

SCIMA におけるメモリアクセス機構の設計と評価

大根田 拓[†] 近藤 正章[†] 中村 宏[†]

プロセッサとメモリの性能格差の問題への対処を目的として、主にハイパフォーマンスコンピューティング分野をターゲットとしたプロセッサアーキテクチャSCIMAを提案している。SCIMAではチップ上のメモリとして従来のキャッシュに加えオンチップメモリを搭載し、オンチップメモリへのデータ転送命令として新たにpage-load/page-store命令を備える。本稿では、SCIMAのメモリアクセス機構の評価を行なった。従来のプロセッサにおけるメモリアクセス機構とSCIMAへの拡張を施した機構をともに設計し、SCIMAへの拡張が面積・遅延へ与える影響を調べた。

A Design and Evaluation of Memory Access Structure for SCIMA

TAKU OHNEDA,[†] MASAOKI KONDO[†]
and HIROSHI NAKAMURA[†]

The performance gap between processor and main memory is serious problem especially in high performance computing. In order to overcome this problem, we have proposed a new processor architecture called SCIMA, which integrates software-controllable addressable memory into processor chip as a part of main memory in addition to ordinary cache. We designed a memory access unit of SCIMA which is an extension of ordinary microprocessors. In this paper, we present its mechanism and evaluate its impact on area and clock frequency.

1. はじめに

近年、プロセッサの性能向上は著しく、それに対しメモリの性能はあまり向上していない。このため、プロセッサの性能がメモリアクセスによって制限されてしまうことが問題となっている。この問題に対し、従来のプロセッサではキャッシュを用いている。

しかし、特にハイパフォーマンスコンピューティング(HPC)分野のアプリケーションでは、キャッシュが有効に機能しない場合が多い²⁾。キャッシュ容量に対しデータセットが非常に大きく、またデータの時間的局所性がほとんどないためである。結果としてキャッシュミスが頻発し、性能が大きく低下してしまう。そこで、このHPC分野のアプリケーションを対象とした新しいアーキテクチャSCIMA (Software Controlled Integrated Memory Architecture)が提案されている⁶⁾。

SCIMAは、従来のキャッシュに加え、ソフトウェアで制御可能なチップ上のメモリ(オンチップメモリ)を持つ。オンチップメモリは、新たに追加されたpage-load/page-storeと呼ばれる命令を用いることで、オフチップメモリとのデータ転送が行なわれる。従って、

ユーザによる明示的なデータ配置、データの置き換えが可能となり、個々のプログラムに対し最適なデータ転送の制御を行わせることができる。

現在、SCIMAのメモリアクセス機構として、従来のプロセッサに簡単な拡張を施すことで実現できる構成を検討し、その評価を行なっている⁴⁾。

本稿では、従来のメモリアクセス機構に対し、SCIMAの拡張機構が面積・遅延に与える影響を評価することで、検討するメモリアクセス機構の妥当性を示す。また、本機構を用いることで、面積・遅延の影響を考慮しても、これまで確認されているSCIMAの有効性が変わらないことを示す。

2. SCIMAの概要

図1に、SCIMAの構成を示す。

SCIMAでは、チップ上のメモリとして、キャッシュに加えオンチップメモリを搭載する。キャッシュはハードウェア制御によりデータ配置・置き換えが行われるのに対し、オンチップメモリは、ソフトウェアでデータ配置・置き換えの指定が可能である。

2.1 アドレス空間

SCIMAでは、論理アドレス空間上にオンチップメモリ領域をマップする。オンチップメモリは大きな連続ブロック領域であるため、この管理をTLBではなく

[†] 東京大学 先端科学技術研究センター
Research Center for Advanced Science and Technology,
The University of Tokyo

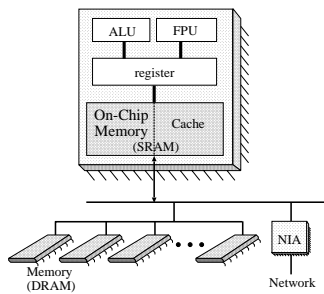


図1 SCIMAの構成

専用レジスタで行ない、TLB ミスの頻発を防ぐ。導入されるレジスタは、オンチップメモリの開始アドレスを保持する ASR (On-Chip Address Start Register) とオンチップメモリ容量を保持する AMR (On-Chip Address Mask Register) である。

2.2 拡張命令

SCIMA では、page-load/page-store と呼ぶオンチップメモリ-主記憶間の転送命令を ISA (命令セットアーキテクチャ) 上に備える。この命令により、オンチップメモリのデータ配置・置き換えをソフトウェアで行うことが可能となる。また、本命令は、ブロックストライド転送機能を備える。データ転送元の開始番地、データ転送先の開始番地、転送サイズ、ブロック幅、ストライド幅の 5 オペランドを用いたブロックストライド転送機能により、不連続なデータをオンチップメモリ上の連続領域に転送させることができるため、無駄なデータ転送を省き、チップ内の記憶領域を有効に利用可能である。

オンチップメモリ領域は *page* と呼ぶ複数のブロックに分割され、この *page* を単位として管理する。*page* のサイズは 2 のべき乗である。page-load/page-store 命令で転送できる最大データサイズはこの *page* のサイズであり、*page* を跨いでの転送はできない。

2.3 キャッシュ・オンチップメモリ統合機構

SCIMA では、キャッシュとオンチップメモリをハードウェア的に統合し、それらに割り振られる容量比を、対象とするアプリケーションの性質に合わせて実行時に再構成できる⁵⁾。本機構では、従来のキャッシュに対し、way 単位でキャッシュ・オンチップメモリの属性を設け、データ参照を制御する。属性は WLR (Way Lock Register) に保持され、参照する way の判別に用いられる。

3. SCIMA のメモリアクセス機構

3.1 R10000 のメモリアクセス機構

本稿で評価する SCIMA のメモリアクセス機構は、MIPS R10000 プロセッサ³⁾ の構成をベースとする。図2に、R10000 上でのメモリアクセス命令のパイプライン動作を示す。メモリアクセス命令の動作を中心に、R10000 の命令実行動作について述べる。

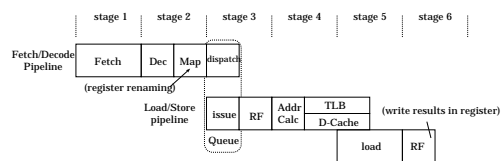


図2 MIPS R10000 のパイプライン動作

R10000 では、命令のフェッチ・デコード後、命令の種類に応じた命令 queue に格納される。本稿ではこの動作を *dispatch* と呼ぶ。load/store 命令の場合は *Address queue* と呼ばれる queue に dispatch される。なお、Address queue は circular FIFO 構造をしており、命令の格納および削除は in-order に行なわれ。次に queue に dispatch された命令から実行可能なものが選択され、演算器へ発行される。本稿ではこの動作を *issue* と呼ぶ。issue された命令は、各演算器で命令が実行され、結果をレジスタに書き込み実行を終了する。

メモリアクセス命令の中で、load 命令はオペランドが揃い次第 issue され、out-of-order に実行される。ここで、先行する store 命令が同じアドレスに対するものであった場合、メモリの RAW ハザードが生じる。しかし R10000 では、先行する store 命令との依存関係の判定はせずに投機的に load 命令の実行がなされる。

一方、store 命令は precise interrupt の保証のため、先行する命令が全て完了した後に初めて issue される。この時、後続の load 命令とのアドレスの比較を行なう。アドレスの一致した load 命令が既に実行されていた場合は、その load は不正な値を読み出していることになるため、当該 load 命令以降の全命令を破棄して実行をやり直す。なお、store 命令は命令順に issue されることから、WAW ハザード、WAR ハザードは生じない。

3.2 SCIMA のメモリアクセス機構

3.2.1 メモリアクセス制御

SCIMA では、従来のメモリアクセスに加え、以下の制御が必要となる。

- load/store 命令のオンチップメモリ参照
- page-load/page-store 命令の制御
- load/store 命令と page-load/page-store 命令のメモリアクセス順序保証

以下、各制御の実現方法について述べる。

load/store 命令のオンチップメモリ参照

SCIMA では、キャッシュを参照する load/store 命令とオンチップメモリを参照する load/store 命令は ISA 上で同一であり、参照アドレスによってキャッシュ/オンチップメモリへの参照を判別する必要がある。この制御は 2.1 節で述べた ASR と AMR を用いて行なう。

page-load/page-store 命令の制御

page-load/page-store 命令は ISA 上に新たに追加される命令であり、この命令を処理するためのハードウェア機構の拡張が必要となる。よって、*PLS queue* (*Page-*

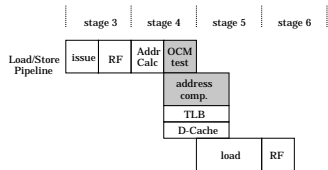


図3 load命令のパイプライン動作

Load/page-Store queue)と呼ぶqueueを新たに設ける。デコード・リネームされたpage-load/page-store命令は、このqueueにdispatchされる。PLS queueはaddress queueと同様にFIFO構造をしており、in-orderにエントリの追加・削除が行われる。

メモリアクセス順序保証

R10000におけるメモリアクセスの順序保証については3.1節で述べた。SCIMAにおいても、load/store間の順序保証はR10000と同様に解決できる。また、store、page-load、page-store命令はいずれもメモリの内容を書き換える命令であり、precise interruptの保証上in-orderに実行するため、順序保証は問題とならない。一方、load命令とpage-load/page-store命令については検討する必要がある。性能への影響を検討した結果、page-load/page-store命令に対してメモリ依存のない後続のload命令を先行してissueできる機構を実現する必要がある⁴⁾。この時、load命令とpage-load/page-store命令間での順序保証が必要となる。

今回、page-load/page-store - load命令間の順序保証はload命令の実行時に行なう。よって、load命令が自身に先行するpage-load/page-store命令を識別するためのマスクビットを導入する。load命令は、命令実行の際に、マスクビットを用いて識別した先行page-load/page-store命令との依存判定を行なう。依存していた場合、実行を中断して後に再実行する。

3.2.2 メモリアクセス命令の動作

R10000に対しSCIMA用追加制御を加えた、load/store命令のパイプライン動作を図3に示す。図2のstage 3以降の動作を示しており、図3の斜線部が追加動作である。OCM testでオンチップメモリアドレスかどうかを判定し、address comp.で先行page-load/page-store命令との依存判定を行なう。以上の追加動作は従来のメモリアクセス動作と並列に行なわれる。

page-load/page-store命令はPLS queue格納後、転送元・転送先アドレスのオペランドが利用可能になり次第レジスタファイルを参照し、アドレスを取得して後述するPLS address stackに格納する。その後、先行命令が全て実行完了した後にissueされ、全オペランドの情報を取得し転送を開始する。転送終了後、PLS queueのエントリを削除する。

4. メモリアクセス機構の設計

本章では、設計したSCIMAのメモリアクセス機構

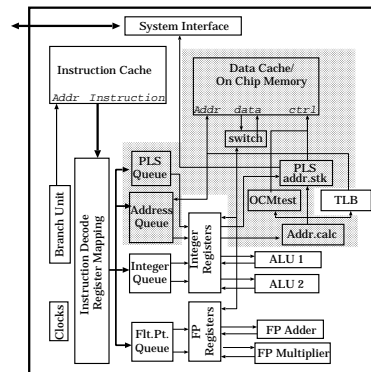


図4 設計範囲

の動作について述べる。設計対象とする範囲を図4の斜線部で示す。今回の設計では、SCIMAでの追加拡張機構に対象を絞って設計を行なった。設計した各機構の動作を以下に詳述する。

wakeup & select logic

以降では、address queue、PLS queue、およびselect logicをまとめてwakeup & select logicと表記する。

SCIMAではpage-load/page-store命令用の命令queueとしてPLS queueを新たに設ける。また、今回の設計では、load/store命令とpage-load/page-store命令の実行において共有するリソースが多いことから、load/store命令とpage-load/page-store命令のissueポートを統合する。

Address queue

Address queueはR10000に存在するload/store命令専用の命令queueであり、16エントリのcircular FIFO構造をしている。Address queueの各エントリにおける制御は、以下の通りである。

1. デコード後の命令が1つdispatchされる
2. issue条件が揃うとready信号を立てる
3. select信号が返ると演算器へ命令をissueする
4. 実行完了信号を受け、queueの先頭に達した時点でエントリを開放する

SCIMAの拡張として、page-load/page-store命令との依存等、issueの条件が増えている。また、実行情報として、先行するpage-load/page-store命令を識別するマスクビットを各エントリに保持する。

PLS queue

PLS queueはSCIMAの追加機構であり、page-load/page-store命令専用の命令queueである。今回作成したPLS queueは4エントリのcircular FIFO構造をしている。

PLS queueの各エントリにおける制御は、Address queueと同様である。ただし、issue条件および1エントリのビット数がAddress queueと異なる。

select logic

select logicは命令queueのready信号を検知し、その中から発行を許可する命令を選択する。

今回 SCIMA への拡張を施した select logic では、前述の issue ポート統合のため、Address queue と PLS queue の両者より ready 信号を受け取り、その中から 1 つを選択する構成となっている。

PLS address stack

PLS address stack は、SCIMA で追加される機構で、PLS queue とエントリ数が同じ queue であり、各エントリは PLS queue のエントリと一意に対応している。

page-load/page-store 命令は取得した転送元・転送先アドレスを PLS address stack に保持する。保持したアドレスは、転送開始時に利用する。一方、load/store 命令の実行時には、エントリ内のアドレスとのアドレス比較により、page-load/page-store 命令との依存判定 (図 3 の address comp. 動作) を行なう。

OCM test 機構

OCM test は SCIMA で追加される機構である。OCM test では、2.1 節で述べた ASR、AMR および WLR を用いてオンチップメモリアドレスかどうかの判定を行なう。同時に、オンチップメモリアドレスであればキャッシュ・オンチップメモリの参照 way を示す信号を作成する。

cache data select

データキャッシュをキャッシュ・オンチップメモリ統合機構として扱うため、従来のデータキャッシュが持つ data select 部を SCIMA 用に拡張する。

従来の n-way セットアソシアティブキャッシュにおいて、どの way のデータを取得するかはタグ比較の結果によりなされる。これに対し、SCIMA への拡張では OCM test/OCM test および address comp. の結果によりオンチップメモリアドレスであると判定された場合には、タグの判定結果を OCM test 機構より送られた way 選択信号で置き換える制御を追加する。

5. 評価

5.1 面積・遅延の評価方法

本章では、SCIMA 用拡張機構がサイクルタイムおよび面積に与える影響を評価する。R10000 に準じたメモリアクセス機構 (cache model)、および 4 章で述べたメモリアクセス機構 (SCIMA model) をそれぞれ Verilog での RTL 記述により設計し、その面積と遅延を調べる。評価には VDEC 提供の $0.35\mu\text{m}$ プロセスライブラリを用いる。

5.2 面積・遅延の評価結果、考察

論理合成ツール上で、速度を最優先として合成した結果を表 1 に示す。なお、address calculator および cache decoder は SCIMA model においても cache model のものと同じとする。

合成の際には、モデル間共通の目標値に対する速度最適化を行なっている。なお、面積に配線領域は含まれていない。遅延は、fan-out 数に応じた配線負荷の

影響を含む。また、遅延については、評価基準の統一のため、全ての機構でフリップフロップの setup - hold 時間を無視している。

wakeup & select logic

cache model と SCIMA model で Address queue を比較すると、SCIMA model での面積が約 1.13 倍に増えている。これは、SCIMA model の Address queue では、マスクビットの保持、制御の変化等により 1 エントリが保持するビット数が増えていることが直接影響している。同様に PLS queue の面積をみると、PLS queue のエントリ数は Address queue の 1/4 であり、1 エントリのビット数は Address queue に比べ少ない。これらの影響により、Address queue より面積が小さくなっている。また、モデル間で select logic の面積を比較すると、SCIMA model の select logic は 2 つの queue に対する選択および調停機構の追加により面積が増大している。

次に Address queue の遅延を見ると、SCIMA model で遅延がわずかに伸びている。これは、page-load/page-store 命令の依存等、load/store 命令の wakeup 条件が R10000 に比べ複雑になるためである。また、SCIMA model での select logic の遅延は、調停の処理の影響によりわずかに増大している。PLS queue の遅延は Address queue の遅延に比べ小さく、Address queue からの wakeup 動作に隠蔽されるため、PLS queue がサイクルタイムに直接影響をおよぼすことはない。

OCMtest & PLS address stack

OCMtest 機構はオンチップメモリアドレスかどうかの判定のための比較器が主である。面積・遅延ともに小さい。

PLS address stack のエントリ数は PLS queue と同一であるが、参照アドレスの保持のため queue 1 エントリあたりのビット数が PLS queue に比べ多く、PLS queue より面積が増えている。一方、内部の制御は PLS queue に比べ単純であり、遅延は PLS queue より短くなっている。

cache data select

SCIMA model では、キャッシュでのタグ比較から一致した way のデータを選択する回路に対して外部信号を選択する制御が増えており、遅延がわずかに増大している。なお、両モデルの面積を比較すると、SCIMA modelの方が面積が小さい。これは SCIMA model での制御が増え遅延が伸びたことで、同一回路内の他のパスへの遅延制約が緩くなったためと思われる。

5.2.1 サイクルタイムへの影響

R10000 のパイプライン構成に合わせてサイクルを区切ると、[Address queue/PLS queue] と select logic で 1 サイクル、address calc. と [cache decode/OCM

表1 面積・遅延の評価(速度最優先)

cache model	area [mm ²]	latency [ns]	SCIMA model	area [mm ²]	latency [ns]
wakeup & select logic			wakeup & select logic		
- Address queue	1.64	2.07	- Address queue	1.86	2.27
- select logic	0.03	4.04	- select logic	0.05	4.21
			- PLS queue	0.37	1.80
address calc. logic	0.15	3.21			
			OnChipMemory Control logic		
			- OCM test logic	0.09	1.43
			- PLS address stack	0.49	1.59
cache			cache/OnChipMemory		
- decode logic	0.28	1.82			
- data select logic	2.50	3.31	- data select logic	2.43	3.50

test/PLSstack]で1サイクル, SRAM macroの参照とcache data selectで1サイクルとなる。

SRAM macroは設計対象外であり,表1に載っていないが,SRAM macroの遅延が2.8[ns]以上であれば,SRAM macroとcache data selectがサイクルタイムとなる。一方,表1中で最もlatencyが長いのはwakeup & select logicであり,メモリアクセス機構の中ではwakeup & selectとSRAM macro & data selectがクリティカルパスの候補となる。

wakeup & selectがクリティカルパスであった場合,サイクルタイムには4.9%程度の影響があり,SRAM macro & data selectがクリティカルパスであった場合,SRAM macroを両モデルとともに2.8[ns]と仮定すると,サイクルタイムには3.1%程度の影響があることになる。よって,SCIMA用拡張がサイクルタイムに影響を及ぼしたとしても最大で4.9%である。

5.2.2 面積への影響

R10000のフロアプラン¹⁾から求めたAddress queueの面積は総面積の約2.7%である。表1より,SCIMA modelにおける追加拡張機構によって,面積はcache modelでのAddress queueの約1.7倍となるが,それでも総面積の4.8%である。すなわち,SCIMA modelでの増分は約2%であり,面積に大した影響はない。

5.3 PLS queue エントリ数についての検討

PLS queueはSCIMA modelで新たに追加した機構であり,page-load/page-store命令用の命令queueである。PLS queueのエントリ数は多ければ特に面積への影響が大きく,サイクルタイムにも影響を及ぼす可能性がある。少なければ,page-load/page-store命令のdispatchができずにストールを頻発してしまい,性能が向上しない。よって,5.2節の評価ではエントリ数を4としているが,最適なエントリ数を性能と遅延・面積の両面から検討する必要がある。

評価コードには行列積とCGを用いる。いずれも,ソフトウェアパイプラインングにより転送と演算をインターリーブするコードとなっている。行列積,CGの評価にはサイクルレベルのシミュレータを用いる。

表1中のAddress queue, select, PLS queue, OCM test, PLS address stackが対象

表2 評価に用いるパラメータ

同時実行命令数	
- 整数演算	2
- 浮動小数点演算(積和)	1
- 浮動小数点演算(除算/平方)	1
- ロード・ストア	1
命令queueサイズ	
- integer, FP, load/store	各16
Active list エントリ	48
キャッシュラインサイズ	32B, 128B
pageサイズ	4KB
オフチップメモリスループット	2B/cycle
オフチップメモリーテンシ	80cycle
メモリサイズ 合計	64kB
- cache model	キャッシュ 64kB(4-way)
- SCIMA model	キャッシュ16kB(1-way)
(software pipelining)	オンチップメモリ 48kB

評価に用いるパラメータを表2に示す。

図5(a)にPLS queueのエントリ数に対する面積,遅延の変化および行列積とCGの性能の関係を示す。PLS queueのエントリ数が1, 2, 4, 8, 16の場合を評価している。面積は,今回設計した機構全体にPLS queueのエントリ数の変化が与える影響を評価している。遅延は,クリティカルパスをwakeup & selectとしてサイクルタイムを求めている。行列積・CGはそれぞれ,エントリ数が1の場合に対する相対性能を表す。

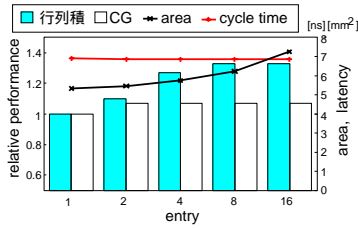
面積は,エントリ数に対しほぼ線形に面積が増大している。一方,サイクルタイムをみると,クリティカルパスはAddress queueのissue動作であり,PLS queueのエントリ数は直接影響しない。

行列積・CGによる性能の評価に着目すると,演算とインターリーブするpage-load/page-store数は,行列積の場合通常2~3で,局所的に6程度まで増加し,CGの場合は2以上にはならない。よって,それぞれエントリ数が8と2の時点で性能が飽和している。

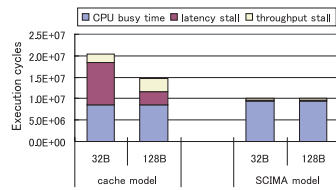
図5(a)と考察から,行列積とCGに対してエントリ数は4~8程度で十分である。一方,エントリ数が16以下の場合,サイクルタイムには影響しないものの,面積はエントリ数に比例して増えてしまう。よって,設計の際に選択したエントリ数の4は妥当と思われる。

5.4 総合評価

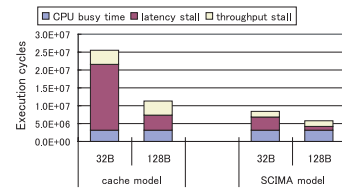
5.2, 5.3節の結果を受け,サイクル数と面積・遅延の両面を考慮したSCIMAの全体性能を評価する。



(a) PLS queue エントリ数 vs. 面積, 遅延, 相対性能



(b) 総合評価 (行列積)



(c) 総合評価 (CG)

図5 評価結果

5.4.1 評価方法

5.2 節で検討した追加拡張機構のサイクルタイムへの影響を考慮し, cache model と SCIMA model でのモデル間の相対評価を行なう。評価にはサイクルレベルシミュレータを用いる。また, 評価におけるパラメータは表 2 に従う。PLS queue エントリ数は 4 である。

評価では, プロセッサの実行時間を CPU-busy time (T_b), latency-stall (T_l), throughput-stall (T_t) の 3 つに分類する。CPU-busy time とはプロセッサが実際に計算処理を行っている時間であり, latency-stall は主記憶のアクセスレイテンシがもたらすストール時間を, また throughput-stall はオフチップメモリのスループット不足に起因するストール時間を指す。

ここで, T_b, T_l, T_t を, プロセッサの総実行時間 T , オフチップメモリスループットが無限大とした場合の実行時間 T_∞ , オフチップメモリスループットが無限大かつオフチップメモリレイテンシが 0 とした場合の実行時間 T_p を用い, 以下のように定義する。

$$\begin{aligned} T_b &= T_p \\ T_l &= T_\infty - T_p \\ T_t &= T - T_\infty \end{aligned}$$

面積の影響は微小であり, 性能へ影響しないものとする。サイクルタイムの伸びは CPU の動作速度に影響し, SCIMA model での CPU 動作時間は cache model に比べ 1 サイクルあたり 4.9% 長くなる。一方, メモリ参照のレイテンシ・スループットはメモリの性能によるため, サイクルタイムの変化は影響しない。よって, latency-stall, throughput-stall については両モデル間で変化しないものとし, CPU-busy time が 4.9% 伸びるものとして実行時間の相対評価を行なう。

5.4.2 評価結果, 考察

評価結果を図 5(b), 5(c) に示す。ただし, 縦軸は cache model のサイクル数を表す。SCIMA model は CPU-busy time が 5% 増加した結果, 増加しないと仮定した場合に比べ, 行列積で 4.5%, CG で 2.7% 程度実行時間が長くなる。しかし, 本来の SCIMA の

利点であるメモリアクセスの最適化により, latency-stall と throughput-stall で表されるオフチップメモリアクセス時のペナルティが大幅に削減されている。従って, 全体性能で見ると, cache model に対して SCIMA model での実行時間が大幅に短縮され, ラインサイズが 32B の時に行列積で 103%, CG で 197% 性能が向上し, ラインサイズが 128B の時に行列積で 45%, CG で 95% 性能が向上している。

6. まとめ

今回, 高い性能を得られる SCIMA のメモリアクセス機構を, 従来のプロセッサに対する拡張機構として設計し, その面積・遅延の影響を調べた。その結果, 面積への影響はほとんどなく, サイクルタイムへの影響は高々 5% 程度である。総合評価により, 本機構を用いることで十分な性能を得られることがわかった。

謝辞 本研究での設計と評価に関しアドバイスを頂いた, 東京大学 南谷教授, 今井助手, 小沢博士に感謝致します。なお, 本研究の一部は日本学術振興会 未来開拓学術研究推進事業「計算科学」(Project No. JSPS-RFTF 97P01102) によるものである。

参考文献

- 1) Ali Ahi, et al. R10000 superscalar microprocessor. In *Proc. HOT Chips VII*, August 1995.
- 2) D. Callahan and A. Porterfield. Data Cache Performance of Supercomputer Applications. In *Proceedings of Supercomputing '91*, pp. 564–572, 1990.
- 3) Kenneth C. Yeager. The MIPS R10000 superscalar microprocessor. *IEEE Micro*, Vol. 16, No. 4, pp. 28–40, August 1996.
- 4) 大根田 拓, 近藤正章, 中村宏. SCIMA におけるメモリアクセス機構の検討. 情報処理学会研究報告, No. ARC-144, pp. 165–170, 2001.
- 5) 近藤正章, 中村宏, 朴泰祐. SCIMA における性能最適化手法の検討. 情報処理学会論文誌, Vol. 42, No. SIG12(HPS4), pp. 37–48, 2001.
- 6) 中村宏, 近藤正章, 大河原英樹, 朴泰祐. ハイパフォーマンスコンピュータ向けアーキテクチャ SCIMA. 情報処理学会論文誌, Vol. 41, No. SIG5(HPS1), pp. 15–27, 2000.

ラインサイズ 128B の場合。