

3次元積層型ヘテロジニアスプロセッサのための シミュレータ開発とその応用

小島 拓也[†] 池添 赳治[†] 天野 英晴[†]

[†] 慶應義塾大学大学院 理工学研究科 223-8522 神奈川県横浜市港北区日吉 3-14-1

E-mail: †{tkojima,tikezoe,hunga}@am.ics.keio.ac.jp

あらまし 近年、IoT デバイスや組み込み機器で要求される機能、性能、インターフェースが多様化している。これを、単一の SoC(System On a Chip) で満たすのはコストの観点から現実的ではない。そこで、複数のチップを LSI の 3 次元積層技術を用いてシステムを構築する SiP(System In a Package) が注目されている。なかでも、TCI(Thru-Chip Interface) はチップ間で無線通信を可能とする技術であり、積層のためのコストが小さい。TCI を用いることでホストプロセッサに加えて電力効率に優れるアクセラレータコアを 3 次元的に積層し、協調動作させることで用途に応じて多種多様なシステムを提供することができる。こうしたシステムの性能を評価する際に Verilog をはじめとするハードウェア記述言語を用いたシミュレーションを用いると所要時間が長く、また、メモリバンド幅などのシステム構成を変更して評価を行うのが容易ではなかった。そこで、本研究では MIPS R3000 互換プロセッサをホストプロセッサとするサイクルアキュートなシミュレータを開発し、柔軟なシステムの評価を可能にした。

キーワード CGRA, 粗粒度再構成可能アーキテクチャ, 3次元積層 LSI, TCI, サイクルアキュートシミュレータ

Takuya KOJIMA[†], Takeharu IKEZOE[†], and Hideharu AMANO[†]

[†] Graduate School of Science and Technology, Keio University Hiyoshi 3-14-1, Kohoku-ku, Yokohama, Kanagawa, 223-8522 Japan

E-mail: †{tkojima,tikezoe,hunga}@am.ics.keio.ac.jp

1. はじめに

近年、組み込み機器や IoT デバイスで要求される性能や機能が多様化している。したがって、用途毎に単一の SoC(System On a Chip) を設計、製造するのは NRE コストの高さゆえ現実的な手段ではない。そこで、プロセッサコアやメモリ、IO インターフェースなどのモジュール化された比較的小さな LSI チップを製造し、それらを組み合わせた SiP(System In a Package) によるソリューションが注目されている。用途に応じて利用するモジュールを選択すれば良いため、短期間でシステムの設計ができる。一方で、機能モジュールの製造ではチップのダイサイズが小さくなることにより、歩留まりの向上が期待できる。

SiP 技術は大きく分けて、シリコンインターポーザも用いた 2.5 次元実装と、TSV(Through Silicon Via) などのチップ間通信技術を用いた 3 次元実装が存在する。2.5 次元実装では複数のダイを平面状に配置するためスケラビリティに限界があるが、TSV を用いた 3 次元実装では、複数のチップを垂直方向へ積層していくため、チップ同士はより高密度に接続され高い転送バンド幅を得る。さらに、3 次元実装ではシステムを小さな

パッケージへ高密度に集積できるため、配線長の削減などによる高性能化も得られる。一方で、TSV は特殊な製造工程を必要とし、チップ単体の製造コストが増加してしまう。加えて、ハイパフォーマンス向けのプロセッサなどを積層する場合、放熱の問題が顕在化する。したがって、現在 TSV を用いた 3 次元実装技術が利用される製品はハイパフォーマンス向けのメモリ技術である HMC [1] や HBM [2] などに限られる。

対して、3 次元積層のための技術としてチップ間無線通信インターフェース TCI(Thru-Chip Interface) が提案されている [3]。TCI ではコイルの磁束変化を用いて通信を行うため、TSV のような物理的接続が不要であり、通信用のコイルは CMOS プロセスの配線層を用いるため設計に特殊な EDA のサポートや、専用の製造プロセスを必要としない。TCI を利用したシステムのプロトタイプとして汎用のプロセッサやアクセラレータを積層した Cube-1 [4] や Cube-2 [5] が報告されている。

このような TCI を積層したヘテロジニアスなマルチコアシステムを設計し、利用する場合にいくつか課題が存在する。現在、アプリケーションの動作検証には設計資産の RTL を用いたシミュレーションしかできない。したがって、実用的で複雑なア

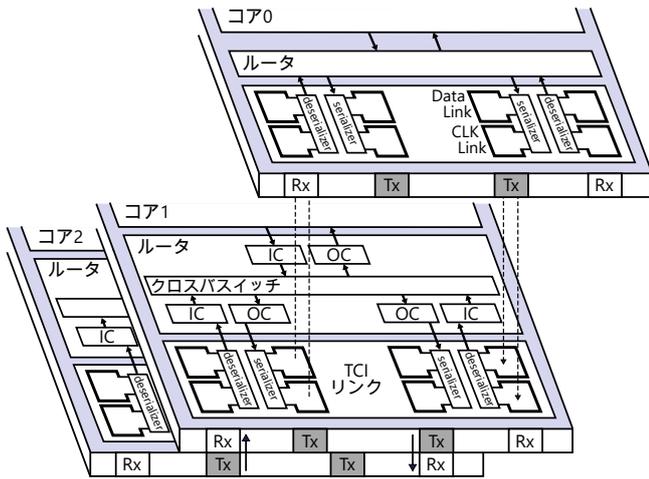


図1 TCI-IPによるエスカレーターネットワーク

アプリケーションを想定した場合、シミュレーションにかかる時間とデバッグの困難なさが問題となる。また、RTLシミュレーションを利用する場合、メモリアクセス性能やCPUのキャッシュサイズ、アクセラレータのロジック規模などのパラメータを変更しての設計探索を行うのが困難となる。そこで、本研究ではTCIを用いた3次元積層システム向けのサイクルアキュレートシミュレータを開発し、ケーススタディとしてJPEGのエンコードを行った場合のシステム評価を行う。

2. ビルディング型計算システム

ビルディング型計算システムは複数のLSIチップがTCIによって相互接続されたシステムである。TSV(Through-Silicon-Via)のような物理的接続を必要としないため、チップ製造後に用途に応じて複数のチップを選択し、柔軟に組み合わせることができるシステムである。すでにCube-1[4]とCube-2[5]がプロトタイプとして開発され、評価及び動作検証が行われている。Cube-1はMIPS R3000互換の汎用CPU Geyser[6]と低消費電力CGRA(Coarse-Grained Reconfigurable Architecture)のCMA(Cool Mega Array)[7]を3次元に積層して構成したシステムとして報告された。Cube-2ではTCIによる通信機構をIP化し[8]、CMA以外にCNN(Convolutional Neural Network)アクセラレータSNACC[9]とNon-SQLデータベースアクセラレータKVS[10]が利用可能なアクセラレータとして追加された。

2.1 TCIによるチップ間通信

誘導結合を利用した通信を行うために、TCIではメタル層に長方形のコイルを形成する。1つのコイルは送信用または受信用として利用される。送信および受信チャンネルを形成するために、各チャンネルは同期用クロックとデータ用の2つのコイルを二つを持つ。Cube-1では積層チップが単方向リングネットワークを形成していたが、Cube-2では図1に示すように、上方向送信チャンネル(Tx)、受信チャンネル(Rx)および下方向Tx、Rxの計4チャンネルを用いたエスカレーターネットワーク[11]を形成する。つまり、エスカレーターネットワークを形成するのに1チップあたり計8個のコイルが必要となる。一つのチャンネルは最大で8Gbpsのデータ転送を0.14pJ/bitで行うことができ、

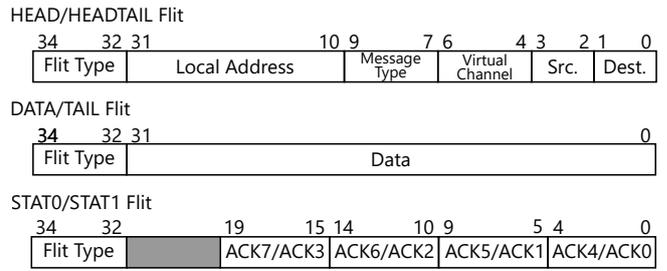


図2 Cubeシステムにおけるパケットフォーマット

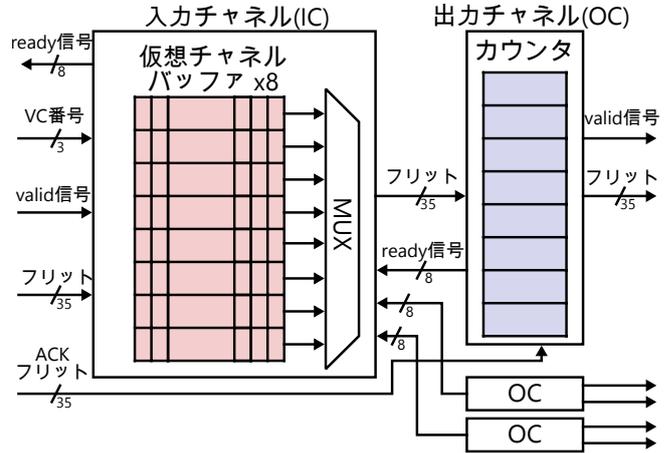


図3 ルータの入出力チャンネル

10^{-12} 以下のビットエラー率でシリアル通信が可能である[12]。

2.2 ルータによるフロー制御

TCI-IPにはSERDESおよび2.5GHzの内部発振回路が含まれ、最大50MHzで動作する35bit幅のチップ間データ転送チャンネルとして設計されている。エスカレーターネットワークでは図2に示す35bitを1フリットとする可変長パッケージがTCI-IPを経由して他のチップへ転送される。データの読み出しおよび書き込みにはシングル転送モード(1word/4Byte転送)とブロック転送モード(16word/64Byte転送)が用意されている。シングル転送モードではHEADフリット+TAILフリットの2フリットで1つのパッケージを構成し、ブロック転送モードではHEADフリット+15個のDATAフリット+TAILフリットの計17フリットで1つのパッケージを構成する。TCI-IPを搭載したチップは図1に示すように3つの入力チャンネル(IC)と3つの出力チャンネル(OC)を持つルータが備わっている。3つのチャンネルはそれぞれ、チップ内のコアとのデータ転送、上層チップとのデータ転送、下層チップとのデータ転送に利用される。各チップには最上層のチップから順にIDが振られ、ルータはヘッダフリットに含まれる宛先ID(Dest.)を元に経路を決定する。現在の設計ではヘッダフリットのチップID用フィールドは2bitであり、最大4枚のチップを積層可能である。

図3にルータの入出力チャンネルの概要を示す。1つの入出力チャンネルは物理的には1つのリンクであるが、仮想的に8つのチャンネル(VC0~VC7)に分割されている。入力チャンネルには各仮想チャンネルごとにバッファが存在し、経路が決定したパッケージのうちもっとも高い優先度を持つ仮想チャンネルが出力チャンネルにデータを転送する。現在の設計ではVC0がもっとも高い

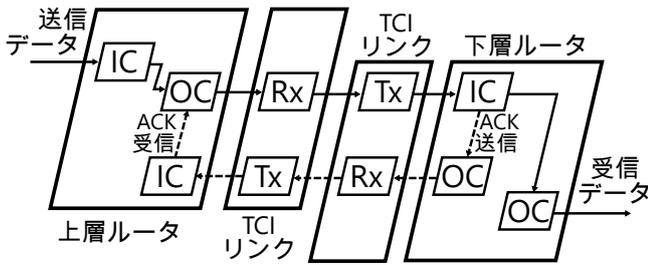


図4 チップ間のデータ転送の様子

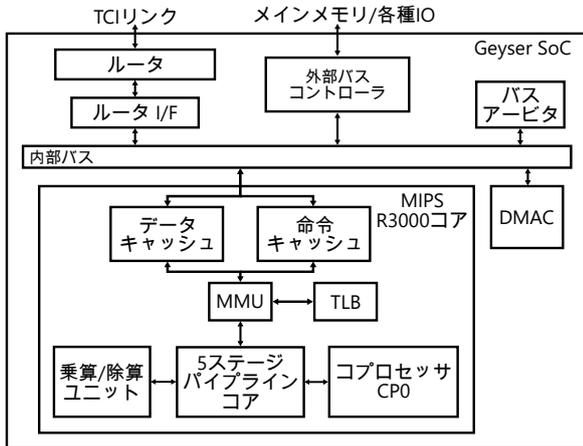


図5 Geyserの構成

優先度を持ち、VC0 から VC7 の順に優先度が低くなる。

TCIは無線接続であるため、転送先の入力チャネルバッファに空きがあるかどうかをピギーバックによって確認している。図4にはある上層チップからすぐ下の下層チップへデータを転送し、ACKを受け取る様子が示されている。実線で表される流れが転送データで、破線で表される流れがピギーバックのためのACKである。ピギーバックには図2に示したSTAT0またはSTAT1フリットを用いている。この例では、上層の出力チャネルがパケットを送信すると、そのフリット数だけカウンタを増やす。下層の入力チャネルは届いたパケットをバッファし、次の転送先へデータの転送が完了すると、出力チャネルにACKの送信を行うよう指示を出す。例えば、このデータ転送にVC0を利用していたとすると出力チャネルはSTAT0フリットのACK0フィールドに転送したフリット数をセットし、上層へ送り返す。上層の出力チャネルはSTAT0フリットが届くと、フィールドにセットされた数の分だけカウンタをデクリメントする。したがって、出力チャネルはカウンタの値によって転送先入力チャネルに空きがないと判断すると、ready信号を下げる。ただし、チップ内のローカルコアへデータを転送する場合は信号線によるハンドシェイクが可能であるためこれは不要である。また、コア側からデータを入力チャネルに転送する際には、入力チャネル側から出力されるready信号によって、バッファに空きがあるかを確認してからデータを入力する。

2.3 ホストプロセッサ Geyser

本研究ではCube-1およびCube-2システムと同様にGeyserをホストCPUとして利用し、Geyserは常に最上層に積層されるシステムを想定する。前述の通り、GeyserはMIPS R3000

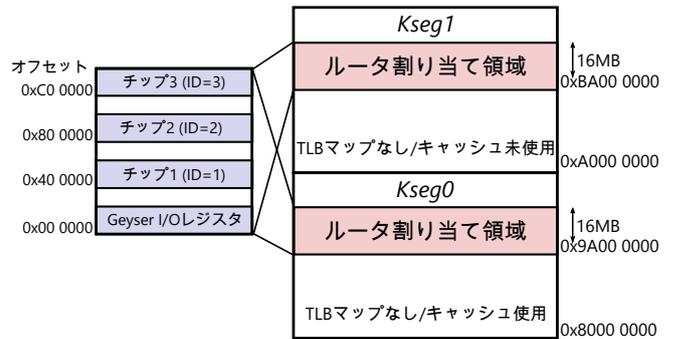


図6 Geyserにおける仮想アドレッシング

命令セットとの互換があるCPUであり、図5に示すようにGeyserは典型的な5段パイプラインのコアと8KBの2-wayセットアソシアティブな命令キャッシュとデータキャッシュ、システム制御用プロセッサCP0、乗除算ユニットなどから構成される。CP0やTLBによって仮想メモリのサポートや例外処理などを行うことが可能で、Linux OSなどの汎用OSも動作可能である。MIPS R3000のコアはDMACやルータとともに内部バスで結ばれている。MIPS R3000の命令セットに加えて、GeyserにはMIPS32互換のキャッシュ命令を備えており、明示的にキャッシュを操作することができる。

MIPS R3000における32bitの仮想アドレス空間は2GBのkuseg、0.5GBのkseg0とkseg1、1GBのkseg2に分割されている。このうち、ルータおよび積層されたチップはkseg0およびkseg1の一部にマップされている。図6はGeyserのkseg0とkseg1のアドレスマップの詳細を表している。それぞれ、0x9A00_0000と0xBA00_0000から16MB確保されており、CPUコアがこれらの領域へ読み出し要求や書き込み要求を出すと、それがルータインタフェースへ伝わり、パケットに変換される。ルータインタフェースは16MB内のオフセット(24bit)のうち上位2bitで、宛先のチップIDを決定する。読み出し要求の場合、ルータインタフェースにデータが到着するまでCPUコアはストールする。一方で、書き込み要求の場合はパケットが送信され次第ストールが解消する。kseg0、kseg1ともにTLBを用いたアドレス変換を行わない領域となっており、カーネルモードでのみアクセス可能な領域となっている。kseg0へのアクセスはキャッシュが利用されるのに対し、kseg1へはそのままアクセスされる。Cubeシステムではkseg0のアドレスで積層チップにアクセスするとキャッシュミスが発生し、ルータはブロック転送モード(17フリット)でデータをフェッチする。したがって、キャッシュラインサイズはブロック転送モードのデータサイズと同じ64Byteとなっている。一方、kseg1のアドレスで積層チップにアクセスすると、キャッシュを用いずそのままシングル転送モードでデータが転送される。

3. サイクルアキュレートシミュレータの開発

あるシステムでアプリケーションの実行時間を見積もるためにだけにRTLシミュレーションを行うのは過剰であり、また、システムの規模が大きくなるにつれて所要時間も非常に大きく

表 1 Geyser におけるストール発生要因

ステージ	要因	ストールサイクル
IF ステージ	キャッシュミス	不定
	メモリストール	不定
ID ステージ	データハザード	1 サイクル
EX ステージ	構造ハザード (CP0)	最大 5 サイクル
	構造ハザード (積算器)	最大 3 サイクル
	構造ハザード (乗算器)	最大 9 サイクル
MEM ステージ	キャッシュミス	不定
	メモリストール	不定

なる。さらに、キャッシュやバス幅などのシステム構成をパラメータ化して、設計探索を行うのは容易ではない。アプリケーション自体のデバッグ環境が乏しいという問題もある。そこで、本研究では 2. 節で説明したパケット交換方式を用いた 3 次元積層システムのサイクルアキュレートシミュレータを開発し、評価を行う。

MIPS R3000 用のシミュレータとしてオープンソースの VMIPS [13] が公開されている。C++ 言語で記述されており、必要に応じてメモリマップドデバイスの追加や、ハードウェア割り込みなどが容易にカスタム可能である。また、GDB によるリモートデバッグをサポートする。VMIPS では命令セット実行パートにおけるサイクル数に関しては遅延スロットも含め正確にエミュレートされているものの、メモリアクセスは理想化されており、全てが 1 サイクルで完了する想定となっている。そのため、キャッシュのシミュレーションは事実上存在しない。また、正しくパイプラインをシミュレーションしているわけではない。

本研究では VMIPS をベースに下記の拡張を行う。

- 5 段パイプラインの正確なシミュレーション
- メモリアクセスレイテンシのパラメータ化
- カスタム可能な n-way セットアソシアティブキャッシュ
- バスアービタの追加
- ルータおよび周辺回路の追加
- 積層チップ用の抽象クラス

3.1 パイプラインシミュレーション

VMIPS ではキャッシュミスなどによるストールが起きない想定となっている。一方で、Cube システムのような積層システムでは全体の処理に対してデータ転送にかかる時間が小さくない。したがって、CPU コアのストールを正しくシミュレーションする必要がある。一方で、ストールの発生要因はステージごとに異なる。表 1 に Geyser におけるストール発生要因と発生するステージを示す。MIPS R3000 では遅延分岐を採用しているため制御ハザードは起きない。Geyser ではデータハザードの多くはフォワーディングにより解決しているが、直前のサイクルでメモリからロードした値を EX ステージで利用する場合 1 サイクルストールする。また、動作周波数の低下を防ぐために、乗算と除算、またコプロセッサ命令は各々 3 サイクル、9 サイクル、5 サイクルのマルチサイクル実行となっており、構造ハザードによるストールが発生し得る。キャッシュミスを含むメモリアクセスに起因するストールは読み出しの場合

はデータが届き次第、または書き込みの場合データを送信し次第ストールが解消するため、固定のサイクル数ではない。特に、ルータ経由で積層チップへアクセスする場合、トラフィックの混雑状況にも依存する。

シミュレータは C++ で記述されたプログラムとして実行されるため、当然シーケンシャルに実行される。そのため、パイプラインステージを上から順に実行すると、後方のステージで発生したストールに対処できない場合がある。そこで、本シミュレータでは各ステージをハードウェアの状態を変更しないパートと変更するパートに分離し、前者でストール発生の有無を確認する。すべてのステージでストールが発生しない場合、後者のパートを実行する。また、例外の取り扱いに関しても同様の問題が発生する。例えば、前方のステージで発生した例外 (e.g. アドレスエラー例外や予約命令例外) を発生した直後にハンドルしてしまうと、本来実行されるはずであった先行するステージが実行されなくなってしまう。または、先行するステージがそのすぐ後に例外を起こす場合、それを先に処理しなくてはならない。そこで、本シミュレータでは発生した例外は一旦ペンディングされ、MEM ステージのメモリアクセスまたは、WB ステージのレジスタ書き込みが行われる前にペンディングされた例外がハンドルされる。

3.2 キャッシュ、メモリアクセス

前述の通り、本シミュレータではメモリアクセスをより実システムに近い形でシミュレートできる。一般に、メモリアクセス要求を出してからデータを受け取るまでに複数サイクル所要するケースが多い。本シミュレータではこの所用サイクル数をパラメータ化し、様々な設計に対応できるようになっている。また、内部バスにつながるモジュールはバスアービタによって、排他的にアクセスされるようシミュレートされる。

現在の Geyser の設計では命令キャッシュ、データキャッシュともに 64B キャッシュラインを 64 ブロック \times 2way 計 8KB の容量を持つ。しかし、本シミュレータではこれらの構成は全てパラメータ化されており、対象システムに最適なキャッシュ構成を探索することが可能である。同様に内部バスのデータ幅もパラメータ化されており、例えばキャッシュがメインメモリからデータをフェッチする際に 1 サイクルに複数ワードのデータを受け取ることが可能な構成をシミュレーションすることができる。

3.3 ルータおよび積層されるチップ

本シミュレータには 2. 節で述べたチップ間のデータ転送を行うルータや Geyser におけるルータインターフェースも含まれる。ルータの内部はパイプライン化されており、次のルータへデータが転送されるには最短でも 4 サイクル所要する。ルータによるデータ転送はこのようなパイプライン処理や、仮想チャネルなども含め正確にシミュレートされる。また、仮想チャネルのバッファサイズもパラメータ化されている。

積層されるアクセラレータコアを容易に追加できるように、ルータとのインターフェースが実装済みの抽象クラスを提供する。したがって、システム設計者はこのクラスを継承し、アクセラレータコアの動作部を記述しオンチップメモリなどの構成

表 2 シミュレーション時間の比較 (秒)

	RTL シミュレーション	本シミュレータ
Geysler 単体	27.024	0.115
Geysler+CMA	24.750	0.171
Geysler+CMAx2	29.231	0.174

などを指定するだけで、新たなアクセラレータをプロセッサと積層してシミュレーションすることが可能となる。

4. 評価

本研究で開発したシミュレータを用いて、Geysler と低消費電力 CGRA の CMA [7] を積層したシステムを対象に評価を行う。CMA は演算処理ユニットである PE(Processing Element) を 2次元のアレイ状に持ち、タスクに応じて PE アレイの構成を変更する再構成可能デバイス的一种である。

評価には静止画像の JPEG エンコードを用いる。CMA をアクセラレータとして用いる場合、Geysler は処理データの転送や演算結果の回収、PE アレイの再構成を含む各種制御を行う。本評価ではシステム構成として 1)Geysler 単体、2)Geysler+CMA、3)Geysler+CMAx2 の 3 種類を用いる。

4.1 シミュレーションの所要時間

はじめに、本シミュレータと RTL シミュレーションの所用時間を比較する。3つのシステム構成ごとにそれぞれシミュレーションを行い、その所要時間を測定する。RTL シミュレーションでは Verilog HDL で記述された設計に Cadence 社 NC-Verilog(INCISIVE 15.20.020) を用いた。いずれのシミュレーションも Intel Core i5-4250U 1.30GHz、8GB DDR3-SDRAM のマシン環境で測定を行った。表 2 に結果をまとめる。最大で約 234 倍のシミュレーション時間短縮を達成している。実際の設計では読み出しと書き込みで異なるアクセスレイテンシであるのに対し、本シミュレータではこれを均一であると見積もっている。したがって、シミュレートされる実行時間には、これによる誤差が含まれるもののその影響は非常に小さい。

4.2 内部バスバンド幅の検討

すでに実チップとして実装されている Geysler の内部バス幅は 1ワード (4Byte) であるが、本シミュレータを用いて、このバス幅を 1ワード、2ワード、4ワードと変更して実行時間の変化を確認する。2.節で説明した通り、現在の TCI-IP では 1クロックサイクルに 35bit のデータを転送する。バス幅の拡張に合わせて、図 7 に示した TCI リンクを検討する。1レーンあたり 1フリットのデータを転送可能であり、図 7(a) は従来の IP に実装されているものである。バス幅が 2ワードの場合は 2レーン構成を利用し、バス幅が 4ワードの場合は 4レーン構成を利用する。

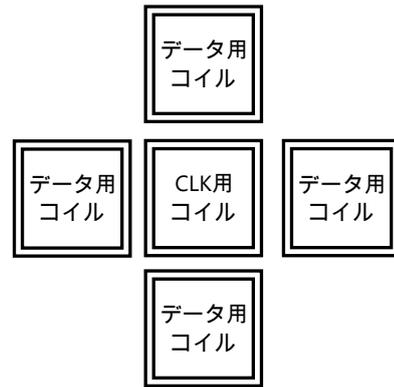
図 8 に各バス幅での実行時間の変化を示す。ここでは、メモリアクセスレイテンシは 8 サイクルに設定している。従来のバス幅 1ワードの場合でも、CMA を 1チップ、または 2チップ利用することで、実行時間が短縮されている。Geysler 単体で実行した場合と比べ、CMA1チップに処理をオフロードすると約 1.36 倍の高速化が得られる。しかし、並列可能な箇所に限界があるため、CMA を 2チップ利用しても CMA を 1チップしか



(a) 1レーン (従来設計)



(b) 2レーン



(c) 4レーン

図 7 複数レーン TCI リンクの検討

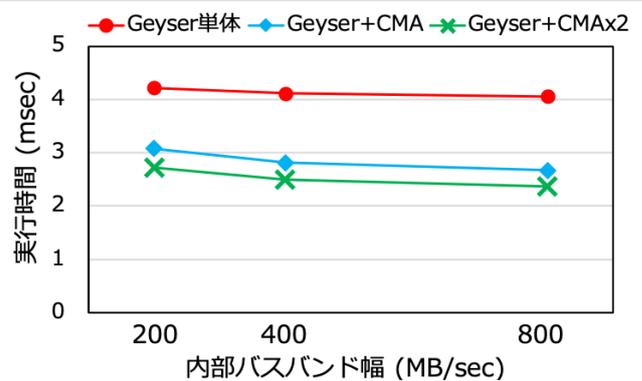


図 8 バスのバンド幅を変更した場合の実行時間比較 (50MHz)

表 3 キャッシュミス率の比較

	命令キャッシュ	データキャッシュ (WB 率)
Geysler 単体	0.14%	0.46% (38.16%)
Geysler+CMA	0.22%	1.60% (69.07%)
Geysler+CMAx2	0.27%	1.79% (54.35%)

利用しない場合と比べ、1.13 倍程度の高速化しか得られない。

4.3 キャッシュ構成の検討

本評価で用いたプログラムでは積層チップ間でデータを転送するのに、キャッシュ命令を用いている。例えば、CMA にデータを送る場合、送りたいアドレス空間のキャッシュブロックを確保し、送信データを一度キャッシュ上に用意する。準備が完了すると、キャッシュ命令を用いて意図的にライトバックを発生させ、ルータを経由してブロック転送される。しかし、この方法では表 3 に示すようにキャッシュミス率が増加してし

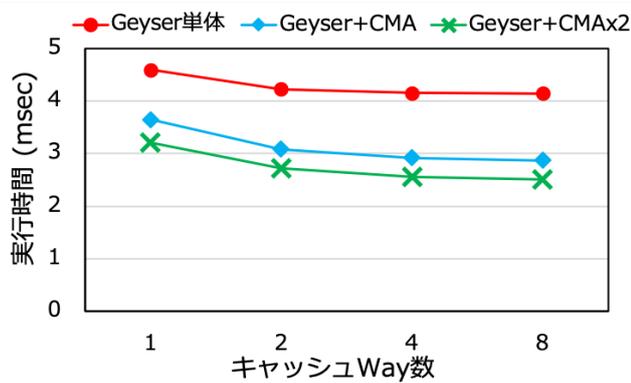


図9 データキャッシュのway数を変更した場合の実行時間比較 (50MHz)

まう。

そこで、最後にキャッシュの構成を変更し、実行時間に与える影響を評価する。図9にデータキャッシュのway数を変化させた場合の実行時間を示す。ただし、way数を増やすことによる動作周波数の低下などをここでは考慮していない。Geysers単体では従来の2wayセットアソシアティブで十分低いミス率であったため、4way、8wayと変更してもほとんど実行時間が変わらない。一方で、CMAを用いる場合、前述の理由からway数を増やすと実行時間が短くなっている。しかし、その削減効果は徐々に小さくなることが確認された。

5. 結論と今後の課題

本研究では誘導結合TCIを用いた3次元積層システム向けのシミュレータを開発した。本シミュレータではホストプロセッサとしてMIPS R3000互換のGeysers CPUを想定し、メモリアクセスレイテンシやキャッシュミスの振る舞いをサイクルレベルで正確にシミュレートすることが可能となった。また、積層チップ間のデータ転送を担うルータ部もシミュレータに含まれている。本シミュレータはオープンソースのVMIPSをベースとしており、デバッグ機能などはそのまま利用することが可能である。

評価を行なった結果、本シミュレータを用いることでRTLシミュレーションと比べ最大で約234倍の高速化を達成した。また、GeysersとアクセラレータとしてCMAを組み合わせたシステムで、JPEGエンコードを行なった場合の実行時間の見積もりを行なった。本シミュレータではバス幅やキャッシュの構成がパラメータ化されており、これらを変更が容易である。この機能を用いて、システムの構成変更が実行時間に与える影響を定量的に評価することができるようになった。

本評価ではキャッシュサイズやバス幅の変更がもたらす電力の変化や、チップ面積の変化などを考慮に入れていない。今後は、これらの見積もりも可能にすることでより本シミュレータはシステムの設計探索に有用となると期待される。

謝 辞

本研究は、科学技術振興機構戦略的研究推進事業 (JST), CREST, JPMJCR19K1 および JSPS 科研費 (B) ビルディン

グブロック型計算システムにおけるチップブリッジを用いた積層方式 (18H03125)、JSPS 科研費 3次元積層技術を応用した粗粒度再構成可能デバイスの研究 (19J21493) の支援を受けたものである。また、東京大学大規模集積システム設計教育研究センターを通し、日本ケイデンス株式会社の協力で行われたものです。関係者の皆様に感謝致します。

文 献

- [1] J. T. Pawlowski: "Hybrid memory cube (HMC)", 2011 IEEE Hot chips 23 symposium (HCS)IEEE, pp. 1-24 (2011).
- [2] J. Standard: "High bandwidth memory (hbm) dram", JESD235 (2013).
- [3] Y. Take, H. Matsutani, D. Sasaki, M. Koibuchi, T. Kuroda and H. Amano: "3D NoC with inductive-coupling links for building-block SiPs", IEEE Transactions on Computers, **63**, 3, pp. 748-763 (2012).
- [4] Y. Koizumi, N. Miura, E. Sasaki, Y. Take, H. Matsutani, T. Kuroda, H. Amano, R. Sakamoto, M. Namiki, K. Usami, et al.: "A scalable 3D heterogeneous multi-core processor with inductive-coupling thruchip interface", IEEE Micro (2013).
- [5] S. Terashima, T. Kojima, H. Okuhara, K. Musha, H. Amano, R. Sakamoto, M. Kondo and M. Namiki: "A Preliminary evaluation of building block computing systems", 2019 IEEE 13th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MC-SoC)IEEE, pp. 312-319 (2019).
- [6] L. Zhao, D. Ikebuchi, Y. Saito, M. Kamata, N. Seki, Y. Kojima, H. Amano, S. Koyama, T. Hashida, Y. Umahashi, et al.: "Geysers-2: The second prototype CPU with fine-grained run-time power gating", Proceedings of the 16th Asia and South Pacific Design Automation ConferenceIEEE Press, pp. 87-88 (2011).
- [7] N. Ozaki, Y. Yasuda, M. Izawa, Y. Saito, D. Ikebuchi, H. Amano, H. Nakamura, K. Usami, M. Namiki and M. Kondo: "Cool Mega-Arrays: Ultralow-Power Reconfigurable Accelerator Chips", IEEE Micro, **31**, 6, pp. 6-18 (2011).
- [8] 松下 悠亮, 増山 滉一朗, 野村 明生, 門本 淳一郎, 四手井 綱章, 黒田 忠広, 天野英晴: "誘導結合ワイヤレスチップ間接続のIP化 (集積回路)", 電子情報通信学会技術研究報告 信学技報, **116**, 364, pp. 7-12 (2016).
- [9] R. Sakamoto, R. Takata, J. Ishii, M. Kondo, H. Nakamura, T. Ohkubo, T. Kojima and H. Amano: "The design and implementation of scalable deep neural network accelerator cores", 2017 IEEE 11th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)IEEE, pp. 13-20 (2017).
- [10] Y. Tokusashi, H. Matsutani and H. Amano: "Key-value Store Chip Design for Low Power Consumption", 2019 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)IEEE, pp. 1-3 (2019).
- [11] A. Nomura, H. Matsutani, T. Kuroda, J. Kadomoto, Y. Matsushita and H. Amano: "Vertical packet switching elevator network using inductive coupling ThruChip interface", 2016 Fourth International Symposium on Computing and Networking (CANDAR)IEEE, pp. 195-201 (2016).
- [12] N. Miura, H. Ishikuro, T. Sakurai and T. Kuroda: "A 0.14 pJ/b inductive-coupling inter-chip data transceiver with digitally-controlled precise pulse shaping", Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE InternationalIEEE, pp. 358-608 (2007).
- [13] B. Gaeke: "The VMIPS Project, Version 1.5.1", <http://vmips.sourceforge.net/vmips/> (2019).