計測自動制御学会東北支部 第 358 回研究集会 (2026.5.25)
資料番号 358-4

メモリアクセスを考慮したアーキテクチャに基づく MLP-Mixer の FPGA 実装

A Memory-Access-Aware Architecture for FPGA Implementation of MLP-Mixer

キム ソンヒョク, ウィッデヤスーリヤ ハシタ ムトゥマラ, 田中 大介, 張山 昌論

Seonghyeok Kim, Hasitha Muthumala Waidyasooriya, Daisuke Tanaka, Masanori Hariyama

東北大学

Tohoku University

キーワード： MLP-Mixer，Vision Transformer，再構成可能集積回路 (reconfigurable VLSI)，FPGA

連絡先： 〒 980-8579 仙台市青葉区荒巻字青葉 6-3-09 電気・情報系 3 号館 308 号室
キム ソンヒョク，Tel.: (022)795-7155 E-mail: kim.seonghyeok.q5@dc.tohoku.ac.jp

1. はじめに

近年，画像認識において Vision Transformer（ViT）が高い認識精度を達成している [1]．しかし，ViT で用いられるセルフアテンション機構は，入力トークン数の増加に伴って計算量およびメモリアクセス量が増大するため，FPGA 上で効率的に実装する際の課題となる [2, 3]．これに対して，MLP-Mixer は図 1 に示すように，セルフアテンション機構を用いず，トークン方向およびチャネル方向の多層パーセプトロン（MLP）のみで構成される画像認識モデルである．その計算は主に規則的な行列乗算から構成されるため，ハードウェアアクセラレーションに適した構造を有する [4, 5, 6]．

一方，MLP-Mixer のチャネルミキシング部では，拡張されたチャネル次元に対する大規模な行列乗算が必要となる．そのため，高い並列度で

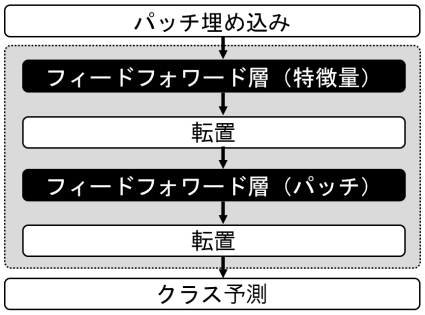

Fig. 1 MLP-Mixer の構造 [7]．

演算を実行するためには，多数の重みデータを供給する必要があり，外部メモリ帯域およびオンチップメモリ容量が実装上の制約となる．特に，大規模な重み行列全体をオンチップメモリに保持することは，多くのメモリリソースを必要とするため困難である．一方で，重みデータを外部メモリから繰り返し読み出す構成では，外部メモリアクセスが性能のボトルネックとなる．

本稿では，メモリアクセスを考慮した MLP-

Mixer の FPGA アーキテクチャを提案する．提案アーキテクチャでは，チャネルミキシング部の大規模な行列乗算に対して，外積演算に基づく 256 個の Processing Element（PE）からなるデータ並列演算構造を用いる．これにより，従来のシストリックアレイで必要となる PE 間の密なデータ転送を削減し，コンパクトな構成で高い空間並列性を実現する．さらに，重み行列全体をオンチップメモリに保持するのではなく，外部メモリから読み出した一部の重みデータのみをオンチップメモリに格納し，複数の演算で再利用する．これにより，オンチップメモリ使用量を抑制するとともに，外部メモリアクセス量を削減する．

提案アーキテクチャを Intel Agilex 7 FPGA 上に実装した結果，497 MHz の動作周波数を達成し，最大リソース使用率は RAM ブロックの 27%であった．また，256 枚の画像に対する処理時間は 2961 ms であった．

2. 提案 FPGA アーキテクチャ

図 2 に，MLP-Mixer のデータフローを示す．MLP-Mixer は，トークン方向の特徴量を混合するトークンミキシング MLP（MLP1）と，チャネル方向の特徴量を混合するチャネルミキシング MLP（MLP2）から構成される．各 MLP は，Layer Normalization，二つの行列乗算，GELU 活性化関数，および残差加算から構成される．

2.1 全体構成

提案アーキテクチャでは，MLP1 および MLP2 を構成する各演算処理をパイプライン化する．図 3 に，一つの MLP に対するパイプライン構成を示す．各演算段の中間結果はオンチップの中間バッファに格納され，後続の演算段で直接再利用される．これにより，中間データを外部メモリへ書き戻す必要がなくなり，外部メモリアクセスを削減できる．

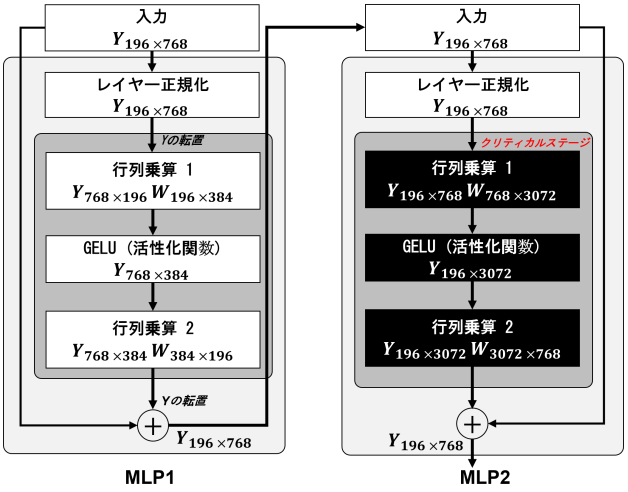

Fig. 2 MLP-Mixer のデータフロー

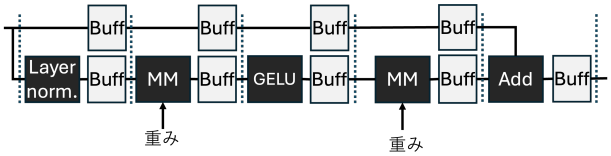

Fig. 3 MLP に対するパイプライン構成

本稿では，MLP-Mixer B/16 モデル [7] を対象として提案アーキテクチャを設計する．表 1 に，対象モデルの構成を示す．入力画像サイズは 224×224 であり，画像を 16×16 画素のパッチに分割することで，196 個のトークンを生成する．各トークンは 768 次元の特徴ベクトルで表現される．トークンミキシング MLP ではトークン次元が 384 まで拡張され，チャネルミキシング MLP ではチャネル次元が 3072 まで拡張される．

表 2 に，各演算処理の計算量を示す．MLP2 の各行列乗算では，約 4.6 億回の積和演算が必要となり，MLP1 の各行列乗算に比べて約 8 倍の計算量を有する．したがって，MLP2 の行列乗算は，MLP-Mixer 全体の実行時間を支配する主要な処理である．

Table 1　MLP-Mixer B/16 モデルの構成

パラメータ	値
モデル規模 / パッチサイズ	B/16
レイヤー数 (L)	12
パッチサイズ (P)	16×16
特徴次元数 (C)	768
トークン数 (S)	196
チャネル MLP 次元 (D_C)	3072
トークン MLP 次元 (D_S)	384
パラメータ数	5900 万

Table 2　各演算処理の計算量

ステージ	処理	計算量
MLP1	Layer Normalization	$150,528 \times k_1$
	行列乗算 1	$57,802,752 \times k_2$
	GELU	$294,912 \times k_3$
	行列乗算 2	$57,802,752 \times k_4$
MLP2	Layer Normalization	$150,528 \times k_1$
	行列乗算 1	$462,422,016 \times k_2$
	GELU	$602,112 \times k_3$
	行列乗算 2	$462,422,016 \times k_4$

k_1, k_2, k_3, k_4 は各演算に依存する定数である．

2.2　メモリアクセスを考慮した行列演算

MLP2 の行列乗算では，768×3072 の大規模な重み行列を扱うため，重み行列全体をオンチップメモリに格納すると，大量のメモリリソースを必要とする．そこで，提案アーキテクチャでは，重み行列を列方向に分割し，16 列分の重みデータのみを外部メモリから読み出してオンチップメモリに格納する．読み出した重みデータは複数の入力データに対する演算で再利用されるため，オンチップメモリ使用量を抑制しながら，外部メモリアクセス量を削減できる．

また，前段の Layer Normalization によって生成された入力データはオンチップメモリに格納し，後続の行列乗算で直接利用する．重みデータおよび活性化値は FP16 形式で保持することで，オンチップメモリ使用量を削減する．一方，積和演算の累積処理には FP32 形式を用いることで，演算精度の低下を抑制する．

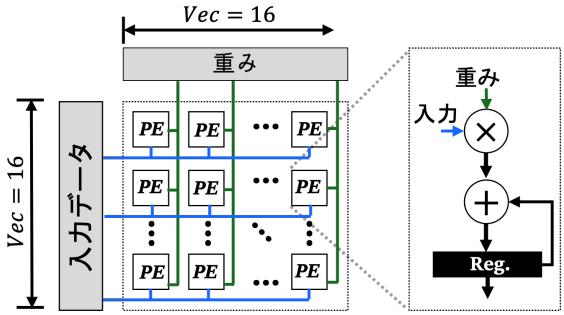

Fig.　4　外積演算に基づく行列乗算用 PE アレイ

FPGA における行列乗算の高速化には，データを隣接 PE 間で転送しながら再利用するシストリックアレイが広く用いられている [8, 9]．しかし，大規模なシストリックアレイでは，PE 間接続および並列演算器へのデータ供給がハードウェア実装上の制約となる．提案アーキテクチャでは，図 4 に示すように，外積演算に基づく 16×16 の PE アレイを用いて行列乗算を実行する．外積演算に基づく行列乗算は，左行列の列ベクトルと右行列の行ベクトルとの外積を逐次計算し，得られた部分結果を累積することで実現される [10]．PE アレイは 256 個の Processing Element（PE）から構成され，各 PE は 16×16 の出力タイルに含まれる一つの要素に対応する．各 PE は，自身に割り当てられた出力要素に対して，独立に積和演算と累積処理を行う..

従来のシストリックアレイでは，隣接する PE 間でデータを転送するための接続構造が必要となる．これに対して，提案する外積演算ベースの PE アレイでは，PE 間で演算データを転送する必要がないため，PE 間の密な接続を削減できる．これにより，配線の複雑さを抑えたコンパクトな構成で，256 PE による高い空間並列性を実現する．

3.　評価

提案アーキテクチャを，Intel Agilex 7 AGF027 FPGA を搭載した BittWare IA-840F ボード上に実装した．FPGA カーネルのコンパイルには，

Table 3　提案アーキテクチャのリソース使用量

リソース	使用量	使用率
ALUT	283,950	16%
FF	464,522	13%
RAM ブロック	3,549	27%
MLAB	5,263	6%
DSP	1,403.5	17%

Intel oneAPI 2024.1 および Quartus Prime Pro 23.1 を用いた．

表 3 に，提案アーキテクチャの実装結果を示す．RAM ブロックは，中間データおよび分割された重みデータを保持するため，最も使用量の大きいリソースとなった．一方，論理リソースおよび DSP の使用量には余裕があり，さらなる並列化の可能性がある．提案アーキテクチャは，497 MHz の動作周波数を達成した．また，256 枚の画像からなるバッチに対する処理時間は 2961 ms であった．以上の結果より，提案アーキテクチャは，高い動作周波数を維持しながら，さらなる並列化の余地を有することが確認された．

4. おわりに

本稿では，MLP-Mixer の FPGA 実装に向けて，メモリアクセスを考慮したアーキテクチャを提案した．提案アーキテクチャでは，実行時間を支配するチャネルミキシング MLP の行列乗算に着目し，大規模な重み行列全体をオンチップメモリに保持するのではなく，16 列単位で読み出した重みデータを再利用する構成を採用した．これにより，オンチップメモリ使用量を抑制するとともに，重みデータに対する外部メモリアクセスの削減を図った．

さらに，外積演算に基づく 256 PE のデータ並列アレイを用いることで，PE 間の密なデータ転送を必要としない行列乗算構造を実現した．これにより，配線の複雑さを抑えながら高い空間並列性を確保できる．Intel Agilex 7 FPGA 上での実装結果より，提案アーキテクチャは 497 MHz の動作周波数を達成し，最大リソース使用率は RAM ブロックの 27%であった．また，256 枚の画像からなるバッチに対して，2961 ms の処理時間を達成した．

参考文献

1) Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

2) Beom Jin Kang, Hae In Lee, Seok Kyu Yoon, Young Chan Kim, Sang Beom Jeong, and Hyun Kim. A survey of fpga and asic designs for transformer inference acceleration and optimization. *Journal of Systems Architecture*, 2024.

3) Hasitha Muthumala Waidyasooriya, Masanori Hariyama, and Daisuke Tanaka. FPGA-Based Deep-Pipelined Architecture for Vision Transformer's Multi-Head Attention. In *25th Workshop on Synthesis And System Integration of Mixed Information Technologies*, pages 160–163, June 2024.

4) Seonghyeok Kim, Hasitha Muthumala Waidyasooriya, Daisuke Tanaka, and Masanori Hariyama. A deeply pipelined fpga accelerator architecture for the mlp-mixer. In *2025 22nd International SoC Design Conference (ISOCC)*, Busan, Korea, Republic of, Oct 2025. IEEE.

5) Chang Sun, Jennifer Ngadiuba, Maurizio Pierini, and Maria Spiropulu. Fast jet tagging with mlp-mixers on fpgas. *Machine Learning: Science and Technology*, 2025.

6) Dhananjay Rao Thallikar Shyam, Shashank Nag, and Lizy K. John. Hmix: An efficient hardware accelerator for quantized mlp-mixer inference. In *Proceedings of the 23rd ACM International Conference on Computing Frontiers (CF '26)*, 2026.

7) Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit,

Mario Lucic, and Alexey Dosovitskiy. MLP-mixer: An all-MLP architecture for vision, 2021.

8) Ahmad Husni Mohd Shapri, Norazeani Abdul Rahman, et al. Optimization and analysis of fpga-based systolic array for matrix multiplication. *AIP Conference Proceedings*, 2024.

9) Zhiyong Liu et al. High-frequency systolic array-based transformer accelerator on fpga. *Electronics*, 12(4), 2023.

10) Subhankar Pal, Jonathan Beaumont, Dong-Hyeon Park, Aporva Amarnath, Siying Feng, Chaitali Chakrabarti, Hun-Seok Kim, David Blaauw, Trevor Mudge, and Ronald Dreslinski. Outerspace: An outer product based sparse matrix multiplication accelerator. In *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2018.