Vol.2012-HPC-133 No.6
2012/3/26

Gather 機能を有する Hybrid Memory Cube の FPGA を用いた予備評価

田邊 昇[†] 堀 喬裕[††]
Boonyasitpichai Nuttapon[††] 中條 拓伯[††]

多くの重点アプリケーションの性能に直結する Byte/FLOP を現在より大幅に下げざるを得ない Exa FLOPS 級マシンにおいて，性能面と電力面の両面から，その対策が望まれている．本報告ではその対策として Gather 機能を有する Hybrid memory cube(HMC)について提案し，FPGA による予備評価を行った．その結果，三次元実装向けのパラメータ空間において設計指針の妥当性を概ね裏付ける結果が得られた．Gather 機能を HMC に実装することにより，配列の間接参照のスループットにおいて 11〜13 倍の性能向上が得られることが確認された．回路規模は現時点での最新より 2 世代前の中規模 FPGA で十分に実装できる範囲にあることを確認した．

Preliminary Evaluations for Hybrid Memory Cube with Gather Functions Using FPGA

Noboru Tanabe[†] Nobuhiro Hori[††]
Boonyasitpichai Nuttapon[††] and Hironori Nakajo[††]

In Exa FLOPS scale machines, Byte/FLOP ratio which decides performance of many focused applications must be degraded severely from now. Therefore, the solution for the problem from the standpoint of performance and power is desired. In this report, the authors propose Hybrid Memory Cube (HMC) with gather functions as a solution for it. It is preliminary evaluated using FPGA. We got results which support the validity of our design policy in the design parameter space for 3D stacking. We confirmed the acceleration ratio from 11 to 13 in the indirect array references can be obtained by implementing gather functions in an HMC. We confirmed the needed capacity of logic gates of the controller is enough with the middle-class FPGA of 2 generation older than that of the newest generation.

1. はじめに

2011 年に日本では文部科学省と国内のスーパーコンピュータセンターが音頭を取り，多様なシステム系研究者を巻き込んだ形で，2018 年頃のスーパーコンピュータとして Exa FLOPS 級の性能を有するシステムの検討[1]が行われた．そこでの電力制限やデバイス技術トレンドの分析から，現状をトレンドの線上に引き伸ばした Exa FLOPS 級マシンの Byte/FLOP 値は現状より 1/5 程度に低くしないと製造できない見込みが明らかになってきた．ユーザー候補であるアプリケーション作業部会の人々からは，多くの重点アプリケーションの性能に直結する Byte/FLOP 値の悪化に対する強い懸念と，システム側で何らかの有効な対策を講じるべきとの要望が上がっている．

米国においては Exa FLOPS マシンに関し 2008 年頃から検討結果 [2][3] [4]が見え始めており，Smart memory などの幾つかの技術に焦点が当てられた[5]．そのフィージビリティ研究予算の成果の一つに Hybrid memory cube[6][7][8] (以下 HMC とする)の開発がある．HMC は米国の Exa FLOPS マシン開発機関である IAA と Micron 社らが，米国の Exa FLOPS マシンのロードマップ[3]をとりまとめた P. M. Kogge 教授の指導の下で研究開発した成果である．HMC は，2011 年 8 月に Hot Chips においてプロトタイプ[7]が発表され，その電力節約とバンド幅向上の両面で革命的な技術として注目された．

電力問題と Memory wall 問題の同時解決は HPC 分野に限られた課題ではない．その解決に現時点で圧倒的な効果が認められている HMC が波及する分野は，組込み機器からクラウドに至るまで，広大である．既にコンソーシアム[9]も立ち上がっており，HMC は DIMM のように国際規格ができあがるのは時間の問題である．その結果，HMC は様々な情報機器に利用され，Exa FLOPS マシンよりはるかに早期に生活へ恩恵を及ぼすのは確実な情勢と考えられる．

一方，現状の HMC プロトタイプはキャッシュが発生するバーストアクセスに適合した設計になっており，Exa FLOPS マシンの文脈からは必ずしも完成形ではない．高い Byte/FLOP 値を要求する HPC 系アプリケーションのアクセスパターンは，キャッシュが苦手とする間接参照に伴うランダムアクセスが多いためである．

そのような問題を解決するために筆者らは米国に先立ち，先行研究[10]-[26]で Scatter/Gather 機能を有するメモリシステムを提案してきた．2003 年の提案以来，数年に渡り国際ワークショップ[10]で P. M. Kogge 教授らとも議論を重ねてきた．文献[18]-[26]では疎行列ベクトル積においても評価を行ない，有効性を示してきた．

米国 IAA は設立当初より Memory project[2]において，Exa FLOPS マシンの文脈から

[†]株式会社 東芝
Toshiba corporation
[††]東京農工大学
Tokyo University of Agriculture and Technology

ⓒ 2012 Information Processing Society of Japan

メモリ側での Scatter/Gather 機能に注目している．現状の HMC プロトタイプにはその機能が入っていないが，この機能が将来の HMC に実装される可能性は高いと考えられる．ただし，この機能はバーストアクセスが高速であれば問題が無いアプリケーションには恩恵がなく，主に HPC 分野への恩恵が大きな機能である．このため，最悪のシナリオでは Scatter/Gather 機能付き HMC は COTS(Commercial Off-The-Shelf)にはなりきれず，米国産 HPC 向けプラットフォーム専用の特注部品となる可能性がある．よって，米国がこれを開発するのを待っているだけでは，日本の HPC 業界における国際競争力維持が危うくなる可能性も懸念される．言わばエクサ時代の HPC 分野における国際的な場での勝敗を左右しうる最重要な構成部品と言っても過言ではないと考えられる．よって，国産技術の粋を結集した国際競争力の高い HMC のフィージビリティ研究を，一刻も早く国策として推進すべきであると考える．

本研究では，以上の認識から，Gather 機能を有する DDR3 DRAM ベースの HMC の予備評価について報告する．処理時間が膨大で探索範囲が限られた従来のソフトウェアシミュレータではなく，実機に近い FPGA ベースの柔軟なハードウェアエミュレータにより，様々な設計パラメータを大きく振った際の効果や，ハードウェア量の変化を確認する．

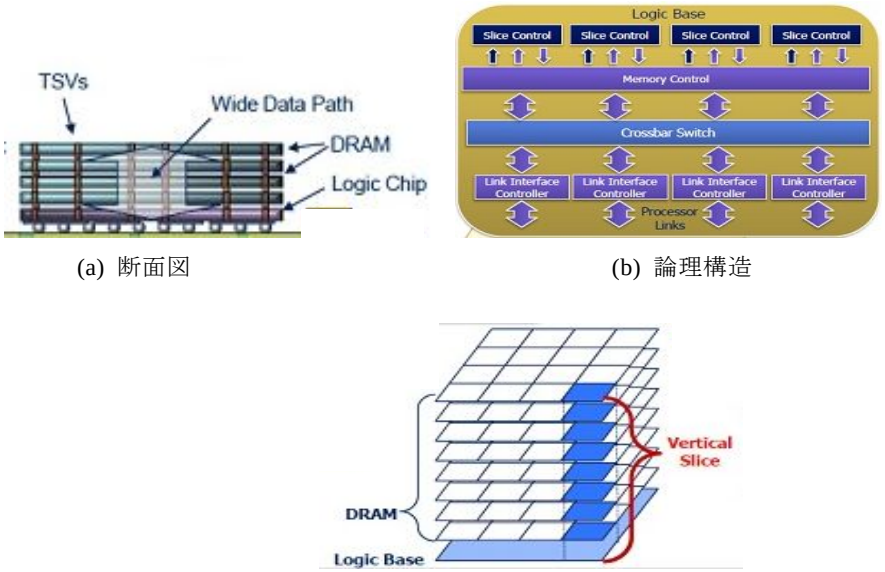
以下，本報告では第 2 章で HMC について紹介する．第 3 章では提案方式について述べる．第 4 章では Gather 機能を有する DDR3 DRAM ベースの HMC の評価を示す．第 5 章で関連研究を紹介したのち，第 6 章でまとめる．

2. Hybrid Memory Cube(HMC)

本章では本研究が対象としている新型メモリデバイスである Hybrid Memory Cube(HMC) [6][7][8]について紹介する．

2.1 概要

2011 年 8 月 HotChips にて Micron Technology 社によって発表された HMC[6][7][8]は，シリコン貫通電極(Through Silicon Via：TSV)を用いた三次元積層メモリである．HMC では図 1(a)に示すように複数の DRAM チップと，ロジックベースと呼ばれる制御回路が 1 つのパッケージとして実装される．従来は CPU あるいはマザーボード上に配置されていたメモリコントローラをロジックベースとしてメモリモジュールと一体化した構成となっているため，TSV によってメモリコントローラと DRAM 間の通信を高速で行うことが可能である．また，HMC の高速なスループットに対応する高速な通信リンクを CPU との間に持つことで，メモリシステム全体のスループットを格段に向上させている．HMC は，主に電力制約の観点から，バンド幅と容量を両立する必要がある多くのアプリケーション向けの主記憶として有望視されている．HMC の採用により電力効率で約 10 倍，バンド幅で約 20 倍の向上が見込める．



(a) 断面図　(b) 論理構造

(c) スライス分割

図 1　Hybrid Memory Cube(HMC)の構造（出典：参考文献[7]）

HMC はロジックベース上に以下の 3 種類のコントローラを持っており，図 1(b)の形でそれぞれ接続されている．

- **スライスコントローラ**： DRAM チップを制御する．
- **リンクインターフェイスコントローラ**： CPU との通信を制御する．
- **メモリコントローラ**：スライスコントローラとリンクインターフェイスコントローラの接続を制御する．

図 1(c)に示されるように1枚の DRAM チップはスライスと呼ばれる単位に分割されており，垂直方向から見て同じ位置にあるスライスはまとめて一つのスライスコントローラが制御する．一つのスライスは 2 バンクを持つため，DRAM チップが 4 枚でスライス分割数が 16 であれば，総バンク数は 128 バンクとなる．これは一般的な 8 バンクのメモリモジュールの 16 倍の数であり，非常に高いバンクの並列性を発揮できる．また，各スライスコントローラは独立して動作するため，1 個の HMC で 16 チャンネルのメモリモジュールに等しい働きを行う．

リンクインターフェイスコントローラは CPU と HMC を接続する通信リンクのメ

ⓒ 2012 Information Processing Society of Japan

モリ側終端端末である．例えば現行の CPU である Core i7 では，QPI と呼ばれる通信リンクを用いて最大 25.6GB/s での転送を行うことができる．しかし，HMC は 100GB/s を超えるスループットを達成しており，現行のインターフェイスでは性能を発揮するには不十分である．このため，HMC では CPU との通信に従来のインターフェイスとは別の高速シリアルリンクを複数設置することで高いスループットを実現する． 1 セットのリンクは 10Gbps で動作する差動信号線を上下各 16 ペア持ち，合計最大 40GB/s のスループットを発揮する．このリンクを最大 8 セット用いる事で，CPU と HMC は 320GB/s という高速通信が可能となる．

メモリコントローラはクロスバースイッチによってリンクインターフェイスコントローラに接続され，リンクインターフェイスコントローラからのアクセス要求をスライスコントローラに振り分ける．

上記の 3 種類のコントローラにより，Hybrid Memory Cube は動作を外部に対して隠蔽できているため，CPU やマザーボードはメモリシステムの内部仕様を気にせず稼働できる．

Micron Technology 社の発表によれば，第 1 世代の Hybrid Memory Cube は 4 枚の DRAM チップを持つ構成で 128GB/s のスループットを実現している．この数値は，現在広く用いられている DDR3-1333 の 10 倍以上の数値である．

3. Gather 機能付き Hybrid Memory Cube

3.1 基本アーキテクチャ

不連続アクセス課題の解決策として，DIMMnet-2 と同様の連続化ハードウェア（分散/収集機構）をプロセッサのコアから見て内部ネットワークよりメモリに近い場所に追加することを提案した．本報告における提案はそれを三次元積層実装により HMC として実現することで，不連続アクセスにおける低消費エネルギーと高実効バンド幅と高実装密度を両立させることである．提案方式の基本コンセプトを図 2 に示す．

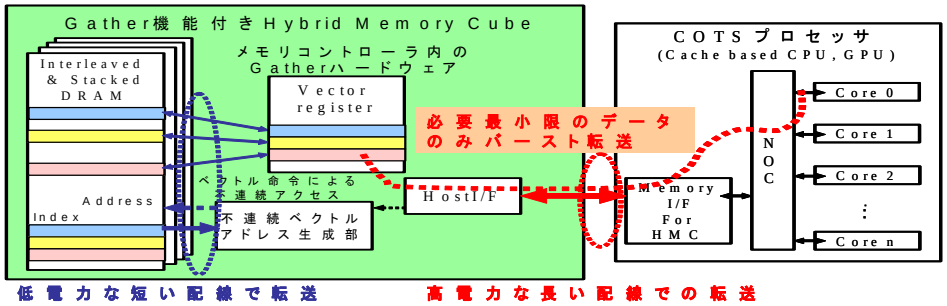


図 2　提案する基本アーキテクチャ

表 1 に DIMMnet-2 の主なベクトル型アクセスコマンドを示す．このうち，等間隔ロード/ストア，リストロード/ストアのコマンドが不連続アクセスの連続化を実行するものである．ロード系が外部メモリから一種のベクトルレジスタである Prefetch Window への収集(Gather)処理を行い，ストア系が一種のベクトルレジスタである Write Window からの外部メモリへの分散(Scatter)処理を行なう．

表 1　DIMMnet-2 の主なベクトル型アクセスコマンド

Load	Burst	VL
	Strided	VLS
	Indexed	VLI
Store	Burst	VS
	Strided	VSS
	Indexed	VSI

3.2 不連続アクセス向けメモリ構成

上記の基本アーキテクチャに基づく Scatter/Gather を効率的に行うには大容量メモリに対する不連続アクセスのスループットが高くなるようにする必要がある．本節では不連続アクセスのスループットを向上させる際の 4 つのポリシーを列挙する．詳細は文献[22]で述べているので参照されたい．なお，後述する性能評価では DDR3 型の現時点で最も一般的に使用されている DRAM のパラメータを用いており，以下の最後の項目は反映していない．新型メモリの効果の評価は今後の課題である．

- 狭いビット幅のチャネル
- 深いインタリーブ
- Open ページポリシー
- 低サイクルタイム型メモリの活用

3.3 ホストインタフェース

Exa FLOPS 級マシンのメモリシステム向けである提案メモリシステムは，DRAM または MRAM を三次元積層実装した HMC として実装される．つまり，ホストインタフェースは HMC のコンソーシアムが近々決定するであろう HMC のチャネルインタフェースに準拠することになる．これはこれまで筆者らが提案してきた DIMMnet-2 ベースのメモリシステムを最新の三次元積層技術で実装しなおしたものと言い換えることができる．まだ，現時点では実機が出現していないが，前述の通り IAA における開発方針と Micron 社の開発動向をあわせれば，この提案システムと類似した実機(Gather 機能つき HMC)が米国で出現する可能性は高いと考えられる．

3.4 三次元実装を採用する意義

三次元実装はメモリチップ間の配線数制約緩和に伴う実効バンド幅の向上と，配線長およびそれに伴う消費エネルギーの削減に寄与する．

ⓒ 2012 Information Processing Society of Japan

筆者らによる先行研究[19][22]では二次元実装を前提にしたパラメータの範囲では Scatter/Gather を効率的に行うには深いインターリーブ（チャネル数を多く取ること）の効果が大きいことをソフトウェアシミュレータによって確認していた．三次元実装によるメモリチップ間の配線数制約緩和によって，チャネルのビット幅をある程度まで維持した上でチャネル数を多くとるアプローチも許容されるようになる．あるいは，チャネルのビット幅をできるだけ絞り，増えた配線数をチャネル数の増強にあてるアプローチも許容されるようになる．どちらのアプローチが有効なのかは三次元実装を前提にした新しい制約下で定量的に評価する必要がある．

三次元実装によりメモリチップとメモリコントローラ間の配線長は，従来の二次元実装による 10cm オーダーから，悪くて数ミリメートル，良い場合はミクロンオーダーに劇的に変わる．データ転送のために消費されるエネルギーは配線長に比例して大きくなるため，この桁違いに短い配線をランダムアクセス性能向上のために必要になるアドレス線の増加分として用いても電力への影響は限定的となるはずである．メモリアクセスによって消費される電力のほとんどが HMC と CPU 等のマスタデバイス間の長距離配線でのデータ転送になると考えられる．4 バイトのランダムな不連続アクセスを一般的なキャッシュラインサイズである 128 バイトで行う場合には最も消費電力が大きい長距離配線を連続アクセスの 32 倍の時間使用することになる．これに対して，提案方式は上記の最も電力を消費する長距離配線を連続アクセスと同等の時間しか使用しない．

またリストベクトル(Index 配列)を用いた間接参照を行う場合，インデックス配列の転送もバンド幅とエネルギーを消費する．プロセッサ側でインデックスからメモリアドレスへの変換を行う従来の方法では上記の最も電力を消費する長距離配線をインデックスおよび変換後のアドレスが通過するため貴重なプロセッサ～メモリ間バンド幅と長距離配線による多くのエネルギーが消費される．これに対して提案方式はプロセッサ～メモリ間バンド幅を消費せず，短距離配線による少ないエネルギー消費で済ませることができる．

以上のように，Scatter/Gather と三次元実装を適切に組み合わせるならば，不連続アクセスの際に桁違いの消費エネルギー削減が達成されると考えられる．

4. 性能評価

4.1 評価方法

4.1.1 評価環境

使用ハードウェアは Altera 社 Stratix III Development Kit である．これが搭載する FPGA は Stratix III EP3 SL150F1152 であり，現時点の最新製品ファミリ(Stratix V)の 2 世代前で，かつ，そのファミリの中でも総ロジックエレメント数は 142,500 と中規模(最大規模の半分以下)の製品である．この FPGA 上に Verilog HDL によって機能を記述し，論理合成した回路を実装した．HMC に対してアクセス要求を発生するテスト用マスタデバイスモデルと DRAM モデルと，HMC のリンクインタフェース以外のコントローラを様々な構成パラメータに対して構築した．その上で，想定システムの 1/20 の周波数で動作する FPGA エミュレータによるサイクル数を計測して得られる性能評価と，回路規模の評価を行う．

4.1.2 FPGA エミュレータの構成と動作概要

(1) テスト用マスタデバイスモデルユニット

テスト用マスタデバイスモデルは CPU の代わりに HMC に対してアクセス要求を発生する．本モデルが発生する命令はスカラロード(Load)命令とベクトル間接ロード（VLI）命令の二種類である．最初の命令発行から最後の読み込みデータの受信までの実行時間を記録する．シミュレーションの経過時間や進捗状況などを開発ボードに表示する．この部分は HMC には本来必要の無いものなので，回路規模の評価対象には入らない．

(2) メモリコントローラ

メモリコントローラ(MC)は図 3 に示すようにコマンドキューと，スカラロード命令処理ユニット，Scatter/Gather 命令処理ユニットを持つ．

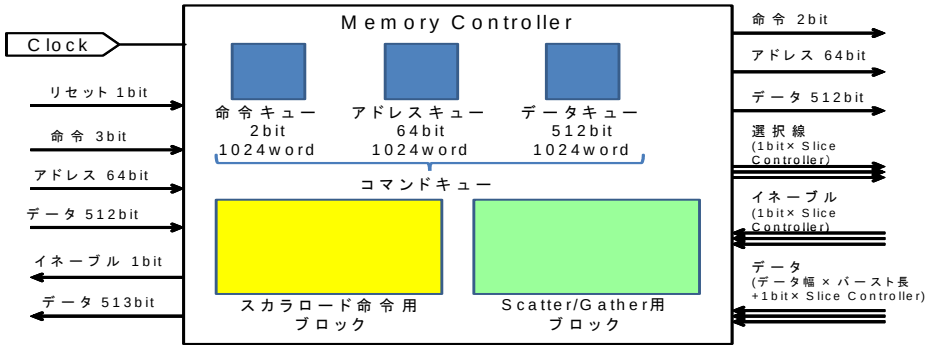


図 3　メモリコントローラの構成

スカラロード命令処理ユニットは，Load 命令を実行するユニットである．スライスコントローラから受信するデータのサイズは，メモリバンクのビット幅×バースト長であるため，メモリバンクのビット幅が小さい場合には 1 回の要求発行でキャッシュラインサイズのデータを揃える事が出来ない．その場合には複数のスライスコントローラに同時に要求発行を行う事でキャッシュラインサイズのデータを揃える．また，スライスコントローラの数が少ない場合には各スライスコントローラに要求発行を複数回行うこともある．そのため，スカラロード命令処理ユニットはキャッシュライン

ⓒ 2012 Information Processing Society of Japan

サイズ(今回は 64 バイト)のバッファを持ち，バッファにデータが全て蓄積した時点でテストユニットにデータを送信する．

Scatter/Gather 命令処理ユニットは VLI 命令を実行するユニットである．内部にはインデックスメモリ，不連続ベクトルアドレス生成部，プリフェッチウィンドウ(一種のベクトルレジスタ)を持つ．Scatter/Gather 命令処理ユニットが命令を受け取ると，不連続ベクトルアドレス生成部はインデックスメモリからインデックスを読み取り，スライスコントローラに要求を発行する．スライスコントローラから受信した読み込みデータはプリフェッチウィンドウに格納し，キャッシュラインサイズまで蓄積されたものをテストユニットに転送を行う．また不連続ベクトルアドレス生成部内にあるスカラレジスタには，収集するデータの個数やプリフェッチウィンドウ内のデータ格納位置など，VLI 命令に必要なパラメータが格納され，不連続ベクトルアドレス生成部はスカラレジスタ内の値を参照して動作を行う．

なお, インデックスを格納するバッファメモリは DRAM 側からベクトルロードする FIFO とするなどして, 内蔵しきれない大きなインデックス配列への対応をすべきであるが，今回の予備評価では内蔵 ROM として実装し，ワークロードのインデックス配列で初期化している．このため評価できるワークロードのサイズは限定されている．

また，今回の予備評価では両ユニットが 1 サイクルあたりに生成するアドレスは 1 個に限定されている．よって，DRAM 側のスループットを限界まで引き出すような構成にはなっていない．先行研究ではチャネル数(スライス数)の並列度を上げていってもアドレス入力のスループットが低いと全体スループットが飽和することがソフトウェアシミュレータ上で確認されている．これらの制約を改善した実装による評価は今後の課題である．

(3)スライスコントローラ

スライスコントローラ(SC)はメモリコントローラからの要求に従いデータを出力するモジュールである．図 4 にその構成を示す．評価環境には HMC の垂直方向配線のように DRAM を外部に多数接続するだけのピン数がないため，スライスコントローラ内に DRAM モデルを一体化している．内部に 8 バンク分の記録域とコマンドキューを持つ．DRAM モデルの記憶域には内蔵メモリを使わず，バンクあたり固定バースト 1 回分のみを記憶している．つまり，タイミングだけ正確で，読み出されるデータは模擬していない．また，バンクに対する操作は現在の主流であるメモリインターフェイスである DDR3 に準じた動作を行う．一方，コマンドキューには内蔵メモリブロックを用いて，64 ワード分を構成している．メモリコントローラから受信した要求はコマンドキューに格納され, 前述のメモリコントローラと同様, フロー制御される．

4.1.3 評価対象のメモリシステム

(1) DRAM モデル

評価に用いた DRAM モデルの主なパラメータ値を表 2 に示す．FPGA 内部に実装されている制約のため，太字の部分が先行研究[22]におけるソフトウェアシミュレータによる測定において用いた値と大きく異なる部分である．

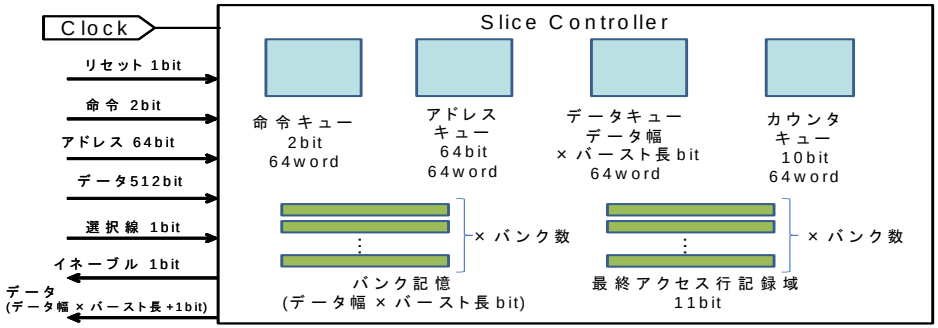


図 4　スライスコントローラの構成

表 2　評価に用いた DRAM モデルの主なパラメータ

	ソフト実装[22]	FPGA 実装
DRAM 種類	DDR3	
行数	32768	**1024**
列数	2048	
バンク数	8	
BL(バースト長)	8	
tCK(転送サイクルタイム)	0.5ns	**10ns**
CL(CAS レイテンシ)	10	9
tRCD(RAS to CAS レイテンシ)	10	9
tRP(RAS プリチャージレイテンシ)	10	9
tRAS(RAS レイテンシ)	24	27

(2)システム構成

評価に用いたメモリシステムのシステム構成パラメータを表 3 に示す．

表 3　評価したメモリシステムのシステム構成パラメータ

システム構成パラメータ	ソフト実装[22]	FPGA 実装
スライスコントローラ数	1, 2, 4, 8, 16	1, 2, 4, 8, 16, **32, 64**
1 サイクルで発生するアドレス数	1, 2, 4, 8	**1**
チャネルあたりのビット幅	8,16,32,64	
ランク数	4	**0**

ⓒ 2012 Information Processing Society of Japan

先行研究[22]で改造して用いたソフトウェアシミュレータ[27][28][29]は実行時間が膨大なため，チャネル数(スライスコントローラ数)を 16 以上にできなかった．一方，本報告の FPGA 実装では桁違いに高速であるため更に増やすことが可能である．

なお, ランク数は DRAM を 4 チップ積層する HMC の場合は 4 であるべきとも考えられるが，3 次元実装によって配線が短いときでも通常と同様にランクが変わる際の 1 サイクルのオーバーヘッドが計上されるべきか否かは不明だったため，今回は全てバンク扱い（2 バンク×4 枚＝8 バンク）として実装されている．

(3) アドレスビットマッピング

インタリーブ構成のアドレスマッピングはアドレスの下位から固定バースト分，スライス，バンク，列，行の順に割り当てた．これにより，固定バースト長単位でスライスが切り替わり複数スライスの並列動作を促進した．

4.1.4 ワークロード

疎行列ベクトル積において提案メモリにオフロードすることを想定し，その際のベクトルへの間接アクセスのトレースを University of Florida Sparse Matrix Collection[30]から FPGA の内蔵 ROM に入りきる小規模な疎行列を選択して作成した．上記コレクションの拡張子 mtx のファイルの index 部分を index としてデータサイズ 8 バイトとして 0 番地から配置される配列をアクセスする際のアドレストレースを生成した．表 4 に評価に用いた疎行列の特徴を示す．

表 4　評価に用いた疎行列の特徴

行列名	応用分野	行数	非零要素数			
			合計	行平均	行最大	標準偏差
msc01440	流体解析	1440	23855	16	40	12.3
bcsstk13	構造解析	2003	42943	21	84	14.68
nasa4704	構造解析	4704	54730	7	20	4.28

また，今回の実験では GPU 向けの評価として，文献[18][21]にて提案した提案拡張メモリと GPU を組み合わせたシステム向けのアルゴリズムの前処理部分を適用したアドレストレースを用いた場合のバンド幅を測定した．なお，今回用いた前処理では折り畳み幅の個別調整は行っていないものを用いた．また，0 パディングが index ファイルには入っているが，値は 0 に固定されるためメモリアクセスを行わなくてもコントローラ内部のレジスタまたはキャッシュなどで代用できるため，0 パディングに対応するアクセスはトレースファイルから省略されている．

4.2 スライスあたりの bit 幅によるアクセスの性能への影響

スライスコントローラを 1 個に固定し，スライスあたりの bit 幅を変化させた際に，スカラロード命令 Load とベクトル間接ロード命令 VLI を用いた場合の二つに対し，表 4 の 3 種類のワークロードの実行時間への影響を図 5，図 6 に示す．

図 5 に示されるように Load 命令のスループットがビット幅に影響を受けやすいことと，図 6 に示されるように VLI 命令のスループットがビット幅に影響を受けにくいことが読み取れる．

Load 命令のスループットがビット幅に大きく影響を受けるのは，スライスコントローラからメモリコントローラに渡されるデータのサイズがビット幅×バーストサイクルとなっているためである．キャッシュラインサイズ(今回は 64 バイト)のデータを揃えようとすると，ビット幅が少なくなるにつれてメモリコントローラが発行する要求が増加する．スライスコントローラの数が少ないと要求が集中するため，読み込みに時間がかかりスループットの低下を招く．

一方 VLI 命令では，プリフェッチウィンドウに格納するデータはスライスコントローラから受け取ったデータのうち 8 バイトであるため，スライスコントローラの扱うデータサイズが 8 バイト以上であれば要求発行回数は変化しない．

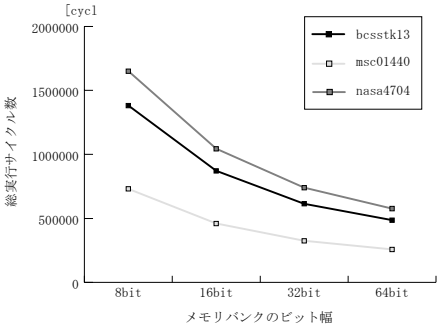


図 5　Load による実行時間とビット幅

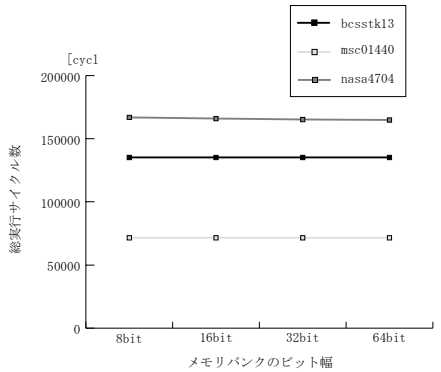


図 6　VLI による実行時間とビット幅

4.3 スライスコントローラ数によるアクセスの性能への影響

ビット幅を 8 ビットに固定し，スライスコントローラ(SC)数を変化させた際に，前記と同様のワークロードの実行時間への影響を図 7，図 8 に示す．

図 7 に示されるように Load 命令ではスライスコントローラ数によるスループットの変化が見られなかった．その理由は，8 バイトのデータをアクセスする場合には 1 回の命令に対し 1 回の要求しか発行されず，スライスコントローラの数が影響を与えないためだと考えられる．

一方，図 8 に示されるようにスライスコントローラ数が多いほど VLI 命令による処

ⓒ 2012 Information Processing Society of Japan

理速度が増加した．これは，1 回の VLI 命令で最大 64 回の要求をスライスコントローラに発行するため，スライスコントローラが増えるほどスライスコントローラの 1 個当たりの要求数が減少し，総読み込み時間が短縮した点が原因と思われる．

図 8 においてはスライスコントローラが 4 程度で性能向上が飽和していることも確認できる．この現象はソフトウェアシミュレータ上で 1 サイクルあたりのアドレス生成数がチャネル数に対して数倍少ない際に確認されていたもの[22]である．DRAM 側のスループットがスライスコントローラ数を増やすことで増加しても，それに投入されるアクセス要求自体がバランスしないと，そちらがボトルネックとなりスループットが上がらなくなると考えられる．HMC を設計する際にはこのバランスを十分に検討した上で決定していく必要がある．今回の実験範囲では 1 サイクルあたりのアドレス生成数が 1 に固定になっている実装のため，スライスコントローラを 4 以上に増やす効果が少ないように観測されているが，ソフトウェアシミュレータで得られた知見[22]を併せると，1 サイクルあたりのアドレス生成数(SIMD 並列度)を増やした実装にしていけば性能向上が飽和しなくなることが予想される．

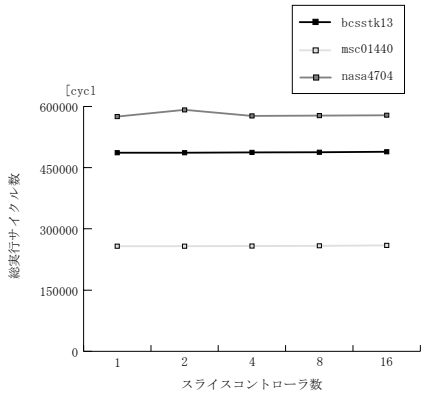


図 7　Load による実行時間と SC 数

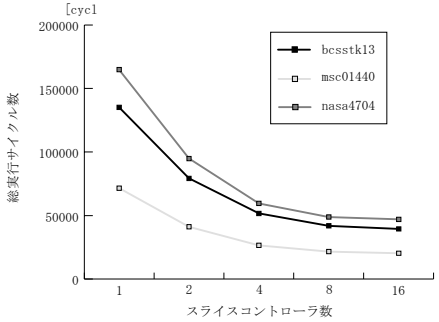


図 8　VLI による実行時間と SC 数

4.4 データ線配線数一定条件下における物量組合せの影響

HMC が TSV による三次元実装(垂直方向配線)によって多数の配線を DRAM チップ間に設置できるようになるとはいえ，その本数は有限である．アドレス線や制御線が誤差を生むものの，データ線についてはビット幅を半分にしたらスライス数を 2 倍に増やすと使用配線数は一定に保たれる．スライスコントローラと DRAM の間がパケット式の通信チャネルで接続される実装では誤差が無い．ここでは，データ線(または通信チャネル)に割り当てられる垂直方向配線を 512 本で一定としてビット幅とスライス数の組合せを変化させた場合の，前記と同様のワークロードの実行時間への影響を図 9，図 10 に示す．

図 9 に示されるように Load 命令ではどの組合せでも同じサイクル数が消費された．Load 命令ではキャッシュラインサイズ(64 バイト＝512 ビット)単位での連続アクセスとなるので，固定バースト長が 8 の DDR3 では 1 回の Load 命令が 64 本のデータ線を用いて 8 サイクルかけて 512bit 分読み出す．本実装のように 1 個のアドレス生成部から複数のスライスコントローラへの分割された要求が，同じサイクル内にスライスコントローラ側で実行される実装になっていれば，どの組合せでも同じサイクル数が消費される．ただし，複数のアドレスが同時に生成される場合はビット幅が狭くスライス数が大きい構成の方が，スライスへの衝突が発生する確率が減る．このため同時に複数の要求をさばける確率が上がるため，性能が高くなると予想される．この点について実際にその条件を FPGA 上に実装した実験確認は今後の課題である．

一方，図 10 に示されるように，前節の実験同様の飽和傾向が見えるものの，ビット幅が狭くスライス数が大きい構成のほうが VLI 命令による処理速度が増加した．これは，上記の Load 命令が同時発行される条件下での傾向と一致する．

以上の 3 つの実験とソフトウェアシミュレータで得られた知見[22]を総合すると「HMC における VLI 命令のスループット向上には，少ないビット幅のスライス構成とし，三次元実装により緩くなった配線数制約で実装可能な範囲でスライス数を増加させ，それにバランスするアドレス生成能力を与えるべき」という設計指針の妥当性が概ね裏付けられた．その設計指針は命令が同時発行される条件下では Load 命令においても同様であると考えられる．実際にその条件を FPGA 上に実装した実験確認は今後の課題である．

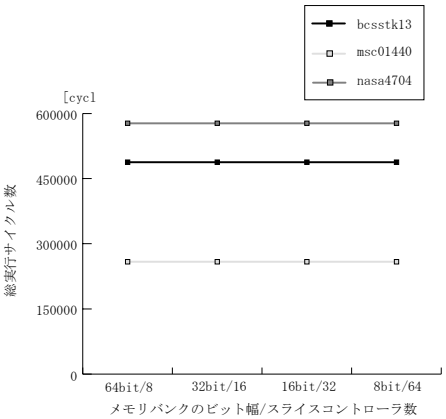


図 9　Load による実行時間と物量組合せ

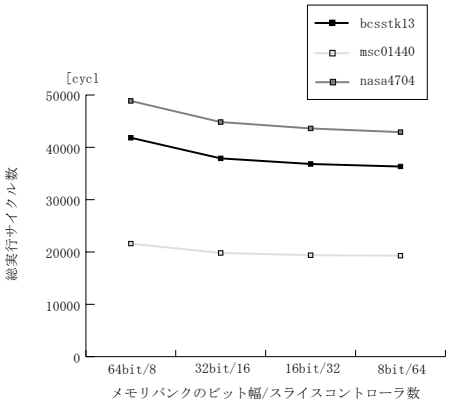


図 10　VLI による実行時間と物量組合せ

ⓒ 2012 Information Processing Society of Japan

4.5 Gather 回路による加速率

Gather 回路が無い Load 命令による実行サイクル数の Gather 回路がある VLI 命令による実行サイクル数に対する比率(Gather 回路による加速率)を図 11 に示す. Load 命令が発生する 64 バイトアクセスと VLI 命令が発生する 8 バイトアクセスの差がある．このため，今回の評価のように CPU 側のキャッシュが効いていない(Load 命令が必ず 64 バイトアクセスを 1 回発生させる)状態では，8 倍のデータ転送が発生する．単純計算では 8 倍の性能差が出ることが予測されるが，実際には 11 倍から 13 倍の性能差が観測された．Load 命令による場合はそれが発生するキャッシュラインの範囲が複数スライスにまたがる場合は複数要求を発生させるという遅延がデータ転送以外にもかかっていることが原因と考えられる．また，京や Nvidia 社の Fermi 世代以降の GPU のキャッシュラインサイズは 128 バイトであり，本評価と同等な評価をキャッシュラインサイズ 128 バイトに置き換えて行うならば，今回の評価より 2 倍近い加速率が得られると考えられる．

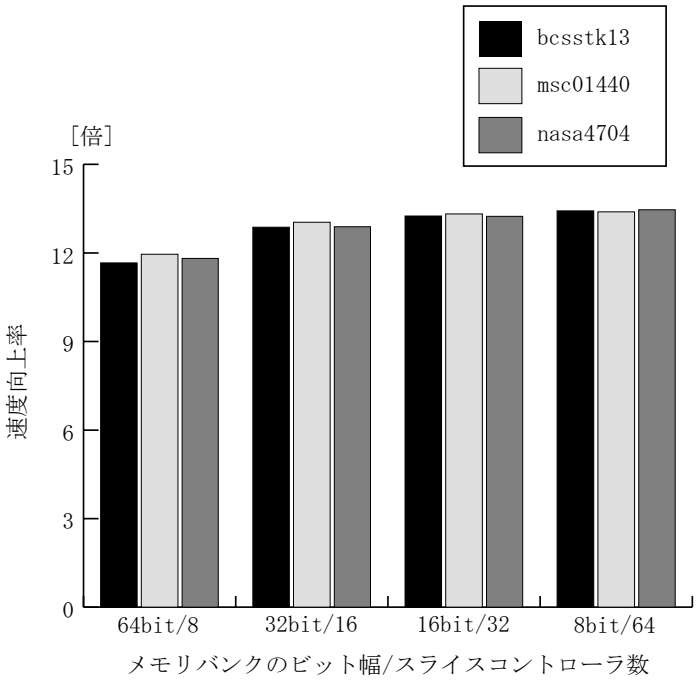


図 11　Gather 回路による加速率

4.6 回路規模

ロジックエレメントの使用率について，スライスコントローラを 1 個に固定しビット幅を変更した場合の変化を図 12 に，ビット幅を 8Byte に固定しスライスコントローラ数を変更した場合の変化を図 13 に示す．ロジックエレメントの使用率はスライスコントローラ数に正比例に近い形で増加し，ビット幅の影響は確認できない．よって，VLI 命令のスループット向上を目指すためにスライスコントローラ数を多くするためにはロジックベースの回路面積制約を満たす必要がある．

回路面積については本実装に用いた 2 世代前のミドルクラスの FPGA に，Gather 機能付のメモリコントローラや，ダミーの DRAM モデル込みで 64 個のスライスコントローラを問題なく実装できている．この他にリンクコントローラや，等間隔アクセス命令や，使い勝手をよくするための複数のウィンドーメモリの追加を行う必要があるが，回路の大半は個数の多いスライスコントローラで占められるという傾向には変化が無いと考えられる．2018 年頃に Exa FLOPS マシンが必要になる頃にはさらにロジックの集積度が大幅に上がることはほぼ確実であるとともに，DRAM のダイサイズより小さくない必要があることから，ロジックベースの回路規模そのものは問題にならないと考えられる．

ただし，アクセスのランダム性が高く，スライスの活性率が高い状況では，スライスが増えて回路規模が大きくなるほど消費電力も大きくなると予想される．今回評価できていないロジックベースの消費電力が問題になる可能性は否定されていないため，より電力を正確に測れる実装またはシミュレーションにより実効電力の評価を行う必要があり，それは今後の課題である．

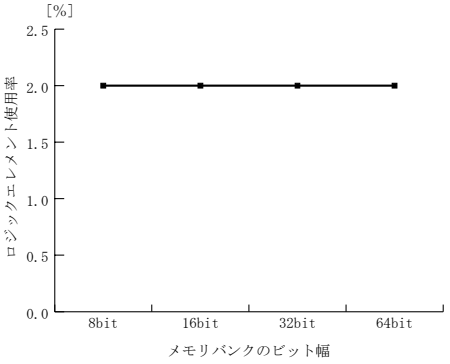


図 12　ビット幅と回路規模の関係

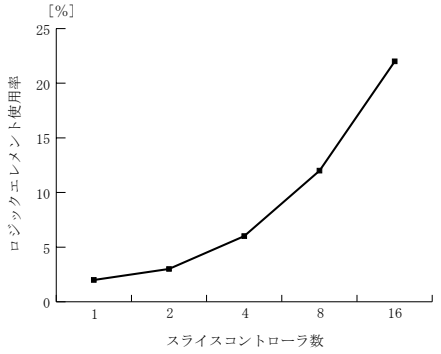


図 13　スライス数と回路規模の関係

ⓒ 2012 Information Processing Society of Japan

5. 関連研究

不連続アクセスに伴う実効バンド幅の低下問題を解決するために筆者らは先行研究[10]-[26]で Scatter/Gather 機能を有する拡張メモリシステムを提案した．特に文献[19][22][23][25]ではソフトウェアシミュレータ上で疎行列ベクトル積を用いて評価を行ない，有効性を示してきた．ただし，これらはシミュレーション時間が膨大であったため，チャネル数や行列サイズを大きくした評価には限界があった．このため 16 チャネルという二次元実装では適切な範囲であっても，本報告が対象とする HMC のような三次元実装を前提にした，より多くのチャネル数(スライス数)に関する評価を取れていなかった．本報告では FPGA を用いたエミュレーションによって模擬速度の問題を解決して，より広いパラメータ空間を手軽に探索できるようになった．さらに本報告で得られたような回路規模に関する知見は，それらの先行研究では全く得られていなかった．

FPGA を用いた Scatter/Gather 機能を有する拡張メモリシステムの評価研究としては DIMMnet-2 のプロトタイプを用いた研究[11][12]も存在する．それらは FPGA の外部に実際に少数の DRAM(SO-DIMM)を実装した試作に基づく．これに対し本報告ではリアルな DRAM を使用していない．このためタイミングは正確に模擬しているが，データの内容までは模擬しておらず，ハードウェアを用いた模擬(エミュレーション)を行っている．その結果，リアルな DRAM のチャネル数制約に依存しない，HMC の三次元実装を前提にしたパラメータ空間における性能評価を実現している．

京などの現行マシン上での最適化では，何らかの制約(例えば 1 行内非零要素数は最大 27 に固定した行列のみに通用するような制約)がかかるアプリケーション個別のアルゴリズムレベルの最適化でキャッシュを効率化する手法[31]が主流である．しかし，そのアプローチはアプリケーションごとにスキルの高いプログラマによる高度なチューニングの手間がかかり，汎用性が低い．アプリケーションの更新頻度が高い場合等，チューニングにあまり投資ができない利用形態には向かない．一方，文献[25]は Exa FLOPS マシンの文脈ではキャッシュに頼りきるのは危険であることを，様々な疎行列上での評価をもとに警鐘を鳴らしている．本報告は後者の立場に立った研究であり，行列演算ライブラリやベクトル化コンパイラなどに任意の疎行列を与えるような利用状況を想定する．その状況下で，COTS に付加された先進的なハードウェア(改良版の HMC)の力を借りながら，巨大かつランダム性の強い多様な行列に安定した高性能を手軽に提供するための研究である点で，あまり類を見ない．

6. おわりに

本報告では Gather 機能を有する Hybrid Memory Cube(HMC)を提案し，予備評価を行った．具体的には FPGA を用いたエミュレーションにより従来のソフトウェアシミュレータでは測りにくかった三次元実装向けのパラメータ空間の評価を疎行列ベクトル積のワークロードに対して行なった．

その結果，三つの実験とソフトウェアシミュレータで得られた知見[22]を総合すると「HMC における間接ベクトル参照のスループット向上には，少ないビット幅のスライス構成とし，三次元実装により緩くなった配線数制約で実装可能な範囲でスライス数を増加させ，それにバランスするアドレス生成能力を与えるべき」という設計指針の妥当性が概ね裏付けられた．その方向性は命令が同時発行される条件下ではキャッシュラインベースのスカラ Load 命令においても同様であると考えられる．

また，Gather 機能を HMC に実装することにより，様々な構成で配列の間接参照のスループットが 11〜13 倍の性能向上が得られることが確認された．回路規模は現時点の最新より 2 世代前の中規模 FPGA で十分に実装できる範囲にあることを確認した．

今後の課題としては，インデックス配列を外部メモリから取り込める構成に改善することで実現する大きな行列群での評価がある．さらに，同時生成アドレス並列度を増加させた構成に改善することで実現するスライスを増加させた際の性能向上の正確な評価も今後の課題である．提案方式は処理性能だけでなく電力削減も同時に狙っており，電力の評価も重要な今後の課題である．

今回評価に用いた DDR3 より高性能が期待できる XDR 型や DDR4 型 DRAM や 2015 年に DRAM を代替するというロードマップも引かれている STT 型 MRAM[32]等のメモリを用いた場合の評価も今後の課題である．これらの検討は米国の Micron 社任せではなく国内で独自開発を進めるべきか否かを判断する材料として重要性が高まってきている．今回は 3 種類の小さな行列での評価であるが，様々な重点アプリケーション(あるいはそこで用いられる行列や間接参照をボトルネックとするカーネル)上で本アプローチの有用性および汎用性を確認することも重要な今後の課題である．さらに Gather 機能を有するメモリシステムの利用を促進するベクトル化コンパイラや行列演算ライブラリなどの基本ソフトウェアの整備も今後の課題である．

謝辞 本研究の一部(DIMMnet-2 の開発)は総務省戦略的情報通信研究開発推進制度(SCOPE)の一環として行われたものである．

参考文献

1) 平木 : "[招待講演]将来の HPC アーキテクチャ", ハイパフォーマンスコンピューティングと計算科学シンポジウム 2012 (HPCS'12), pp.163-167, Jan.2012
2) IAA : "Exascale Computing and The Institute for Advanced Architectures and Algorithms (IAA) ", http://www.hpcuserforum.com/presentations/Norfolk/Sandia%20IAA.hpcuser.ppt, Apr. 2008
3) P. M. Kogge et.al. : "ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems," Univ. of Notre Dame, CSE Dept. Tech. Report TR-2008-13, Sep. 2008.

ⓒ 2012 Information Processing Society of Japan

4) R. C. Murphy, A. F. Rodrigues, J. A. Ang : “Memory Opportunities for High Performance Computing (MOHPC) Final Report”, SANDIA REPORT, SAND2009-7291 Feb. 2009. http://www.cs.sandia.gov/CSRI/Workshops/2008/MOHPC/MOHPC-1.pdf
5) IAA : “Focus area”, http://iaa.sandia.gov/focus-areas/index.html
6) Micron Technology, Inc. : "ハイブリッドメモリキューブ", http://jp.micron.com/innovations/hmc.html
7) Micron Technology, Inc. : " Hybrid Memory Cube : Breakthrough DRAM Performance with a Fundamentally Re-Architected DRAM Subsystem”, Hotchips 23, Aug. 2011.
8) Hisa Ando :” Hotchips23 – メモリバンド幅を画期的に高める Hybrid Memory Cube”, Sep. 2011. http://journal.mycom.co.jp/articles/2011/09/13/hot_chips23_micron/index.html
9) Hybrid Memory Cube Consortium : http://www.hybridmemorycube.org/
10) N. Tanabe, M. Nakatake, H. Hakozaki, Y. Dohi, H. Nakajo, H. Amano : "A New Memory Module for COTS-Based Personal Supercomputing", 7th International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems (IWIA2004), pp.40-48 Jan. 2004
11) T. Miyashiro, A. Kitamura, M. Yoshimi, H. Amano, H. Nakajo, N. Tanabe:” DIMMnet-2: A Reconfigurable Board Connected Into a Memory Slot”. International Conference on Field Programmable Logic and Applications (FPL'06), pp.1-4 Aug. 2006.
12) 宮代, 宮部, 北村, 田邊, 中條, 天野 : "DIMMnet-2 を用いた間接メモリアクセスの高速化", 情報処理学会研究報告. 計算機アーキテクチャ研究会報告, Vol. 2006, No.127, pp. 85-90, Nov. 2006.
13) N. Tanabe, H. Nakajo : " An Enhancer of Memory and Network for Cluster and Its Applications", IEEE PDCAT'08, pp.99-106, Dec. 2008.
14) N. Tanabe, H. Nakajo : " High Performance Computing and Database Processing with COTS and Extended Memory Modules", HPC Asia2009 (Best paper award), Mar. 2009.
15) N. Tanabe, M. Sasaki, H. Nakajo, M. Takata, K. Joe : "The Architecture of Visualization System using Memory with Memory-side Gathering and CPUs with DMA-type Memory Accessing", International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'09) , pp. 427-433, Jul. 2009.
16) N. Tanabe, H. Hakozaki, Y. Dohi, Z. Luo, H. Nakajo : " An enhancer of memory and network for applications with large-capacity data and non-continuous data accessing", The Journal of Supercomputing, Vol. 51, No. 3, pp. 279-309, Mar. 2010.
17) N. Tanabe, T. Tsukamoto, A. Ohta, H. Nakajo : " Efficiency Improvement for Discontinuous Accesses of Cell/B.E. Using Hardwired Scatter/Gather on Memory-side ", IEEE ICCEE'10, Nov. 2010
18) N. Tanabe, Y. Ogawa, M. Takata, K. Joe : " Scaleable Sparse Matrix-Vector Multiplication with Functional Memory and GPUs", Euromicro PDP2011, Feb.2011
19) N. Tanabe, B. Nuttapon, H. Nakajo, Y. Ogawa, J. Kogou, M. Takata, K. Joe : "A memory accelerator with gather functions for bandwidth-bound irregular applications", Proceedings of the first workshop on Irregular applications: architectures and algorithm (IAAA'11) in conjunction with SC11, pp.35-42 Nov.2011.
20) N. Tanabe, J. Kogou, M. Takata, K. Joe : " Evaluation of a GPU Enhanced with a Memory Accelerator by Using Sparse Matrix-Vector Product", Work-in-progresss session in Euromicro PDP2012, Feb.2012
21) 小川, 田邊, 高田, 城 : “機能メモリと GPU の PCI express 接続によるヘテロ環境における超大規模疎行列ベクトル積の性能予測”, 情報処理学会 HPC 研究会 Vol.2010-HPC-126 No.20, Aug. 2010.
22) 田邊, Nuttapon, 中條 : “Gather 機能付き拡張メモリのアクセス性能の評価”, 情報処理学会 HPC 研究会, Vol.2010-HPC-128, Dec. 2010.
23) 田邊, Nuttapon, 中條, 小川, 高田, 城 : “GPU と拡張メモリによる疎行列ベクトル積性能の行列形状依存性軽減”, 情報処理学会 HPC 研究会, Vol.2010-HPC-129, Mar. 2011.
24) 小郷, 田邊, 高田, 城: “メモリアクセラレータで強化した GPU の CG 法による評価”, 情報処理学会 HPC 研究会, Vol.2010-HPC-130, Aug. 2011.
25) 田邊, Nuttapon, 中條, 小郷, 高田, 城 : "不規則型応用を加速するメモリアクセラレータ- Exa FLOPS マシンの文脈から, 情報処理学会 HPC 研究会, Vol.2011-HPC-132, pp.1-8, Nov.2011
26) 田邊, 小郷, 小川, 高田, 城 : "Gather 機能を有するメモリアクセラレータの疎行列計算への応用", ハイパフォーマンスコンピューティングと計算科学シンポジウム 2012 (HPCS’12), pp.32-41, Jan.2012
27) D. Wang, B. Ganesh, N. Tuaycharoen, K. Baynes, A. Jaleel, B. Jacob : "DRAMsim: a memory system simulator", SIGARCH Computer Architecture News Vol.33, No.4, pp.100-107, Sep.2005
28) B. Jacob : "DRAMsim: A Detailed Memory-System Simulation Framework", http://www.ece.umd.edu/dramsim/v1/
29) B. Jacob : "DRAMSim2", http://www.ece.umd.edu/dramsim/
30) Tim Davis : " The University of Florida Sparse Matrix Collection", http://www.cise.ufl.edu/research/sparse/matrices/
31) 南, 井上, 堤, 前田, 長谷川, 黒田, 寺井, 横川 : "「京」コンピュータにおける疎行列とベクトル積の性能チューニングと性能評価", ハイパフォーマンスコンピューティングと計算科学シンポジウム 2012 (HPCS’12), pp.32-41, Jan.2012.
32) A. D. Smith, D. Apalkov, V. Nikitin, X. Tang, S. Watts, D. Lottis, K. Moon, A. Khvalkovskiy, R. Kawakami, X. Luo, A. Ong, E. Chen, M. Krounbi : "Latest Advances and Roadmap for In-Plane and Perpendicular STT-RAM", 3rd IEEE International Memory Workshop (IMW), pp.1-3 May 2011.

ⓒ 2012 Information Processing Society of Japan