2H−4

PCIカードを用いた共有メモリ制御モジュールの設計

大濱 智宏† 久家 裕司† 田中 康一郎‡ 有田 五次郎†
†九州工業大学 情報工学部
‡九州工業大学 マイクロ化総合技術センター

1 はじめに

メモリ共有型システムは,以下の理由により期待が高い.(1) 細粒度の並列処理が可能 (2) データー貫性保持の負担が軽い (3) 逐次プログラムから移行の手間が少ない.しかし一般にメモリ共有型システムは特殊なハードウェア構成であるためにそれを自由に拡張することは困難である.

本研究では,高拡張性を持つ分散共有メモリシステムの構築を目的としている.その評価環境として,システム規模 (ノード数) を自由に調節するためにPC(*Personal Computer*) をノードとみなし,自製PCI(*Peripheral Component Interconnect*) カードを用いてノードを互いに接続することで分散共有メモリシステムを実現する.

本稿では,対象とする並列計算機の構成を述べ,それを実現するための共有メモリモジュールの詳細について言及する.

2 並列計算機の構成

既存のメモリ共有型並列計算機は,ハードウェアの拡張を行うようなものには対応していないためそれを用いて,ハードウェアの拡張を行うのは困難である.

そこで既存のコンピュータを用いて,内部システム構成の変更によるシステム拡張を行うのではなく,付加モジュールでメモリ共有を実現するためのハードウェアを構成することで,システム構築・拡張が容易な共有メモリ型並列計算機を構成することを目的とする.

しかし,大規模なメモリ共有型並列計算機を構成する場合には,そのメモリアクセス競合が大きな問題となる.そこで本研究では,各ノードのメモリを分散共有メモリとして扱い,この問題に対処する.この場合,共有メモリをどのように扱うかという問題がある.

図 1にその共有メモリ空間の構成を示す.(a) ノードに関係なく連続する物理メモリ空間として扱う,(b) ノード毎に分割して非連続の物理メモリ空間として扱う,といった2つのアプローチがある.(a) の場合,全共有メモリ空間を1次元の物理メモリ空間としてそのまま認識することが出来るが,各ノードから見た共有メモリ空間の物理アドレスが異なるため,他ノード共有メモリ上のデータを直接アクセスすることが出来ない.(b) の場合,他ノードにある共有メモリを指定するためには,1 次元のアドレス指定ではなく,そのノードを示すアドレス情報と,ノード上のメモリアドレス情報の 2 つを用いて 2 次元のアドレス指定を行なう必要があるが,任意の共有メモリ空間の物理アドレスは一意に決定される.(a) の方法は,他ノードのデータを取得する際にそのノードの CPU にデータのアドレスを問い合わせなければならないため,レスポンスが低下する.(b) の方法は,2 次元アドレス指定なのでメモリの拡張性が高く,共有メモリは全ノードから直接アクセスすることができる.以上の理由により,(b) のアプローチを用いる.

3 PCIカード概略

図 2に PCI カード構成を示す [1]. PCI カード上には,拡張メモリであり共有メモリとして使用される DIMM(*Dual Inline Memory Module*) を搭載している.さらに,

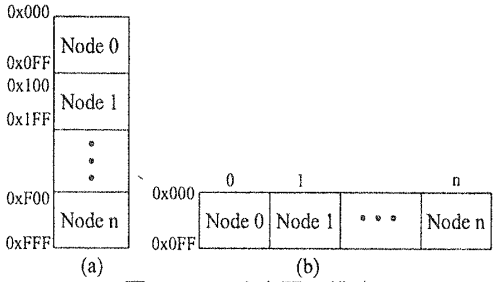

図 1: メモリ空間の構成

その DIMM をコントロールするための Xilinx 社の FPGA(*Field Programmable Gate Array*) である XC4028XL と,カード全体の制御を行うための PCI コントローラである AMCC 社の S5933 を搭載している.付加モジュールと

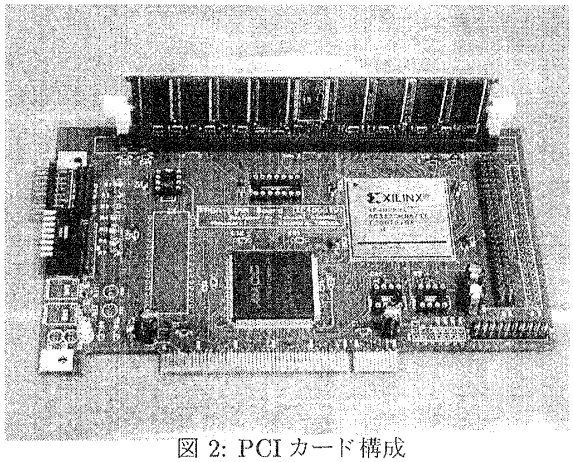
図 2: PCI カード構成

してメモリ管理モジュールを構成するが,共有メモリ計算機がその性能を発揮するには十分なデータ転送速度が必要とされる.そのために高速なデータ転送が可能である PCI バスをターゲットとしている.さらに,PCI コントローラは,自ノードからの要求の場合は PCI ターゲットとしての動作で十分であるが,他ノードからの要求がメインメモリへのアクセス要求である場合は PCI バスマスタとして動作しなければならないために,バスマスタとしての動作が可能である S5933 を用いている.ネットワークにも高速なデータ転送が必要とされるため,新たに通信用モジュールを付加することなども考えられる.それらの付加回路が今後追加された場合にも柔軟にその制御を行うモジュールを再構成するために,FPGA を用いている.さらに,大規模な物理メモリ空間を構成することを目的としているので,高拡張性を持つ DIMM を拡張メモリとして用いている.

最終的に,ローカルメモリ,拡張メモリ共に全ノードで全メモリを共有するシステムを調査することを目的としており,その調査の第一段階として PCI カード上の拡張メモリのみを分散共有メモリとする (図 3).

PCI カード上のメモリを共有メモリとしてシステムに認識させ,そのアクセスを制御する機構はデバイスドライバで行う.デバイスドライバの作成のためには,開発環境が整っている Windows NT 4.0 Workstation を OS として用いている.デバイスドライバの開発ツールとして ToolCraft 社の WinDK を用い, FPGA 設計には 設計言語に Verilog HDL ,論理合成ツールには Synopsys 社の FPGA Compiler, 自

Design of Shared Memory Module on the PCI Card. By Tomohiro Oohama†, Yuji Kuge†, Koichiro Tanaka‡, Itsujiro Arita† (†Department of Artificial Intelligence, Kyushu Institute of Technology. ‡Center for Microelectronic Systems, Kyushu Institute of Technology., 680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan)

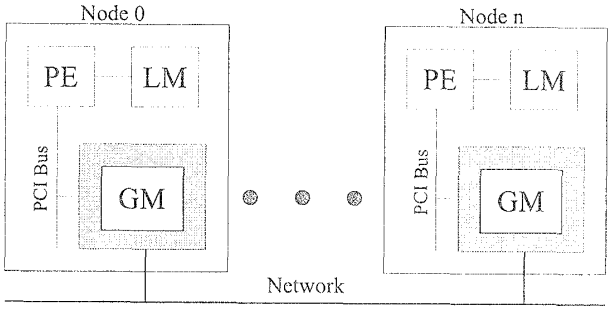

図 3: メモリ共有型並列計算機の構成

動配置配線ツールには Xilinx 社の Design Manager を用いている.

4 共有メモリ制御モジュール

4.1 デバイスドライバ

一般的な OS では, ユーザプログラムはユーザモードで動作しているため, ユーザプログラムがハードウェアにアクセスする際には, カーネルモードで稼働し, アプリケーションとハードウェアの仲介として機能するデバイスドライバが必要となる [2]. 図 4に共有メモリを管理するデバイスドライバの概要を示す.

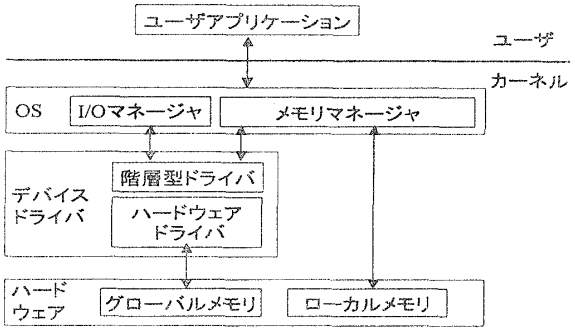

図 4: 共有メモリ管理デバイスドライバ

階層型ドライバはアプリケーションのメモリ要求が共有メモリへの要求であった場合に起動し, ハードウェアドライバを制御することで共有メモリをアクセスする. この場合アクセスする共有メモリが自ノード, 他ノードであるかをユーザアプリケーションから隠蔽することができる.

4.2 FPGA

実際の共有メモリへのアクセスは,FPGA で設計するメモリアクセスモジュール [3] によって行われる. デバイスドライバによってメモリアクセス要求は 2 次元アドレス情報に変換されており, メモリアクセスモジュールはその情報を受け取り, そのアドレス情報が自ノードである場合は拡張メモリへアクセスし, 他ノードである場合はネットワークを介してそのノードへアドレス情報と要求を送る.

4.3 PCI コントローラ

共有メモリアクセスに必要な情報は,(1) ノードアドレス,(2) メモリアドレス, (3) 要求種別の 3 種類である. この情報をメモリアクセスモジュールに送る方法として,S5933 には Mailbox,Pass-Thru,FIFO といった 3 つの方式がある [4]. Mailbox,Pass-Thru はターゲットとして動作する場合の転送方式であり,FIFO はバスマスタとして動作する場合の転送方式である.

5 検証

Mailbox は 32 ビットデータをやり取りできるがレジスタが入出力 4 個づつとなっているため, 大量のデータ転送には向かない. Pass-Thru はメモリアクセスのための機能であり, アドレス情報, データ情報, リードライト要求を送ることができ, バースト転送にも対応している. このため通常のメモリアクセスの場合は Pass-Thru だけで十分であるが, 本研究ではノードアドレス情報も必要である. ノード情報を送る方法としては,(1)Pass-Thru の最初のデータをノード情報と見なす,(2)Mailbox を用いてノード情報を別に転送する, といった方法が考えられる. メモリ要求が他ノードのデータに対する要求である場合, そのノードへアドレス情報, データ情報を転送するには, ターゲットノードが要求を受け付ける準備ができるまで待たなければならない.(1) の方式はデータの転送が始まると途中で待たせることができないので, この処理には向いていないものと思われる. そこで, 本研究では (2) の方式を選択する.

図 5に本研究で用いる 2 次元メモリアドレス指定時の動作を示す. 今回は Mailbox,Pass-Thru を組み合わせた動作に対応できる回路を設計し, 一台の PC 上で仮のノード情報によって, 仮のアドレス情報, データ情報を出力する回路を作成し, 動作を行うことができることを確認した.

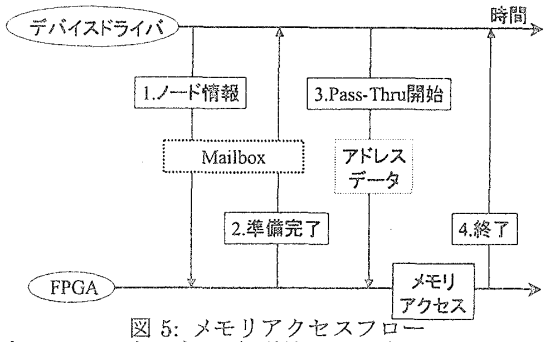

図 5: メモリアクセスフロー

今回はシングルデータ転送機構を構成した. この構成では,1,2 の Mailbox の部分だけで 22 クロック必要であった.3,4 の Pass-Thru 部分はライト時のアドレスとデータ情報の受け渡しに 4 クロック, リード時は 5 クロックでシングルデータ転送は可能である.

6 おわりに

本稿では, 我々が目的とする共有メモリ型並列計算機モデルについて示し, その実現のためのモジュールである PCI カード, 共有メモリアクセス機構について説明した. 設計結果より, 共有メモリアクセスのための 2 次元アドレス方式が Mailbox,Pass-Thru を用いることで実現可能であることを確認できた.

謝辞

デバイスドライバ開発に尽力して下さった本学知能情報工学科立川純氏に感謝します.

参考文献

[1] 田中康一郎, 平野孝明, 浅野種正, 有田五次郎 : システム LSI 時代に向けた FPGA と DSP によるシステム設計教育, in *Proceedings of The Seventh Japanese FPGA/PLD Design Conference & Exhibit*, pp. 53 – 60 (1999).

[2] 立川純, 林悠平, 大濱智宏, 田中康一郎, 有田五次郎 : 分散共有メモリモジュールのためのデバイスドライバの作製, 若手の会セミナー講演論文集, pp. 17 – 18 (1999).

[3] 久家裕司, 平野孝明, 大濱智宏, 田中康一郎, 有田五次郎 : FPGA/DIMM を実装した PCI カードを用いた共有メモリモジュールの設計, 若手の会セミナー講演論文集, pp. 15 – 16 (1999).

[4] Applied Micro Circuits Corp.: *PCI PRODUCTS DATA BOOK* (1998).