Vol.2016-ARC-223 No.1
2016/11/28

スケーラブルなディープラーニング向けアクセラレータチップの設計と評価

高田 遼[1]　石井 潤[1]　坂本 龍一[1]　近藤 正章[1]　中村 宏[1]　大久保 徹以[2]　小島 拓也[2]　天野 英晴[2]

概要：
近年，組込みシステムにディープラーニング技術を導入することが期待されており，電力効率に優れたディープラーニングアクセラレータの開発が重要な課題となっている．しかしながら，これまで提案されてきたアクセラレータは 2 次元畳込み演算などの特定のネットワーク構造向けに最適化されたものや，データアクセス削減のためにネットワーク構造を工夫するものが多い．対象とするネットワーク構成が限定され汎用性に課題がある．そこで我々は，多様なネットワーク構成に柔軟に対応可能で電力効率の良いアクセラレータのアーキテクチャを検討している．今回，命令により動作するマイクロコントローラと SIMD 型積和演算器から構成される電力効率の良いコアを設計し，マルチコア構成のアクセラレータに対しスケーラビリティの評価を行った．

1. はじめに

近年，組込みシステムでディープラーニング技術を利用することが期待されている．一例として，自動車やモバイル機器などでは，学習済みの畳込みニューラルネットワーク (Convolutional Neural Network: CNN) を用いて画像認識を行いつつ種々の機能を提供することが実用化されつつある．そこで，CNN などのディープラーニングによる識別高速化を組込みシステムの電力制約下で実現するために，電力効率に優れたディープラーニング専用アクセラレータの開発が重要な課題となっている．

ディープラーニング向けのアクセラレータによる CNN 識別高速化では，畳込み層における 2 次元畳込みの演算処理と全結合層における外部メモリとのデータ転送を考慮することが重要となる．前者は演算ボトルネックに，後者はメモリボトルネックとなりやすいためである．畳込み層の 2 次元畳込みの演算の高速化・省電力化の研究としては，Chen らの Eyeriss[1] がある．Eyeriss は組込みシステム向けの CNN アクセラレータで，2 次元的に配置した演算ユニット間のオンチップネットワークやチップ内バッファなどを工夫することで外部メモリへのアクセス回数を減らし，消費電力を削減する手法を提案している．一方で，全結合層の外部メモリとのデータ転送に対しては，多数の先行研究が存在する．DaDianNao[2] はチップ内に大量の eDRAM を配置することで，データ転送自体を不要とするアプローチをとっている．また，EIE[3], [4] では不要な学習済みパラメータの省略や，学習済みパラメータ 1 つあたりのビット数の削減を行うことで，推定精度を落とさずに学習済みニューラルネットワークモデルのデータサイズを圧縮している．

しかしながら，これらの従来研究では消費電力削減のために畳込み層などの特定のネットワーク構成向けに最適化したアクセラレータや，データアクセス削減のためにネットワーク構造に手を加える研究が多い．対象とするネットワーク構成が限られる可能性もあり，進化を続けるディープニューラルネットワークの多様なネットワーク構成を扱うための柔軟性が課題となる．それに対して，我々は多様な種類のネットワーク構成に対し，省電力生を失わずかつ柔軟に対応可能なアクセラレータアーキテクチャを検討しており，そのベースとなる高電力効率かつ命令制御可能なコアの設計やアーキテクチャ探索を行っている．本発表では，提案するアクセラレータの全体構成とコアの内部構成・命令セットについて述べた後，マルチコア構成のアクセラレータに対しスケーラビリティの評価を行った結果について報告する．

2. 畳込みニューラルネットワーク

一般的に，CNN は畳込み層とプーリング層を交互に積層したあと全結合層を 3 層ほどの積層する構成をとる．7 層の CNN であれば図 1 のようなネットワーク構成となる．

[1] 東京大学
[2] 慶應義塾大学

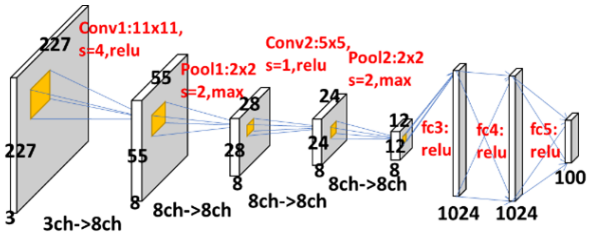

図 1　7 層畳込みニューラルネットワーク

畳込み層とプーリング層，全結合層における演算は式 (1)〜(5) のように定義される．

$$a_{n_o}[i,j] = \sum_{n_i}\sum_{p}\sum_{q} \omega_{n_i,n_o}[i,j]x[i+p,j+q] + b_{n_o} \quad (1)$$

$$y_{n_o}[i,j] = f(a_{n_o}[i,j]) \quad (2)$$

$$y[i,j] = \max_{p,q}(x[i+p,j+q]) \quad (3)$$

$$a[i] = \sum_{j} \omega[i,j]x[j] + b_i \quad (4)$$

$$y[i] = f(a[i]) \quad (5)$$

$f(\cdot)$ は活性化関数と呼ばれ，ReLU 関数 $f(x) = \max(0, x)$ や sigmoid 関数 $f(x) = 1/(1 + \mathrm{e}^{-1})$ がよく用いられる．

ここで，畳込み層と全結合層の比較を行う．畳込み層は 227×227 の RGB 画像に 11×11 のフィルタをストライド 5 で畳込み，8 枚の出力マップを得る．また，全結合層は 396 入力 396 出力である．16bit 固定小数点形式を仮定した場合，処理対象データはどちらも 300kB 程度であるにも関わらず，積和演算回数を比較すると前者が 2928200 回で後者が 156816 回である．従って，畳込み層の演算強度は 9.30 で演算ボトルネックとなりやすく，全結合層の演算強度は 0.498 でメモリボトルネックとなりやすい．なお，演算強度は 16bit 固定小数点形式の処理対象データ 1Byte あたりの積和演算回数・MAX 演算回数と定義する．

3. アクセラレータのアーキテクチャ

3.1 全体構成

本研究では，マイクロコントローラと SIMD 型積和演算器を主な構成要素とするコアを複数搭載したマルチコアアクセラレータを検討している．4 コア構成のアクセラレータを図 2 に示す．各コアは，命令メモリ (inst)，ストリームバッファ (sbuf)，データメモリ (dmem)，ルックアップテーブル (lut)，データ出力用メモリ (omem) の 5 つのメモリを持つ．基本メモリ構成としては別々のアドレス空間を持つ分散メモリシステムであるが，CNN を始めとした多層のニューラルネットワークを複数コアで実行する際，演算結果をコア間で共有する必要があるため，出力用メモリの omem はコア間で共有する．

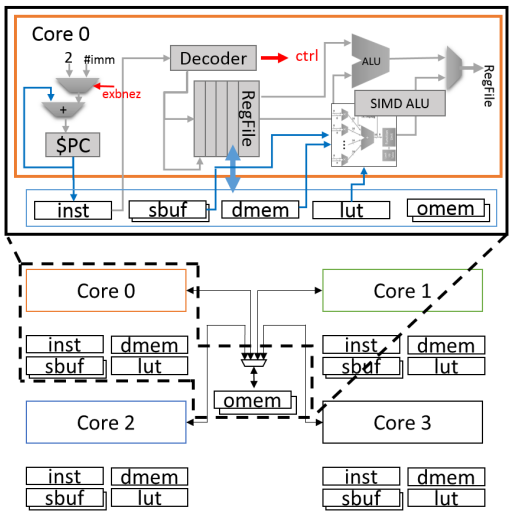

図 2　4 コア構成のアクセラレータ

3.2 コアのアーキテクチャ

コアは回路規模の小さなマイクロコントローラと SIMD 型積和演算器から構成される．マイクロコントローラは 16bit 固定長の命令セットにより動作する．命令長が短いために実装可能な命令の機能は単純なものに限られるが，命令デコーダや制御回路も単純化されるため，小型で高電力効率なコントローラとなっている．

一方で，ディープラーニング専用アクセラレータでは，膨大な積和演算を効率よく実行できることが重要である．そこで，本研究のコアには後述する SIMD 型積和演算器と独自のカスタム SIMD 算術命令を実装した．

メモリ構成

各コアが持つ 5 つのメモリ (inst, sbuf, dmem, lut, omem) は 32bit のアドレス空間に割りつけられており，load/store 命令によって全てのメモリとレジスタファイル間でデータをやり取りできる．inst は命令メモリで，sbuf と dmem は処理対象データ用メモリである．ただし，sbuf と dmem はデータを直接 SIMD 型積和演算器に供給するため 64bit 幅のデータバスを持つ．lut はニューラルネットワークの活性化関数に利用し，omem は出力データ用である．

ディープラーニングアクセラレータの先行研究では演算ユニットに入力用 2 つと出力用の 3 つのバッファが接続された構成をとるものが多い．本研究も同様で，SIMD 型積和演算器と sbuf, dmem, omem が用意されている．これは，ニューラルネットワークの主要な演算が学習済み重みパラメータと各層の入出力データの積和計算であるためである．ただし，先行研究では入力用の 2 つのバッファを学習済み重みパラメータ用のバッファと入力用データ用のバッファとして用いるが [2]，本研究のアクセラレータは再利用性の低いストリームデータ用のバッファ (sbuf) と再利用性の高いデータ用のバッファ (dmem) として用いる点が異なる．sbuf 側のほうがデータ転送量が多くメモリボトルネックの原因となりやすいため，そこで，ダブルバッ

ファリングを行うことで演算処理と subf へのデータ転送をオーバーラップさせ，実行時間の削減を図る．

以上のようなメモリ構成をとる目的は，畳込み層と全結合層のデータ再利用性に関する特性の違いに対応するためである．畳込み層では学習済みデータの再利用性が高く各層の入力データの再利用性が低いが，全結合層では逆転し各層の入力データの再利用性が高く学習済みデータの再利用性が低いという違いがある．

マイクロコントローラ

提案アーキテクチャのマイクロコントローラはパイプライン 4 段のインオーダー実行で，MIPS に近い形式の 16bit 固定長命令セットを解釈実行する．主要な役割は SIMD 型積和演算器の制御やループの制御，メモリへの load/store で，回路規模の削減・省電力化を図るため浮動小数点演算器・浮動小数点レジスタファイルやその他複雑な制御回路は搭載していない．レジスタファイルは 32bit16 本であるが，そのうち 4 本を SIMD 型積和演算器の演算結果が格納される特殊レジスタに，1 本をプログラムカウンタに割り当てているため，汎用レジスタは 11 本である．演算器は 32bit 長で論理算術演算が可能である．現在の実装では命令パイプライン化が十分ではないが，今後パイプライン化された実装に拡張する予定である．

命令セットアーキテクチャ

本研究のアクセラレータは，ディープラーニングの演算自体は SIMD 型積和演算器を用いて行うが，汎用的な処理も一部実行可能であり，様々なネットワーク構成に柔軟に対応できる．命令形式は表 1 に示した 2 種類で，論理・算術演算，load/store 命令，分岐命令に加え，いくつかの命令を追加している．具体的には，ダブルバッファの切り替え制御命令や DMA 発行命令，SIMD 型積和演算器の制御命令などである．

表 1 命令形式

	4bit	4bit	4bit	4bit
R-type	opcode	rd	rs	function
I-type	opcode	rd	immediate	

特にマルチサイクルのカスタム SIMD 算術命令を定義しており，ディープラーニングの積和演算を行う際の制御オーバーヘッドを軽減する．具体的な制御オーバーヘッドとして，処理対象データにアクセスするためのアドレス計算や，ループの制御，条件分岐などの処理が挙げられる．本研究のアクセラレータでは，これらの処理と SIMD 型積和演算の動作シーケンスをハードウェアで実装しマルチサイクルのカスタム SIMD 算術命令に集約している．これは，汎用命令セットでソフトウェア実装するのに比べ，CNN の識別高速化と消費電力削減の両方に効果がある．

SIMD 型積和演算器

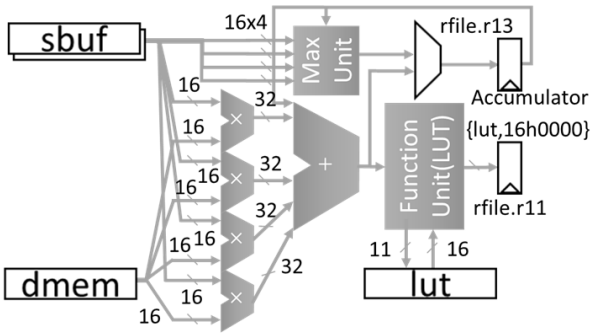

図 3 SIMD 型積和演算器

SIMD 型積和演算器の基本構成を図 3 に示す．SIMD 型積和演算器は 16bit 長データを 4 並列で演算を行うことができ，実行可能な演算はテーブルルックアップ付きの積和演算と MAX 演算である．処理対象データはレジスタファイルを介さずに，sbuf と dmem から直接演算器に供給され，データバスは 64bit 幅である．ルックアップテーブル (lut) はニューラルネットワークの活性化関数に利用する．ディープラーニングアクセラレータの先行研究では活性化関数に ReLU 関数のみをサポートするものもあるが，汎用性の観点からルックアップテーブルによる実装を採用した．また，SIMD 型積和演算器の演算結果は，積和演算や MAX 演算の場合はレジスタファイルの 13 番レジスタに自動的に保存され，ルックアップテーブルの場合は 11 番レジスタに保存される．

積和演算の詳細な動作としては，sbuf と dmem から 16bit 固定小数点形式サイズ 4 のベクトルデータが SIMD 型積和演算器に投入され，その内積内積演算結果が 13 番レジスタにアキュムレートされる．なお，4 つの乗算器はマスクレジスタによって制御可能である．一方，MAX 演算の場合は，sbuf から供給された 16bit 固定小数点形式データ 4 つと現在の 13 番レジスタの値の MAX 演算結果を 13 番レジスタに保存する．こちらも sbuf から供給された 4 つのデータに対しマスクレジスタによる制御が可能である．前述のマルチサイクルのカスタム SIMD 算術命令は，積和演算や MAX 演算を sbuf・dmem アドレスをインクリメントしながら指定回数連続実行する．

また，SIMD 型積和演算器は図 4 に示すように 8bit 長データ 8 並列に拡張する特殊な動作モードを持つ．具体的には，図 3 のような 4 並列積和演算器が 2 基並列に搭載されている．sbuf 側は 8bit 固定小数点形式に切り替え，サイズ 8 のベクトルデータとし，2 基の 4 並列積和演算器に 4 つずつ供給する．ただし，乗算器に入力直前に 16bit 固定小数点形式に拡張する．一方 dmem 側は 16bit 固定小数点形式サイズ 4 のベクトルデータのままだが，2 個の 4 並列積和演算器に同一データを供給する．

この動作モードの主目的は，全結合層のメモリボトルネック軽減である．sbuf 側のデータ転送量は dmem 側と比

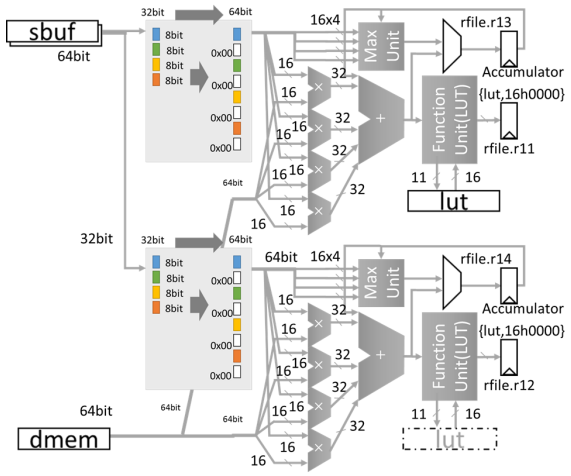

図 4　SIMD 型積和演算器 特殊動作モード

べ非常に大きいため，sbuf 側を 8bit 固定小数点形式とするとデータ転送量が半減し，トータルのデータ転送時間はほぼ半減することが期待される．また，dmem 側は 16bit 固定小数点形式のままで 2 個の 4 並列積和演算器に同一データを供給する必要があるが，dmem 側のデータは再利用性が高いためこの問題は限定的である．以上のような仕組みにより，実質的な演算強度が向上し，メモリボトルネックを軽減することができる．

4. 評価実験

4.1 評価アプリケーション

今回の評価に用いたアプリケーションは，ILSVRC[5] で使用される ImageNet データセット 100 クラス分類問題を行う 7 層 CNN (図 1) である．ネットワーク構成は入力側から順に畳込み層 1 (conv1)，プーリング層 1 (pool1)，畳込み層 2 (conv2)，プーリング層 2 (pool2)，全結合層 3 (fc3)，全結合層 4 (fc4)，全結合層 5 (fc5) で，入力層は 227 × 227 ピクセルの RGB 画像，出力層は 100 次元のベクトルとなる．7 層 CNN の処理対象データサイズと演算強度，発行命令数を表 2 に示す．データサイズは 16bit 固定小数点形式を仮定しており，発行命令数は本研究の命令セットで実装した場合のものである．

4.2 評価環境

評価環境として，データ転送時間を含めた 7 層 CNN の実行時間を見積もるシミュレータを作成した．ただし，inst，lut のメモリサイズを 2kB，sbuf，dmem，omem のメモリサイズを 64kB と仮定したため，7 層 CNN を 161 個のタスクに分割して複数コアで並列実行する．また，メインプロセッサの主記憶とコアのメモリ間のデータ転送におけるバンド幅とレイテンシは全コアのメモリで同一であると仮定する．シミュレータのパラメータは，アクセラレータ動作周波数，コア数，実行する命令列，処理対象データサイズ，DMA データ転送のバンド幅とレイテンシである．なお，DMAC が対処できる DMA リクエスト数は 1 つのみであるが，ブロードキャスト転送が可能である．例えば全てのコアの dmem や lut に同一データを転送する場合に用いる．また，このシミュレータは sbuf におけるダブルバッファリング機能を考慮している．

表 2　7 層畳込みニューラルネットワークの構成

7 層 CNN の構成	Data Size (16bit fixed point)		演算強度	発行命令数
入力	227x227 (RGB 画像)	300kB		
畳込み層 1: conv1	学習済みパラメータ	6kB	9.29	519552
	出力 55x55 8ch	47kB		
Pooling 層 1: pool1	出力 28x28 8ch	300kB	0.518	70136
畳込み層 2: conv2	学習済みパラメータ	3kB	7.30	88836
	出力 24x24 8ch	9kB		
Pooling 層 2: pool2	出力 12x12 8ch	2kB	0.50	13176
全結合層 3: fc3	学習済みパラメータ	2MB	0.499	308
	出力 1024	2kB		
全結合層 4: fc4	学習済みパラメータ	2MB	0.499	346
	出力 1024	2kB		
全結合層 5: fc5	学習済みパラメータ	200kB	0.499	287
	出力 100	200B		

4.3 評価結果

7 層 CNN のシミュレーションを行い，本アクセラレータのスケーラビリティを評価した．パラメータは DMA データ転送のバンド幅と実行コア数である．ただし，アクセラレータ動作周波数は 50MHz，DMA データ転送バンド幅のレイテンシは固定値として 2usec を設定した．横軸を DMA データ転送のバンド幅，縦軸を 7 層 CNN の実行時間として，各実行コア数評価した結果を図 5 に示す．評価結果から，スケーラビリティを得るには約 500MB/s 以上の DMA データ転送バンド幅必要であること，8 コア以上でスケーラビリティを得られていないことが分かった．

前者は，DMA データ転送バンド幅が低速の場合は，7 層 CNN 全体の実行時間が全結合層 (fc3, fc4, fc5) で律速してしまい，かつその全結合層がスケールできていないためである．一般的に，メモリボトルネックな fc3, fc4, fc5 は演算ボトルネックな畳込み層 (conv1, conv2) よりもスケールし

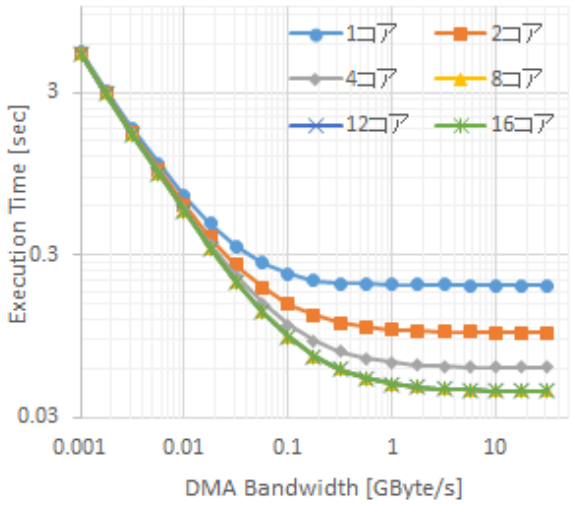

図 5　スケーラビリティ評価

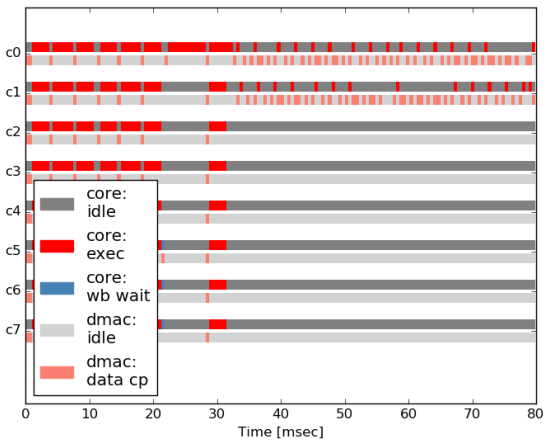

図 6　各コアの動作状況 (8 コア, Bw:100MB/s)

にくく，fc3, fc4, fc5 のスケールには DMA データ転送バンド幅が 500MB/s 以上必要であることが今回のシミュレーションから分かった．また，実行時間において支配的な演算が全結合層から畳込み層に逆転する DMA データ転送バンド幅を調べたところ，こちらも約 500MB/s であった．

後者は，実装依存の問題であり，conv1 の最大スレッドレベル並列度が 8 に制限されているためである．conv1 はアクセラレータ実行時間に占める割合が高く，スケールした際に CNN 全体の実行時間に与える寄与が 7 層中最大である．従って，conv1 がスケールできない 8 コア以上ではスケーラビリティを得られなかった．今回の実装では，conv1 を 48 個のタスクに分割しているが同時に実行可能なタスク数は最大 8 タスクまでとなっている．この制約は出力マップ数 8 の conv1 を出力マップ並列で実装したことによるものであり，命令列のプログラミングを工夫することで改善が可能である．

5. 考察

評価結果の解析のためシミュレータのログから各コアの動作状況を調査した．図 6 は DMA データ転送バンド幅 100MB/s，8 コア構成で 7 層 CNN を実行した際の各コアの動作状況である．横軸は経過時間 [msec] で縦軸は各コア番号を示している．各コアそれぞれに 2 つの積み上げ棒グラフが表示されているが，上側がコアの動作状態で下側が DMAC によるデータ転送状況である．上側の積み上げ棒グラフにおいて赤，青，グレーで表示されている区間が，それぞれ実行状態，ライトバック状態，アイドル状態である．実行状態では，コアが命令を解釈実行し演算を行っている．ライトバック状態は演算結果を主記憶にライトバックするため DMA データ転送をリクエストし，完了を待っている状態である．コアは実行状態のあとほぼ毎回ライトバック状態に遷移するのだが，区間が短すぎるため図 6 では確認できない．また，下側の積み上げ棒グラフでは，DMAC がそのコアのメモリにデータ転送を行っている区間を薄い赤，そうではない区間を灰色で表示している．

図 6 では 37msec までが畳込み層とプーリング層 (conv1, pool1, conv2, pool2) で，37msec 以降が全結合層 (fc3, fc4, fc5) である．図では全結合層が 2 コアで並列化されているように見えるが，実際には互い違いになっており並列化できていない．DMA データ転送バンド幅が 100MB/s 程度の場合，畳込み層は既に 8 コアにスケールしているのに対し全結合層は全くスケールできていないがということが分かった．そこで，全結合層がスケールするのに必要な最低バンド幅を調べたところ 500MB/s となった．図 7 は DMA データ転送バンド幅 500MB/s，8 コア構成でシミュレーションを行い，8 コア中コア 0〜コア 3 の 4 コアを表示している．29msec 以降の全結合層が 3 コアまでスケールしていることが確認できる．最後に，16 コア構成で全結合層が全てのコアにスケールするのに必要なバンド幅を調べたところ，10GB/s となった．このときのコアの稼働状況 (図 8) を見ると，全結合層は 27msec から 29msec の区間に該当し，並列化による恩恵がごく僅かである．

可視化の結果，畳込み層と全結合層の並列化について次のようなことが分かった．全結合層は単純な行列・ベクトル積であるため，メモリ帯域さえあれば容易にスレッドレベル並列化を行うことができる．しかしながら，メモリボトルネックでスケールには高速なメモリ帯域が必要な上，そのような高速なメモリ帯域下では畳込み層によって全体の実行時間が律速しているため，全結合層の高速化は重要ではない．それに対し，畳込み層は演算ボトルネックなため低速なメモリ帯域でも十分にスケールし，高速なメモリ帯域でも実行時間の大半を占めるため，畳込み層のスレッドレベル並列化は重要である．

以上の結果を踏まえ，本研究のアクセラレータに必要な

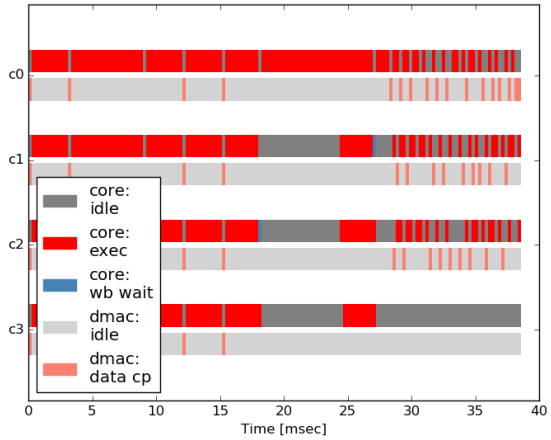

図 7 各コアの動作状況 (8 コア, Bw:500MB/s)

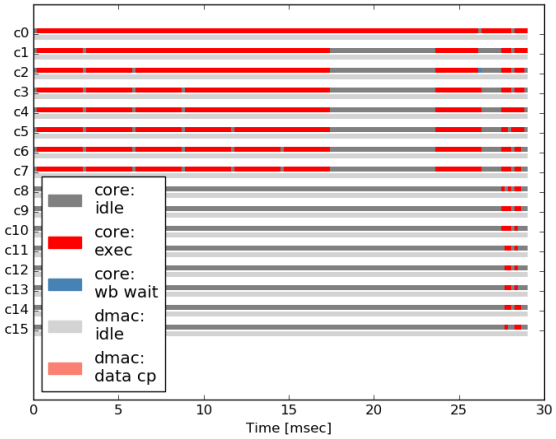

図 8 各コアの動作状況 (16 コア, Bw:10GB/s)

DMA データ転送バンド幅は 1GB/s であることが分かった．図 5 から，さらにスレッドレベルで並列化することを考えないのであれば，これ以上高速なバンド幅は必要ないことは明らかである．また，命令列のプログラミングを工夫した場合や conv1 の出力マップ数を増やした場合など 8 コア以上でもスケーラビリティを得られるアプリケーションで評価を行ったとしても，本評価の仮定のもとでは畳込み層は 500MB/s 程度の DMA データ転送バンド幅で十分にスケールするので，いずれにせよ 1GB/s 以上のバンド幅は必要ないと言える．

6. まとめ

本稿では，高電力効率かつプログラマブルな動作が可能なディープラーニング向けアクセラレータのアーキテクチャを検討した．特に，アクセラレータのベースとなる小型かつ命令制御可能なマイクロコントローラとディープラーニングの積和演算高速化のための SIMD 型積和演算器について述べた．

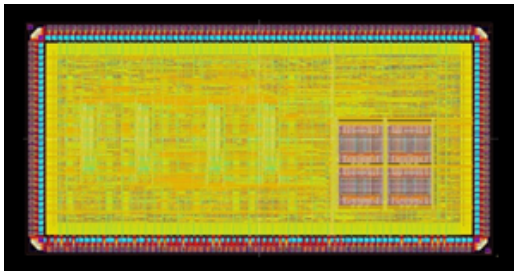

図 9 チップレイアウト図

また，マルチコア構成でアクセラレータの性能を評価するためシミュレータを作成し，スケーラビリティの評価を行った．評価の結果，畳込み層が適切にスレッドレベル並列化された実装であれば，本研究のアクセラレータはスケールアウトによる高速化が可能であり，必要な DMA データ転送バンド幅は 1GB/s であることが分かった．

なお，これまでの検討したアーキテクチャを参考に 4 コア構成のアクセラレータを LSI チップへ実装した (図 9)．1 チップ 4 コア ×3 チップの 12 コア構成となっており，コア間は共有バス，チップ間は磁界結合による 3 次元積層で結合される．チップ面積は 3mm×6mm でテクノロジは Renesas Electronics 65nm SOTB である．

今後の課題としては，設計した LSI チップでの消費電力評価を行う．また，電力効率を評価基準とし，アクセラレータ動作周波数やコア数，SIMD 長など複数のパラメータに対する包括的なアーキテクチャ探索を行う予定である．

謝辞 また，本研究は JSPS 科研費基盤研究 (S) 25220002 の助成によるものである．

参考文献

[1] Chen, Y.-H., Krishna, T., Emer, J. and Sze, V.: Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks, *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, IEEE, pp. 262–263 (2016).

[2] Chen, Y., Luo, T., Liu, S., Zhang, S., He, L., Wang, J., Li, L., Chen, T., Xu, Z., Sun, N. et al.: Dadiannao: A machine-learning supercomputer, *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*, IEEE Computer Society, pp. 609–622 (2014).

[3] Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M. A. and Dally, W. J.: EIE: efficient inference engine on compressed deep neural network, *arXiv preprint arXiv:1602.01528* (2016).

[4] Han, S., Mao, H. and Dally, W. J.: Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding, *CoRR, abs/1510.00149*, Vol. 2 (2015).

[5] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. et al.: Imagenet large scale visual recognition challenge, *International Journal of Computer Vision*, Vol. 115, No. 3, pp. 211–252 (2015).