 命令は、あるプロセッサがこの命令を実行中に、他のプロセッサが Test & Set の対象となるビットにアクセスすることを禁じており、この方法は、Test & Set 用ビットを共有メモリに置ける利点がある反面、Test & Set という専用命令と、その命令中他のプロセッサが Test & Set 用ビットにアクセスすることを禁ずるハードウェアが必要となる。

この論文は、我々が開発中の ACE²⁾ と呼ぶバス結合されたマルチプロセッサシステムの相互排斥を実現するために設計した相互排斥モジュール (MECM) に関するものである。ここでは相互排斥の機能を 1 つの独立したハードウェア (MECM) に集め、各プロセッサは、Test & Set といった専用命令を用いず、MECM に対する一般のリード・ライト命令で相互排斥が実現される。そして同様の考え方に基づく文献 1) が Test & Set 機能を実現しているのに対し、MECM は Test & Set 機能の他、プロセスとプロセッサが対応付けられたマルチプロセッサシステムにおいて初期値 1 のセマホアを直接実現できる。以下に ACE の通信方式、

MECM の機構、相互排斥の実現法を示す。

2. ACE の通信方式

この章は、MECM の説明に必要な ACE の通信方式⁴⁾の概略を示す。故にここで述べるものは、以下の説明に必要な最小限のものであり、厳密さを欠く点もある。Fig. 1 は、ACE の一部を示している。Cバスを使用したいモジュールは、Cバスコントローラ (CC) にリクエスト (REQ) を送る。CC は、Cバスが使えるとき、REQ を発生したモジュールにセレクト (SEL) を返し、次にチェック (CK) という信号をCバスに結合された全モジュールに送る。REQ を発したモジュールは、CK に同期してCバス上にアドレスやリード/ライト等の情報を持つチェックデータ (CD) という情報を乗せ、他のモジュールは、CD から自分のモジュールが REQ に対してどう反応するかをチェックアンサ (CA) で答える。CA には、通信に即応できることを示す YES、通信に関与しない NO、通信のための準備中 WAIT、他の仕事で REQ に答えられない BUSY がある。現在 ACE のプロセッサモジュールでは、CA の決定用に 12 ビットのレグナイズレジスタ (RR) を設け、CD のアドレスの上位 12

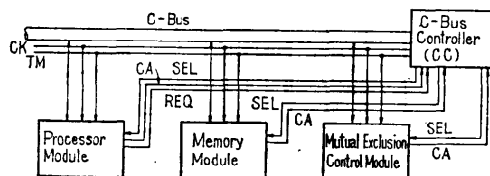


Fig. 1 Subsystem of ACE

* A Design of Mutual Exclusion Control Module for a Multiprocessor System by Tatsumi FURUYA, Hajime IIZUKA and Ryoichi OHOMOTE (Electric Computer Division, Electro-technical Laboratory)

** 電子技術総合研究所電子計算機部

ビットが RR にセットされた値と等しいとき YES, 等しくないとき NO としている。CC では、Cバス上の全モジュールの CA が YES か NO になったとき、通信を行うモジュールを決定し、それらに SEL を送る。ここで通信を行うモジュールの決定は、REQ がライト要求のときは、CA に YES を答えた全モジュールに SEL を出してブロードキャスト方式のライトを行い、リード要求で複数のモジュールが YES を答えたときは、このうち最も優先度の高いモジュールを選ぶ。このようにして通信経路が決定した後は、データが CD で指定される語数だけトランスミット (TM) 信号に同期してバースト転送される。

3. MECM の機構

この章では、MECM の構成と基本動作を示す。MECM は、独立したモジュールとしてバスに結合され、バス上の他のモジュールは、これに対して、システムの通信方式に則った以下に示す2つの基本命令を与えることにより相互排斥を実現できる。一般に相互排斥を達成するためには、共有資源に対応させてロックバイトやセマホアと呼ばれる物理的実体を使うが、MECM では、各共有資源に対応して、ロックセル (LC) と呼ぶセルを用意している。Fig. 2 が MECM の構成であり、LC の集合であるメモリと、制御回路、加算回路よりなる。LC は、共有資源の使用状態を示すテストアンドセット・ビット (T & S) と、待ち行列を管理するサイクリックカウンタ (CTR) よりなり、各セルにはそれぞれアドレスが与えられている。この度のインプリメントでは、LC のアドレスとして、Fig. 3 に示すアドレスのビット 2~7 を割当てている。

MECM への基本命令と、それに対する MECM の反応は次のとおりである。(Fig. 3 参照)

- ① リード; ある LC に対するリード命令
 - アドレスされた LC の内容が Cバスに読み出され、その後 LC の CTR には 1 が加えられ、

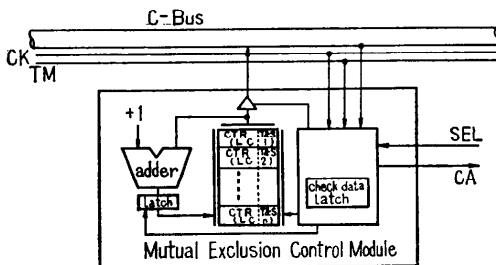


Fig. 2 Mutual Exclusion Control (MECM)

T & S はセットされる。

- ② ライト; ある LC に対するライト命令
 - アドレスされた LC の T & S は共有資源を待っているものがないとき、リセットされる。

なお、T & S ビットで共有資源を待っているものがあるか無いかの識別には、リード命令で +1 され、ライト命令で -1 される CTR とは別のカウンタ (メモリ) を用いている。

4. 相互排斥の実現

MECM を用いて相互排斥を実現する方法は次のとおりである。共有資源を使用したいモジュールは、まず、共有資源に対応付けた LC にリード命令を出し LC の内容をモジュール内に読み込む。そして T & S がリセットされている場合は、共有資源が空き状態であると解釈し、共有資源を使用する。なおリード命令では、3. で述べたように、読み出し終了後 CTR は +1 され、T & S はセットされるため、使用が終了しないうちに他のモジュールが同じ LC にリード命令を出しても T & S を調べると使用中であることがわかる。

リード命令の結果 T & S がセットされていた場合は、CTR の値を格納し、自分が資源を使用できるようになったとき、それをすぐに認識できるよう準備をして待ちに入る。ACE プロセッサモジュールの場合では、RR に、Fig. 3 のアドレスのビット 0~11 に対応するデータ、即ちリードで読み込んだ LC の番号と CTR の値をセットして待つ。もちろんこの待ちの間モジュールは、他の仕事を行うことも可能である。またプロセッサモジュールが RR に認識用データをセットしている間に共有資源を使用していたモジュールがライト命令を出す恐れがある場合は、一度 CA に WAIT または BUSY を出すようにしておき準備完了後 YES を返す方法や資源使用終了後のライト命令で

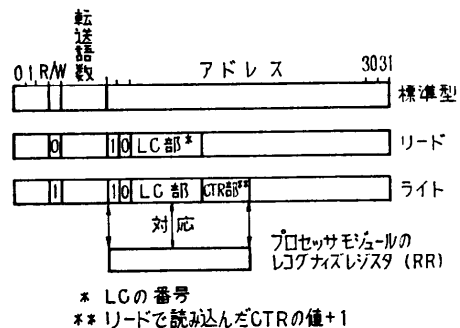


Fig. 3 Check Data and Bit Assignment for MECM

ダミーのデータを何語かライトすることにより時間をかせぐ方法を用いる。

共有資源の使用を終えたモジュールは、対応した LC にライト命令を出す。この時ライトアドレスとして、Fig. 3 のように LC 部には LC の番号を、CTR 部には、自分がリード命令を出したとき得た LC の CTR の値に 1 を加えたものをセットする。このライト命令により、MECM は、アドレスのビット 0.1 を見て、CA に YES を返し、資源を使用する番になったモジュールは、ビット 0~11 が自分の RR と一致することで、これを認知できる。そして自分の使用順番が廻って来たモジュールは、資源を使用する。以上が相互排斥の実現方法であるが、CTR のビット巾は有限であるため、modulo 2^n (n はカウンタのビット数) でサイクリックに動作するため、所定の個数以上が待ちに入ることは許されない。今回は、カウンタのビット巾を 4 にしている。

5. むすび

MECM は、待ちに入る数の制限や故障対策などの点で十分とはいえないが、共有資源を有するマルチプロセッサシステムの相互排斥に必要な機能を簡単なハードウェアで実現する一方法を提案している。

現在 ACE に結合されているものは、LC が 64 個、CTR のビット数が 4 で、IC の数は 28 個である。ME-

CM が相互排斥を効率良く実現していることが ACE のブロードキャスト式通信方式による点は大きい。このモジュールではライト命令のアドレスの CTR 部で次に共有資源を使用するモジュールを指定しているが、これを用いず、ライト命令のライトデータとしてこれを送り、資源を待っている全モジュールが、その値から自分の番か否かを決定するようにすれば、割込み機能のあるシステムにはこの考え方が適用できる。

最後に、本研究の機会を与えられた当研究所黒川、西野、石井の各部長、並びに本研究に関して熱心に御討論いただいた計算機方式研究室の皆様に感謝の意を表します。

参 考 文 献

- 1) L.C. Widdoes, Jr.: THE MINERVA MULTI-PROCESSOR, 3rd Annual Symposium on Computer Architecture. (Jan. 1976).
- 2) H. Iizuka et al.: ACE...A New Modular Computer Architecture, Proc. 2nd US-J Comp. Conf. (Aug. 1975)
- 3) E. W. Dijkstra: The structure of the multi-programming system, CACM, Vol. 11, No. 5, (May 1968).
- 4) 電総研: C-BUS SPECIFICATION, 内部レポート, (May 1974).

(昭和 51 年 8 月 12 日受付)

(昭和 51 年 9 月 24 日再受付)