

共有メッセージバスを持つマルチプロセッサシステムのデバッガ

永松 礼夫、漆原 茂、瀬部 昇、森下 巖

東京大学 工学部 計数工学科

ローカル・メモリを中心に計算の進行するようなマルチプロセッサシステムにおいて、全てのプロセス間の相互作用がメッセージでなされる場合を考える。そのようなシステムでは、各プロセスの入出力とプロセス内部の状態からシステムの全活動が把握出来る。多くのプロセスからなるタスクのデバッグには特定のメッセージごとに全プロセスを止め、内部を確認して、再開するのが有効である。デバッグによってメッセージの順序が変化しないよう（ユーザから透明）にプロセスを停止／再開させるためのハードウェアを試作した。更に、それを用いたデバッガ・プログラムの実験もおこなった。

"Debug Support for Multiprocessor System with Common Message Bus"

(in Japanese)

Leo NAGHAMATSU Sigeru URUSIBARA Noboru SEBE Iwao MORISHITA

Department of mathematical engineering and instrumentation physics,
University of Tokyo

(7-3-1 Bunkyo-ku, Tokyo 113, Japan)

There is a class of multiprocessor system in which all interaction between processes are message passing. On such system, behavior of a task (set of processes) can be traced with recording all messages. It's a powerful method for debugging to freeze up whole system with specified message arrival, check status of processes , and restart them. We implement support hardware to interrupt and continue execution of processes without disturbing message sequence. And also debug utility software is implemented on this hardware.

1. はじめに

我々の研究室では68000CPUをもつコンピュータボードを複数台接続した図1のようなマルチプロセッサシステムの試作をおこなった(文献1)。本稿では、そのシステム上でマルチプロセスのプログラムを開発する際のデバッグ操作を補助するために各プロセス間で通信されるメッセージの記録あるいは履歴を保存し、任意のメッセージパターンの出現と同時にシステム全体に割り込みをかけ、各CPUをデバッグモードに移行させた上で各プロセスの内部状態の観察および変更を行い、しかる後プロセスを再スタートさせるユーティリティについて報告する。

2. メッセージバスのハードウェア

システムは図1に示す様な構造を持っており、二つの共有バスがある。Dバスと呼ぶものは共有メモリを用いてプロセッサ間の通信をおこなうためのもので通常のVMEバスである。主にプログラムのロード等に用いる。もうひとつのMバスはプロセス間の短いメッセージの通信への応用を想定したもので、バスネットワークが生じにくいよう16ビットのワードを125nSサイクルで転送する。各CPUモジュールはCPU・ローカルメモリ・Mバスインタフェースよりなる。

つぎの節で述べるようにメッセージはサーバ・プロセスへの要求と応答として用いられることを想定して、ハード的にメッセージにrequest/replyの型をつけることが出来る。ハードウェアを簡略にするため各Mバスインタフェースは送信・受信用に一つずつのバッファのみを持っている。そのため、同一の送り先へのメッセージが短い時間のうちに重なると二つめのメッセージはバッファが使用中であるという理由でリジェクトされる。

送信をするにはインタフェースにメッセージの内容を書き込むことから始める。つぎに、メッセージの型を指示し、長さをコントロール・レジスタに書くと送信が自動的に開始される。送信インタフェースはMバスの使用权を得るまで待ち、データを送出して、結果(リジェクトか否か)をステータス・レジスタに格納したのちCPUに割り込みで通知する(図2)。Mバスの送信権は巡回するトークンを用いることで各CPUモジュールが公平となるよう設計してある。

メッセージを受信した場合も割り込みによって受信インタフェースからCPUに通知がなされる。なお、割り込みの優先度は受信が高くなっている。

3. カーネルの構造

前の節でのべた様なハードウェア上にクライアント・サーバ関係に基づいたOS核を作成した(文献2)。OS核は各CPUモジュール内でのプロセスの管

理を行うローカル・プロセス・マネージャや送信／受信割り込みの処理ルーチンなどからなっている。通信はブロッキング方式で、送信 (request) を行ったプロセスは応答 (reply) が返るまでスケジュールからはずされる。つまり、送信処理のなかでプロセスの切り替えがおきる (図3)。

4. デバッグ支援系的方式

本システムでは、すべてのメッセージがMバスを経由してやり取りされる。Mバス上では同時にひとつのメッセージが処理されるだけなので、全てのメッセージに順序関係をつけることが出来る。また、システムのソフトウェアのモデルはプロセス間の相互作用はメッセージのみに依るとしているので、計算の進行をメッセージの記録で示すことが出来る (図4)。

ここで、プログラマの予期しない結果が出たとき、誤りのありかを切り分けるにはある時点 (図の点線) ですべてのプロセスを同時に止め、内部の状態を調べるのが有効である。

5. 試作したデバッグ支援回路

VME規格のユニバーサル・ボードの上に約50個のTTLとメモリを用いて、Mバス上を転送される全てのデータ・ワードの記録、特定のデータの含まれるメッセージの検出、メッセージを計数しその回数によってCPUを「デバッグ状態」にするための割り込み発生などの機能をもつ「デバッガ・ボード」を作成した。図5にブロック図を示す。

6. デバッガのソフトウェアの構成

ソフトウェアはマルチプロセッサ本体とマンマシンインタフェースを担当するホスト計算機のうえにおかれる。両者は通信回線で結ばれ、「実行状態」の時 (図6) と「デバッグ状態」の時 (図7) では異なるサーバがホストと交信をする。「実行状態」中でも共有メモリ空間を用いたアクセスによってホスト側からデバッグを起動することも可能である (図6の点線)。

7. タイミングに関する問題点

デバッガ回路がデバッグ状態に移行するための割り込み (debug-in) を発生するとき、各CPUは同時に割り込みを受け付けることが望ましい。また、デバッグ状態からの復帰 (debug-out) も同時であれば、デバッグ状態にあった期間は各プロセスの相対関係になんら影響しない (図8)。この操作がデバッグの対象となるプロセスから透明であるには、厳密にはシステム・クロック単位で停止／再開を制御する必要がある。しかし、割り込みはその定義からして命令の区切りでかかるものである。更に、カーネル内部のコードのように割り

込み禁止の部分があるとより多くの時間ずれが生ずる（図9）。これらについて現在のインプリメントでは特に配慮をしていない。

8. システムにおけるデバッグ状態の位置づけ

現在のインプリメントではデバッガはカーネルと同じく、プロセスの管理の下にはない。割り込みへの高速な応答を保証するには、プロセス管理の共有構造がロックされている最中でもより高い優先度で応答する必要があるからである。そのため、デバッグ状態に入ったときのプロセス管理構造は整合のない状態で有り得るのでプロセス・コントロール・ブロックの書き換えなど幾つかの操作には制約が生じる。また、Mバス・インタフェースには同時にひとつのメッセージを保存するだけしかハードウェアバッファがないので、デバッグ状態の最中ではCPUモジュール間の通信にMバスそのままでは使えない。Mバスのハードウェアの内部状態の保存／回復を `debug-in/out` の時におこなえばよいが、2節でのべたようにもMバスの起動には何段階かのステップがあり、その何処でデバッグ移行の割り込みがおこったかを保持する必要がある。

9. まとめ

システムに同時に割り込みをかけて状態を凍結する方式のデバッガの有用性が確認された。本デバッガでは、システムのすべてのCPUモジュールがデバッグを補助するルーチンへ切り替わるので、本質的にマルチ・ユーザの環境でひとりのユーザに関係するプロセス群のみをデバッグしたいといった要求には答えられない。

今後の課題としては、シンボリック・デバッガの組み込みなどユーザ・インタフェースの向上や使用経験の蓄積があげられる。

参考文献

[1] 渡、吉田、永松、森下：「メッセージ／データ通信用2重共有バス型マルチプロセッサシステム」 第32回情報処理学会全国大会論文集5Q-6（1986.3）

[2] 永松、森下：「マルチプロセッサ間でのメッセージ通信オーバーヘッドの短縮について」 第33回情報処理学会全国大会論文集4C-10（1986.10）

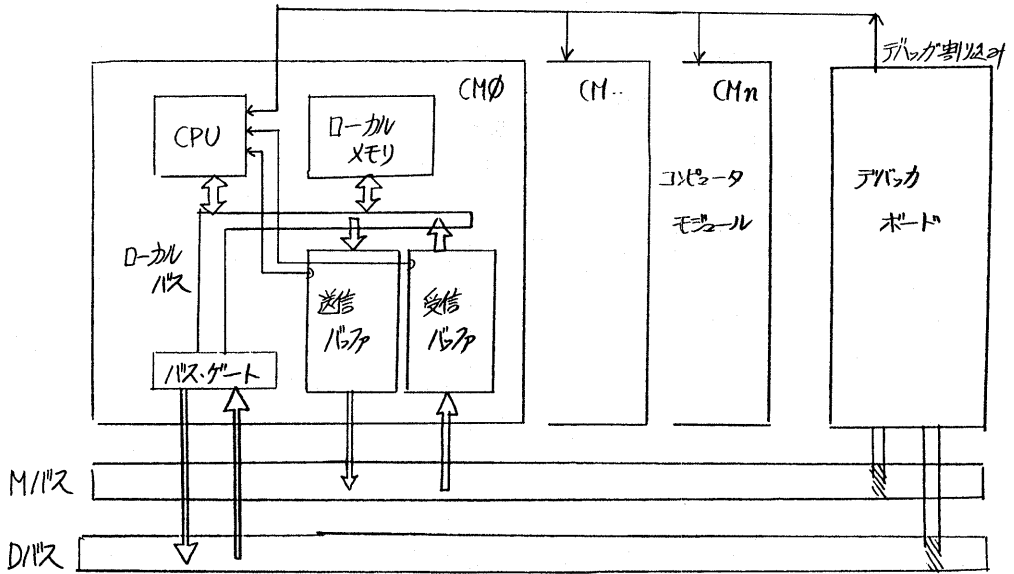


図1 マルチプロセッサシステムの構成

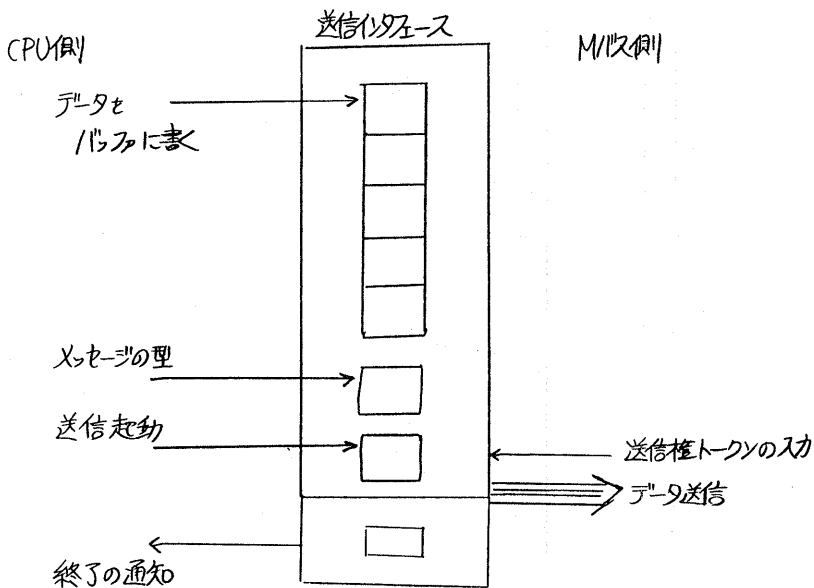


図2 メッセージ送信の手順

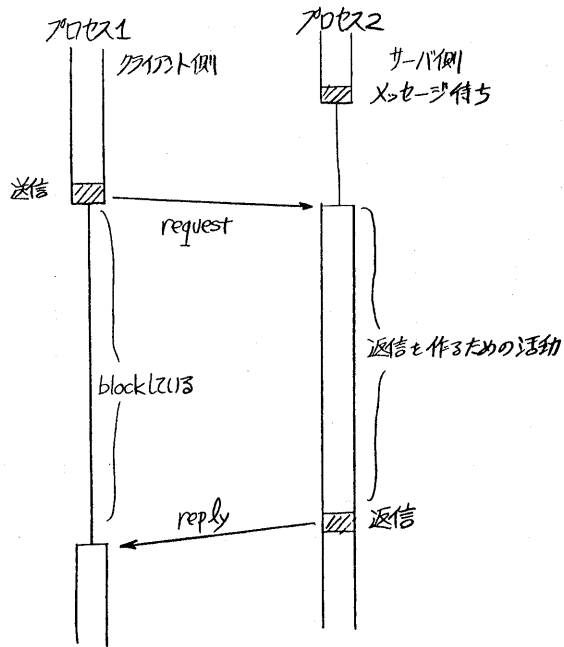


図3：ブロッキング通信

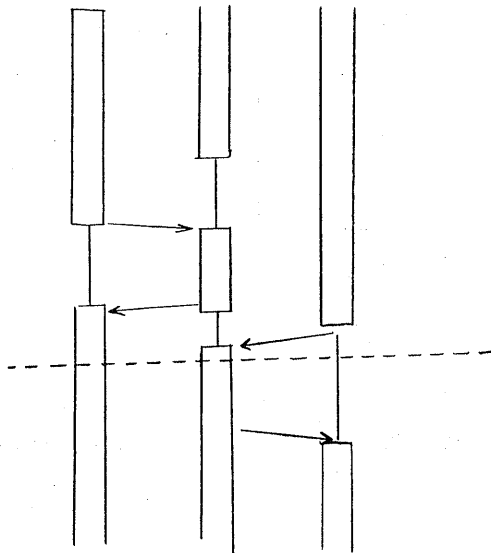


図4 メッセージの記録と計算の進行

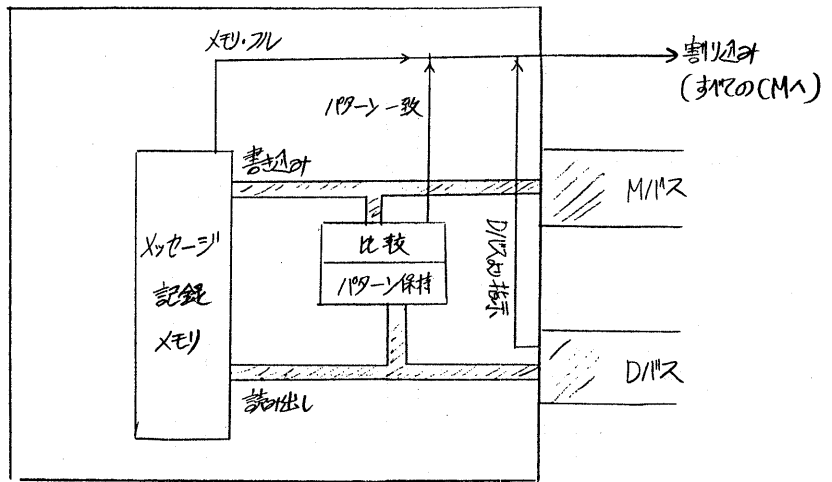


図5 デバッガ・ボードの構成

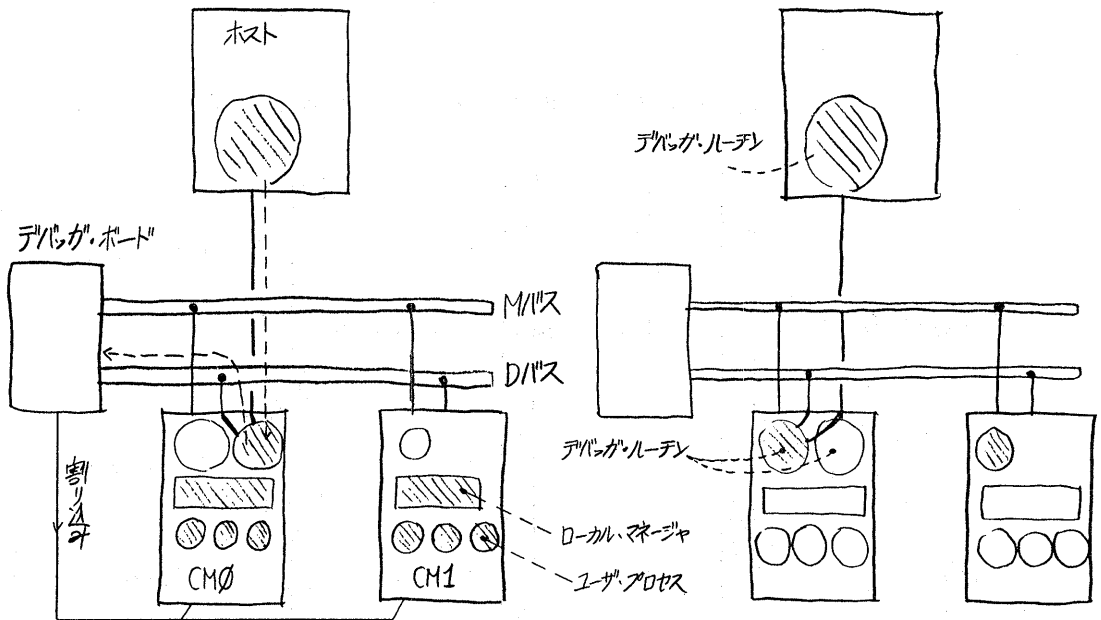


図6 実行状態

図7 デバッグ状態

のルーチンが活動している

