

## マルチプロセッサシステムにおける高速 DRAM の性能評価

上山智義 綾田和晶 西村浩二† 相原玲二†

ueyama@ipc.hiroshima-u.ac.jp

広島大学 大学院 工学研究科

〒739 広島県東広島市鏡山一丁目4番1号

† 広島大学 総合情報処理センター

〒739 広島県東広島市鏡山一丁目4番2号

高速 DRAM (Rambus DRAM、Synchronous DRAM、Cache DRAM) をメインメモリとした共有メモリ型マルチプロセッサシステムを構成し、ベンチマークテストにより各高速 DRAM の速度性能について性能評価、比較を行なった。すべてのシステムは、ハードウェア記述言語 VHDL を用いてモデル化し、そのほとんどを機能記述により行なった。本稿では、まず高速 DRAM のアーキテクチャ、システム構成について説明する。モデルのベンチマークシミュレーション結果を示し、各高速 DRAM の評価結果を述べる。

## PERFORMANCE EVALUATION OF FAST DRAMS ON MULTIPROCESSOR SYSTEMS

Tomoyoshi Ueyama, Kazuaki Ayada, Kouji Nishimura† and Reiji Aibara†

Graduate School of Engineering, Hiroshima University

1-4-1, Kagamiyama, Higashi-Hiroshima, 739 Japan

† Information Processing Center, Hiroshima University

1-4-2, Kagamiyama, Higashi-Hiroshima, 739 Japan

We evaluated performance of fast DRAMs (Rambus DRAM, Synchronous DRAM and Cache DRAM) using several benchmarks on shared-memory, shared-bus multiprocessor systems. The all systems are described in hardware description language VHDL. In this paper, we explain architecture of the fast DRAMs and the systems using them. We demonstrate results of the simulation and the performance evaluation.

## 1 はじめに

プロセッサはLSI技術の進歩に伴い年々高機能化、多機能化している。また、メモリに関してもその集積度は著しく向上した。しかし、速度性能はプロセッサの動作周波数が飛躍的に向上しているのに対し、メインメモリとして使われるDRAMの動作速度がそれほど改善されていない。このボトルネックを解消するため、プロセッサとDRAMの間に二次キャッシュを置く方法が広く用いられてきた。しかし、キャッシュメモリとしての高価なSRAMの使用やボード面積の増加など問題が多い。そのため、近年高速DRAMの研究開発が盛んに行なわれている。1992～1993年にかけて、Rambus DRAM (RDRAM)、Synchronous DRAM (SDRAM)、Cache DRAM (CDRAM) というバースト転送型の高速DRAMが発表された[1]。いずれも高速な外部クロックに同期してデータ転送を高速に行なう。これらのDRAMでは、メモリセルへのアクセス時間は既存のものと同程度であるが、メモリアーキテクチャを工夫することにより、このチップ全体としてアクセス時間の短縮、データ転送速度の高速化を図っている。

また一方で、複数個のプロセッサを同時に動作させて並列処理を行なうことにより高機能化を図るマルチプロセッサシステムが注目されている。共有メモリ型マルチプロセッサシステム[2][3]は汎用の計算機システムにも浸透しつつある。単一のバスを用いるとバス上では単一の通信しか行なうことはできないが、この方法は実用的である。しかし、プロセッサはバスの使用をめぐる競合し、このバス競合による性能低下が問題となる。速度向上のため、各プロセッサにローカルなキャッシュを持たせて、バスに接続されたメモリへの参照回数を減らすなどが、前述のように通常のDRAMはアクセス速度が遅いため、マルチプロセッサにおいては一層問題となる。マルチプロセッサシステムにおける高速DRAMのベンチマークテストによる定量的評価報告はほとんどない。

本研究では、各高速DRAMをメインメモリとした単一バス共有メモリ型マルチプロセッサシステムを構成し、ベンチマークテストにより各高速DRAMの速度性能についての性能評価、比較を行なった。すべてのシステムは、ハードウェア記述言語VHDL (VHSIC Hardware Description Language)[4][5]を用いてモデル化し、そのほとんどはVHDLの機能記述が使用されている。通常、VHDLは大規模チップの論理設計に用いられ、シミュレーションは設計の検証のために行なわれるが、本研究では、VHDLはシステムのモデル化に用い、シミュレーションはシステムの定量的性能評価のために行なった。論理シミュレータ上でいくつかのベンチマークプログラムのシミュレーションを行ない論理的な実行時間を得た。こうして得たシミュレーション結果を分析し、各高速DRAMの性能について考察する。

## 2 アーキテクチャ

### 2.1 高速DRAM

高速DRAMのアーキテクチャ概要を図1～3に示す。これらの高速DRAMはいずれも外部クロックに同期した方式をとっている。メモリセルアレイは既存の汎用標準DRAMと変わらないが、周辺回路は工夫が凝らされておりそれぞれで違った特徴を持っている。

RDRAM[6]は、ラムバス社により開発されたDRAMでラムバスチャネルと呼ぶ9ビットバス(1bitはパリティビット)を250MHzのクロックの立ち上がり立ち下りのエッジを用いて、500MByte/sのデータ転送速度を実現する。RDRAMへのアクセスの方法は、既存の汎用標準DRAMとは異なり、専用のプロトコルを定義しラムバスチャネルを介してパケットを交換する方式である。パケットにはリクエスト、アクノレッジ、データパケットの3種類がある。マスター(CPU側)から命令(読み出し/書き込み)やデータの情報を持ったリクエストパケットをスレーブ(RDRAM)に送り、これをスレーブが解読しアクノレッジパケット(Ack、Nack)を返す。RDRAMでは、センスアンプをキャッシュとして用いており、Ackはセンスアンプキャッシュに所望の列データがある場合(ヒット)で、行アクセスを開始する。Nack(ミス)の場合は、現在の列データをメモリセルに書き戻し、アクセスされた列アドレスのデータをセンスアンプキャッシュに読み出す。

SDRAM[7]は、3種類の高速DRAMで最も従来の汎用標準DRAMに近いアーキテクチャを有している。複数チップを並列に配置することによりバス幅を広くすることができ、例えば外部クロック100MHz、8ビットバスのチップ4枚構成の場合、400Mbyte/sのデータ転送速度となる。2つのバンクのインタリーブによって、行アドレスが異なる時も間断なくアクセスできる。なるべく汎用標準DRAMに近くなるように定義しているため、現システムのデザインを大きく変更することなく、高性能化できるのが特徴とされている。

CDRAM[8]は、1つのチップの上にSRAMとDRAMを集積したものである。SRAMはキャッシュとしてDRAMはメインメモリとして使われ階層化されたメモリ構成をしている。1チップ化したことでDRAMとSRAM間のバス幅を簡単に広げることができ、DRAMとSRAMの間のデータ転送が1サイクルのできるため、ブロックサイズを大きくしても、キャッシュミスの際のアクセスペナルティーを抑えられる。SRAMに所望のデータがある場合(ヒット)、16M CDRAMで10 nsのアクセス時間となる。SDRAMと同様に複数チップを並列に配置することによりバス幅を広くすることができる。ファーストコピーバック[9]と呼ぶ新書き込み方式が開発されており、従来のコピーバック方式の約1/3に短縮している。

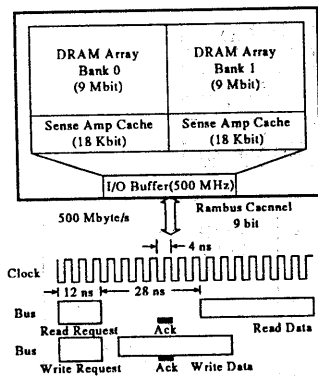


図 1 Rambus DRAM (RDRAM)

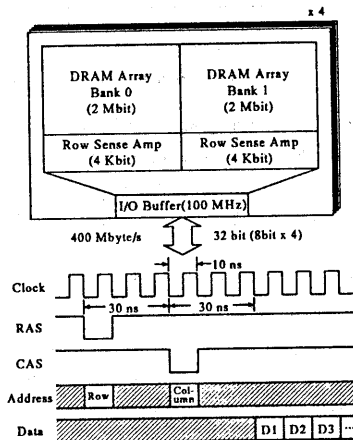


図 2 Synchronous DRAM (SDRAM)

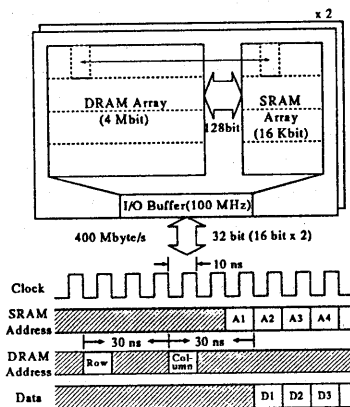


図 3 Cache DRAM (CDRAM)

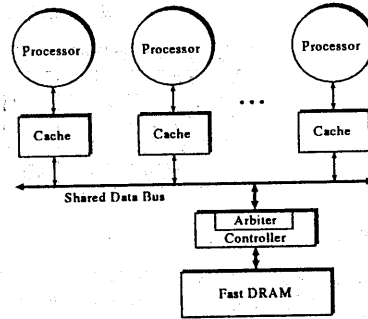


図 4 マルチプロセッサシステムモデルの構成

## 2.2 マルチプロセッサシステム構成

マルチプロセッサシステムの構成を図4のように、各プロセッサとメインメモリを単一バスで結合する共有メモリ型マルチプロセッサ方式とした。CPUは汎用RISCプロセッサであるmips社のR3000を採用した[10]。R3000は完全な32ビットRISC型CPUで、典型的なRISCアーキテクチャを有している。フェッチ、デコード、実行、メモリアクセス、書き込みの5段パイプライン、遅延ロード・遅延分岐等の特徴を持つ。共有されるメインメモリは高速DRAM(RDRAM等)である。各プロセッサは、1次キャッシュとして命令キャッシュとデータキャッシュを持っている。これはローカルなキャッシュとしてバス競合の低減の役割も果たす[11]。書き戻しの方式はコピーバック方式で、マッピング方式はダイレクトマッピングである。コントローラは、高速DRAMを制御するためのもので、RDRAMではリクエストパケットやデータパケットの生成を行ない、SDRAM/CDRAMではコマンド発行やキャッシュの管理などを行なう。調停回路(Arbitrer)は公平なバスの使用を目的とした巡回優先順位アルゴリズム[2]によって、1つのプロセッサにバスの使用許可を出す。キャッシュコヒーレンシ(一貫性)を保つためのキャッシュプロトコルには、Dragonプロトコル[3]を採用した。Dragonプロトコルは、更新型のプロトコルで、共有するブロックに対する書き込み時はシステム中のすべての共有ブロックを最新データで更新する。

## 3 ベンチマークテストによる性能評価

### 3.1 シミュレーションシステムの記述

システムのモデル化はすべてVHDLを用いて記述し、そのほとんどは機能記述が使用されている。まず、CPUのR3000を記述しその動作確認をした。それにはR3000を使用したワークステーション(NEC EWS4800/215)を用いた。すなわち、同じベンチマークプログラムをこのワークステーションとVHDL論理シミュレータで走らせ、両者を比較した。CPU以外は、実際のシステムがないため、シミュレータ上のシグナルの波形やベンチマーク

表 1 高速 DRAM 速度性能の仕様

高速 DRAM の種類		RDRAM	SDRAM	CDRAM
アクセス 時間	ヒット時	40 ns	30 ns	10 ns
	ミス時	116 ns	90 ns	60 ns
バス幅		9 bit	32 bit	32 bit
クロック		500 MHz	100 MHz	100 MHz
データ転送レート		500 Mbyte/s	400 Mbyte/s	400 Mbyte/s

表 2 ベンチマークプログラム

qsort	配列 300 個のデータのクイックソート
krnd	Kunuth の乱数発生法による 100 個の乱数発生
sum	配列 50 個の和

プログラムのトレースを行なうことによって動作の確認をした。

VHDL 論理シミュレータでは、多大なシミュレーション時間を必要とする。今回使用したシミュレータでは、特にメモリの容量がシミュレーション時間に大きく影響した。実際と同じメモリ容量をとるとシミュレーションに時間がかかり過ぎるため、できる限りメモリ容量を少なくし、ベンチマークも短いものに変更した。また、RDRAM/SDRAM のセンスアンプと CDRAM の SRAM の容量は、小さなベンチマークでも特徴が顕著に表れるよう実際の半分とした。1 次キャッシュの命令キャッシュ容量は 1024byte とし、ベンチマークプログラムはこの容量に収まる大きさとした。また、シミュレーションの開始時にはすでに命令キャッシュに入っている状態であるとする。すなわち、命令キャッシュでのキャッシュミスは起こらないため命令キャッシュからメインメモリのアクセスはなく、データキャッシュミスの時のみメインメモリにアクセスする。1 次キャッシュのデータキャッシュ容量は 512byte とかなり小さくすることで小さいプログラムでもキャッシュミスが頻繁に起こるようにした。CPU の動作周波数は 100MHz とした。

### 3.2 ベンチマークプログラム

使用したベンチマークプログラムはすべて整数演算である [12]。ベンチマークプログラムの概要を表 2 に示す。データがアクセスするメモリ領域は、データキャッシュの容量を越えるようにし、必ず書き戻しが起こるようにした。それぞれ違う特徴を持ったプログラムであるが、これは後でシミュレーション結果とともに述べる。

1 次キャッシュラインサイズ、プロセッサ数をパラメータとして行なった。ラインサイズはキャッシュが管理しているブロックで、この単位で 1 次キャッシュとメインメモリとのデータのバースト転送を行なう。

各 CPU はすべて同じベンチマークプログラムをシミュレーション開始と同時に実行する。CPU の数が N の場合は、N 個のプログラムを処理することになる。各 CPU

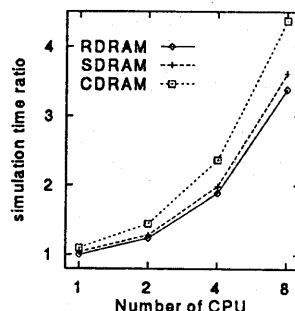


図 5 シミュレーション結果 (qsort, line size 128byte)

が扱うデータも同じ値であるがデータ領域はすべて異なる。すなわち、各 1 次キャッシュは同じブロックのデータを保持することはない。これによって、Dragon プロトコルによる 1 次キャッシュ間のデータのやりとりは発生しない。

RDRAM/SDRAM では 2 バンク構成、CDRAM では 4 ウエイの SRAM となっているが、一つのバンクまたはウエイのみしか使用しない。したがって、データがアクセスする領域は 1 バンクまたは 1 ウエイ内のみである。

シミュレーション結果は、それぞれある 1 つのシミュレーション時間を基準 (=1) として、これからの相対値で示す。

### 3.3 シミュレーション結果

共有メモリマルチプロセッサシステムでのバス競合は、CPU の台数が増えるにしたがって深刻化することが予想される [13]。ここで、CPU 数を変えた時のシミュレーションを行なった。qsort のシミュレーション結果を図 5 に示す。基準は、1 CPU の時のもっとも速かった高速 DRAM の結果である。CPU が増えるにつれシミュレーション時間が増していることがわかる。遅い DRAM (このシミュレーションでは CDRAM) は、CPU 増加に伴う速度低下は顕著になる。

図 6~11 に、4 CPU と 8 CPU の場合の各ベンチマークプログラムのシミュレーション結果を示す。基準は、ベンチマークプログラム各々で、4 CPU、ラインサイズ 16byte の時のもっとも速かった高速 DRAM の結果である。結果をみると 1 次キャッシュのラインサイズ (バースト転送するブロックのサイズ) によって、各ベンチマークプログラ

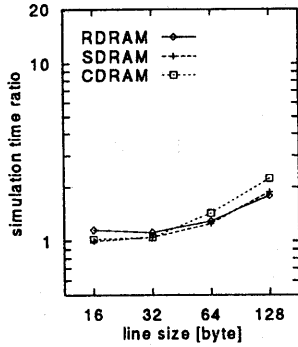


図 6 シミュレーション結果 (qsort, 4 CPU)

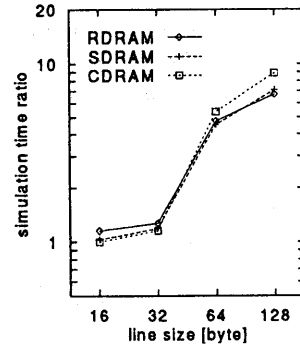


図 8 シミュレーション結果 (krnd, 4 CPU)

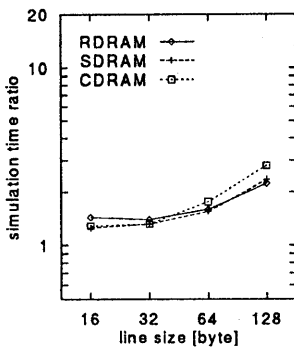


図 7 シミュレーション結果 (qsort, 8 CPU)

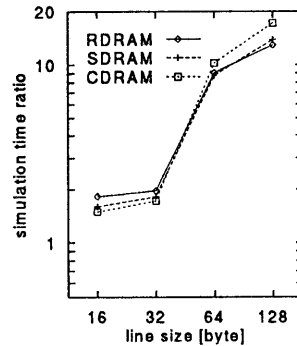


図 9 シミュレーション結果 (krnd, 8 CPU)

ムで特徴がでている。qsort と krnd ではラインサイズが大きくなるにしたがって遅くなっているが、理由は2つ考えられる。1つは、1次キャッシュ中に入る連続領域の数が少なくなっていくために、将来必要になるかもしれない情報が追い出されてしまうからである。128byteのラインサイズでは512byteのキャッシュ容量であるので4つの連続領域しか同時にキャッシュできない。もう1つは、一度にラインサイズを転送するバースト転送なので、転送する情報量は多くなるが、このライン中の不要な情報まで転送してしまうためである。krndでラインサイズが大きくなると極端に遅くなるのは、不要な情報を転送する割合が高いため、メモリ領域を分散的に参照するプログラムである。sumはラインサイズが64byteまで大きくなるにしたがって遅くなっているが、この理由は逐次的に参照されるようなプログラムであり、ブロックサイズが大きくなってもそこに含まれる情報が必要とされるからである。

CPU数の違いでは、先に述べたようにCPUの数が増えるとバス競合による性能低下が見られることとDRAMの性能による差がやや大きくなることが確認できた。

高速DRAMによる違いではベンチマークについては大きな差は見られないが、ラインサイズが小さい時はCDRAMが最も速く、大きい時にはRDRAMが速く、いずれの場合でも中間なのがSDRAMという傾向が見られることがわかった。RDRAMが、ラインサイズが小さい時に他に比べて遅いのはアクセス時間が長いため、ラインサイズが大きくなるとアクセス時間よりも転送時間が効いてきて最も転送速度の速いRDRAMが速くなる。CDRAMが、SDRAMと同じ転送速度にもかかわらずラインサイズの大きい時に遅いのは、チップ内のSRAMとDRAM間を転送されるブロックのサイズ(32byte)よりラインサイズの方が大きくなると、ファーストコピーバックの効果があまりないためである。ファーストコピーバックは、SRAMとDRAMの間にブロックサイズのデータ転送バッファを設け、書き戻し時に一時的にこのバッファに書き込み、その間に読み出しを行なうことにより時間短縮をしたものである[9]。

#### 4 おわりに

いくつかのベンチマークプログラムのシミュレーション結果により各高速DRAMの性能について定量的に述

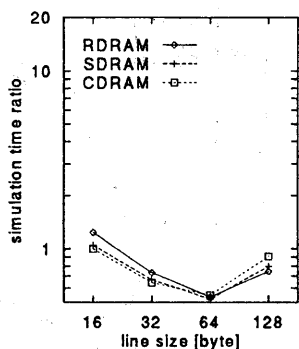


図 10 シミュレーション結果 (sum, 4 CPU)

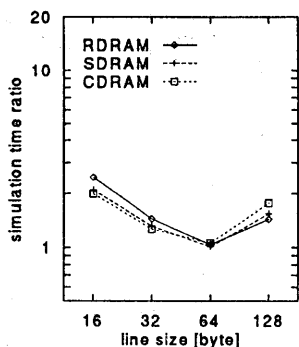


図 11 シミュレーション結果 (sum, 8 CPU)

べた。バス競合による性能の低下が引き起こされることを確認した。概してラインサイズが小さい時は CDRAM、大きい時は RDRAM の速度性能がよく、いずれでも中間的なのが SDRAM という結果を得た。今回使用したベンチマークでは、メモリサイズの制約等から、それほど大きな性能差は出ていないが、VHDL 等のハードウェア記述言語によるシステム記述およびシミュレーションが定量的評価に有効であることを示すことができた。

### 謝辞

本研究を進めるに当たり貴重な助言をいただいた東北大学工学部機械知能工学科の小柳光正教授、日本電気(株)マイクロエレクトロニクス研究所の高田正日出氏に感謝致します。

### 参考文献

- [1] “100MHz時代のCPUボード設計”, NIKKEI ELECTRONICS, No.556, pp.110-141, (Jun.1992).
- [2] B. ウイルキンソン (高橋義造 他訳): 計算機設計技法 マルチプロセッサシステム論, トップラン, (1994).

- [3] 鈴木, 清水, 山内: 共有記憶型並列システムの実装, コロナ社, (1993).
- [4] David R. Coelho: The VHDL Handbook, Kluwer Academic Publishers, (1989).
- [5] R. リブセット 他 (杉山尚志 他訳): VHDL 言語記述によるハードウェア設計のアプローチ, マグロウヒル, (1990).
- [6] Rambus Application Note Version 1.0, Rambus Inc. (Feb.1992).
- [7] 16M bit Synchronous DRAM preliminary data sheet Ver.4.0, NEC, (Mar.1994).
- [8] 4M CACHED DRAM WITH 16K SRAM data sheet, 三菱電機.
- [9] Dosaka, K. et al: A 100 MHz 4Mb Cache DRAM with Fast Copy-Back Scheme, IEEE J. Solid-State Circuit, Vol.27, pp.1534-1539, (Nov.1992).
- [10] V<sub>R</sub>3600A™ ユーザーズ・マニュアル ハードウェア編, NEC, 第3版, (1993).
- [11] 福村, 平野, 塩澤: “バス結合型マルチプロセッサのキャッシュ構成方式”, 信学論, Vol.J74-D-I, No.10, pp.721-728, (1991).
- [12] 奥村晴彦: C 言語による アルゴリズム辞典, 技術評論社, (1991).
- [13] 松永俊雄, 福村好美: “バス結合型マルチプロセッサ方式の性能評価”, 信学論, Vol.J73-D-I, No.9, pp.737-745, (1990).