WP486 (v1.0) 2016 年 11 月 11 日

ザイリンクス デバイスでの INT8 に最適化した深層学習の実装

著者 : Yao Fu、Ephrem Wu、Ashish Sirasao、Sedny Attia、Kamran Khan、Ralph Wittig

ザイリンクス デバイスでの INT8 最適化により、深層学習推論の演算手法において最適なパフォーマンスと最高水準の電力効率が実現します。ザイリンクスの統合 DSP アーキテクチャでは、INT8 深層学習演算において、ほかの FPGA DSP アーキテクチャと比較してソリューション レベルで 1.75 倍のパフォーマンスが達成されます。

概要

このホワイト ペーパーの目的は、ザイリンクスの DSP48E2 スライスに実装された INT8 深層学習演算について考察し、ほかの FPGA と比較することです。INT8 を使用したザイリンクスの DSP アーキテクチャは、INT8 深層学習の毎秒演算数 (OPS) において、リソース数が同一のほかの FPGA と比較して、ソリューション レベルで最大 1.75 倍のパフォーマンスを達成できます。深層学習推論では正確さを損なうことなく下位ビットの精度を利用するため、INT8 を効率的に実装する必要があります。

ザイリンクスの DSP アーキテクチャおよびライブラリは、INT8 深層学習推論向けに最適化されています。この資料では、ザイリンクスの UltraScale および UltraScale+ FPGA の DSP48E2 スライスを使用して、同一のカーネル重みを共有した 2 つの INT8 累積乗算 (MACC) 演算を同時に処理する方法について説明します。また、この手法を利用するために入力サイズとして 24 ビットが最小限である理由を論じます。この点がザイリンクス独自の発想です。また、ニューラル ネットワークの基本的な動作を振り返ることにより INT8 最適化手法の適合性を示すための例も用意しました。

INT8 による深層学習

深層ニューラル ネットワークは、機械学習の分野における改革を推進し、既存の多くの応用分野で人間レベルの人口知能の機能を刷新してきました。

深層学習モデルの精度の向上に伴い、その複雑さに対応するため、高い演算能力と広いメモリ帯域幅が必要とされています。演算密度とメモリ帯域幅を抑えつつ精度とスループットを確保できる新たな深層学習推論モデルの開発では、電力効率が革新の推進力となっています。このオーバーヘッドを削減することが、最終的には電力効率の向上と、必要な総消費電力の節減につながります。

演算時の総消費電力の節減に加えて、演算のビット幅が小さい方がメモリ帯域幅に必要な消費電力も低減できます。これは、同量のメモリ トランザクションでより少ないビットが転送されるからです。

深層学習推論においては、同じレベルの精度を確保するために浮動小数点演算は不要であることが研究により判明しています [参照 1] [参照 2] [参照 3]。また、画像分類など多数の応用分野で推論の許容精度を確保するために必要なのは、INT8 以下の固定小数点演算精度に過ぎないことが判明しています [参照 2] [参照 3]。表 1 に、微調整されたネットワークでの、畳み込み層と完全接続層の場合の固定小数点の動的なパラメーターと出力を示します。かっこ内の数値は微調整なしの精度を示します。

表 1: 固定小数点精度での CNN モデル

	層出力	CONV パラメーター	FC パラメーター	32 ビット浮動小数点ベースライン	固定小数点精度
LeNet (Exp1)	4 ビット	4 ビット	4 ビット	99.1 %	99.0 % (98.7 %)
LeNet (Exp2)	4 ビット	2 ビット	2 ビット	99.1 %	98.8 % (98.0 %)
フル CIFAR-10	8 ビット	8 ビット	8 ビット	81.7 %	81.4 % (80.6 %)
SqueezeNet top-1	8 ビット	8 ビット	8 ビット	57.7 %	57.1 % (55.2 %)
CaffeNet top-1	8 ビット	8 ビット	8 ビット	56.9 %	56.0 % (55.8 %)
GoogLeNet top-1	8 ビット	8 ビット	8 ビット	68.9 %	66.6 % (66.1 %)

注記:

1. 出典 : Gysel ほか、『Hardware-oriented Approximation of Convolutional Neural Networks』、ICLR 2016 [参照 2]

ザイリンクス DSP スライスでの INT8 深層学習

ザイリンクスの DSP48E2 は、1 つの積和演算で、1 クロック サイクル内に最大で 18x27 ビットの乗算と最大で 48 ビットの累算を効率的に実行するように設計されています。図 1 を参照してください。その DSP スライス自体にループバックすることにより、または複数の DSP スライスをチェーン接続することにより、ザイリンクス デバイスでは累積乗算 (MACC) を効率的に実行できます。

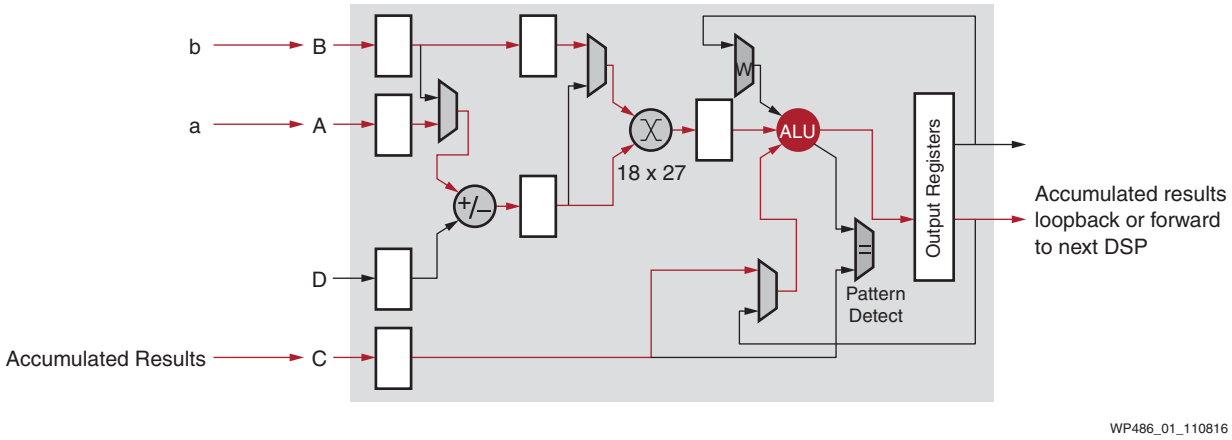

図 1: MACC モードでの DSP スライス

INT8 演算の実行中は、基本的に 27 ビットという広い幅が使用されます。従来の利用法では、(A+B) x C タイプの演算を効率的に実装するために通常は前置加算器が利用されますが、このタイプの演算は深層学習の応用にはあまり見られません。(A+B) x C の結果を A x C と B x C に分けることで、累算を個別のデータ フローで実行できます。これにより、深層学習演算の一般的な要件に適合できます。

18x27 ビット乗算器を備えていることは、INT8 深層学習演算にとってメリットです。1 つの DSP スライスで 2 つの INT8 MACC を同時に実行するには、乗算器への入力の少なくとも 1 つが最低でも 24 ビットで、キャリー アキュムレータが 32 ビットであることが必要です。27 ビットの入力と 48 ビットのアキュムレータを組み合わせることにより、深層学習ソリューションのパフォーマンスが 1.75 倍に向上しました (INT8 深層学習 MACC に対する DSP 乗算器の比が 1.75:1)。ほかのベンダーの FPGA では 1 つの DSP ブロックに 18x19 乗算器があるのみで、INT8 MACC に対する DSP 乗算器の比は 1:1 に限定されています。

スケーラブルな INT8 最適化

目標は、入力 a、b、c の間の乗算結果を a x c と b x c に容易に分けることができるように、a、b、c を効率的にエンコードする方法を見つけることです。

INT8 乗算などの減精度演算では、より精度の高い 10 ビットまたは 19 ビットの入力は 0 または 1 で埋められており、運んでいる情報は 1 ビットのみです。これは、45 ビットの最終の積の上位 29 ビットでも同じです。そのため、下位の 8 ビットおよび 16 ビットの入力結果に影響を与えずに上位 19 ビットを使用して別の演算を実行することが可能です。

未使用の上位ビットを別の演算に利用する際は、一般的に次の 2 つのルールに従う必要があります。

1. 上位ビットが下位ビットの演算に影響を与えてはならない。
2. 下位ビットの演算により上位ビットに影響が生じた場合の検出および回復が可能でなければならない。

上記のルールを満たすため、上位の積の結果の最下位ビットが下位 16 ビットに入らないようにする必要があります。つまり、上位ビットの入力は少なくとも第 17 ビットから始める必要があります。上位が 8 ビットの入力の場合、合計入力サイズは最小で 16 + 8 = 24 ビットが必要です。この 24 ビットの最小入力サイズで保証できるのは、1 つの乗算器での 2 つの同時乗算のみです。これでは、全体で 1.75 倍の MACC スループットを達成するのに十分ではありません。

1 つの DSP48E2 スライスで ac と bc を並列に計算する手順を次に示します。ここではスライスが 27 ビット前置加算器 (入力も出力も 27 ビット幅) と 27x18 乗算器を持つ演算ユニットとして使用されています。詳細は、図 2 を参照してください。

1. 8 ビットの入力 a と b は、前置加算器を通して DSP48E2 乗算器の 27 ビット ポート p にパックされます。これにより、2 ビットのベクターはできるだけ遠ざけられます。
   入力 a は 18 ビットだけ左シフトされます。これは、b < 0 かつ a = –128 である場合に前置加算器でのオーバーフローを防ぐために、最初の項から 27 ビット内に 2 つの符号ビット a を生成するためです。a のシフト量が 18 であること、つまり DSP48E2 乗算器ポート B の幅は偶然です。

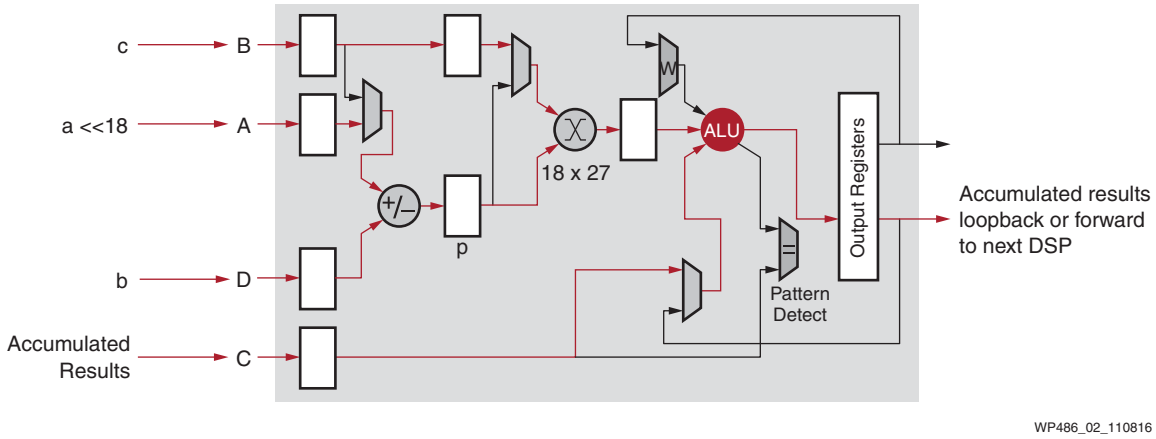

図 2: 8 ビット最適化

2. パックされた 27 ビット ポート p と 18 ビットの c で表される 8 ビット係数の積を 2 の補数フォーマットで計算するために、DSP48E2 27x18 乗算器が使用されます。これで、この 45 ビットの積は、2 つの 44 ビット項の和を 2 の補数フォーマットで表したものになります。つまり、18 ビットだけ左シフトされた ac と bc です。

上述の 45 ビットの積を累算するために後置加算器を使用できます。この積には、上位と下位に分割可能な積項が含まれています。上位項と下位項に対して正しい累算が実行され、単一の 45 ビットの積が累算されます。最終的な累算結果は、オーバーフローが生じていなければ、単純な演算で分割できます。

この手法の限界は、各 DSP スライスで累算できる積項の数にあります。上位と下位の積項の間には 2 ビットが残っているため (図 3 参照)、下位ビットにオーバーフローを生じることなく累算を保証できる積項の数は、最大で 7 つまでです。積項の数が 7 つを超えた場合にこの限界を広げるには、追加の DSP スライスが必要です。結果として、ここで 8 つの DSP スライスは 7x2 INT8 乗算/加算演算を実行します。これは、同じ数の乗算器を持つ競合デバイスと比較して 1.75 倍の INT8 深層学習演算です。

この手法には、実際のユース ケースの要件に応じて幅広く応用可能です。ReLU (正規化線形関数) を使用した畳み込みニューラル ネットワーク (CNN) では、非負のアクティベーションが生成され、符号なしの INT8 フォーマットでは 1 ビットだけ精度が増して、ピーク スループットが 1.75 倍に向上します。

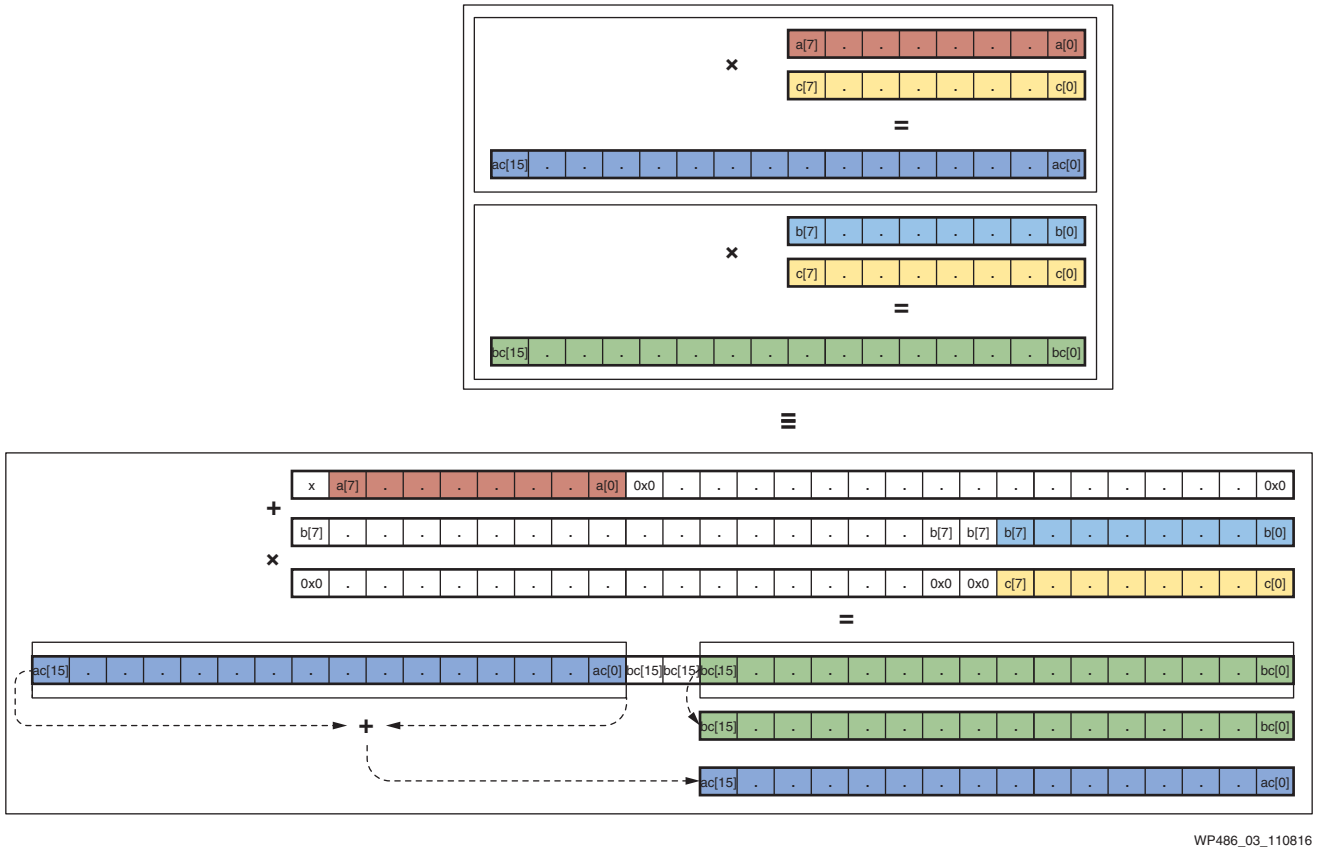

図 3: 1 つの DSP48E2 スライスでの 2 つの INT8 乗算のパック

CNN の演算要件

最新のニューラル ネットワークの多くは、元来のパーセプトロン モデルから派生しています [参照 4]。詳細は、図 4 を参照してください。

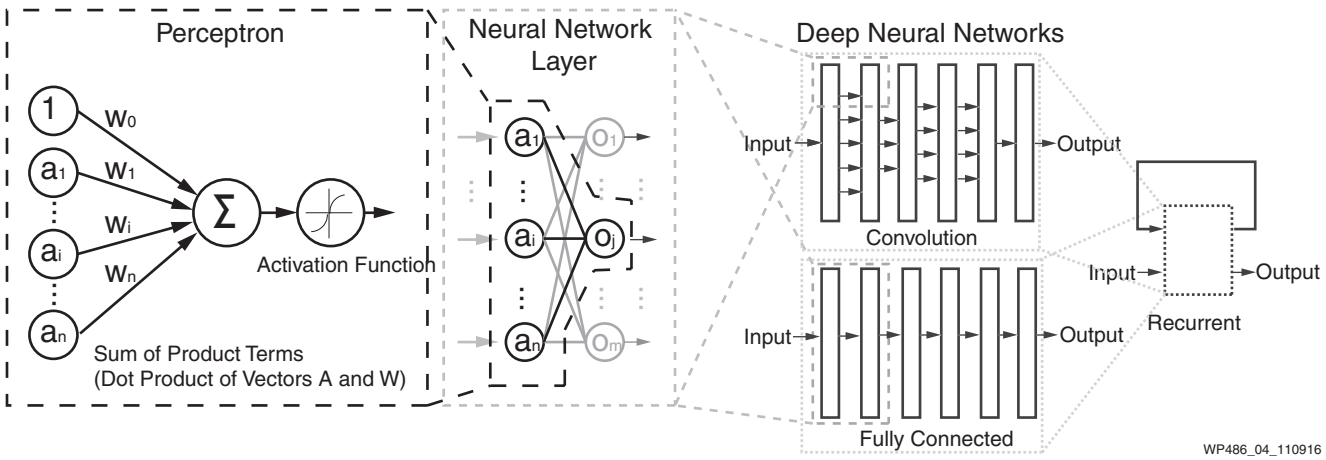

図 4: パーセプトロンと深層ニューラル ネットワーク

深層ニューラル ネットワーク (DNN) とも呼ばれる最新の深層学習の基本的な演算は、標準的なパーセプトロン構造から大きく進歩したとは言え、未だにパーセプトロン的な演算を継承しています。ただし、パーセプトロン構造は全体としてより広く、また、より深く積み重なっています。図 4 ではパーセプトロンの基本的な演算も示しています。この演算は複数の層を介して典型的な深層学習推論ごとに究極的には数百万回から数十億回繰り返されます。図 5 に示すように、ニューラル ネットワークのある層における m 個のパーセプトロン/ニューロン出力

$$o_j \ (j \in [1, m])$$

のそれぞれを計算するための主要な操作は、n 個の入力サンプル

$$a_i (i \in [1, n])$$

を取り込み、各入力を対応する次のカーネル重みで乗じて、

$$w_{i,j} \ (i \in [1, n], j \ \in [1, m])$$

その結果を累算します。

$$o_j = f(\sum_i a_i \, w_{i,j}), \qquad (i \in [1, n])$$

ここで、f(x) は任意のアクティベーション関数です。

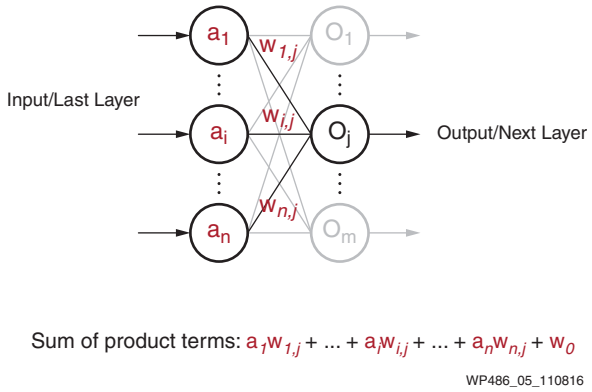

図 5: 深層学習におけるパーセプトロン

a_i と $w_{i,j}$ の精度が INT8 に制限される場合、この積和は、INT8 最適化手法で記述した並列の MACC の最初のものになります。

2 番目の積和では同じ入力 $a_i \ (i \in [1, n])$ を使用しますが、別の一連のカーネル重み $w_{i,k}$ $(i \in [1, n], k \ \in [1, m], and \ k \ \neq j)$ を使用します。

2 番目のパーセプトロン/ニューロン出力の結果は次のようになります。

$$o_k = f(\sum_i a_i \, w_{i,k}), \qquad (i \in [1, n], k \ \neq j)$$

詳細は、図 6 を参照してください。

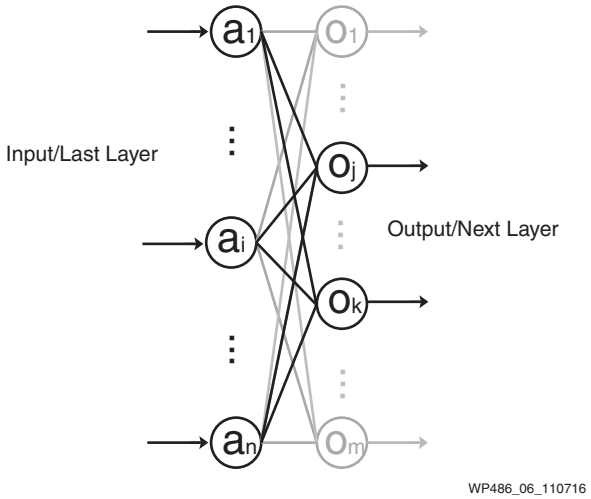

図 6: 共通の入力を使用した並列の 2 つの積和項

INT8 最適化手法を使用して $w_{i,k}$ の値を 18 ビット左シフトすることで、各 DSP スライスでは最終出力値の一部である独立した部分が生成されます。各 DSP スライスのアキュムレータのビット幅は 48 ビットであり、次のスライスにチェーン接続されます。これにより、チェーン接続したブロックの数は 7 に制限されます。これを超えると、シフトされた $w_{i,k}$ が飽和して演算に影響します。つまり、合計 n 個の入力サンプルに対して n 個の DSP スライスで MACC は 2n 個になります。

標準的な DNN の各層には数百から数千の入力サンプルがあります。ただし、7 項を累算した後は、48 ビットのアキュムレータの下位の項が飽和する可能性があるため、7 項の和ごとに DSP48E2 スライスが追加で必要になります。これは、14 の MACC が、7 つの DSP スライスと、過飽和を防ぐためのもう 1 つの DSP スライスで得られることを意味します。結果として、スループットが 7/4、つまり 1.75 倍に向上します。

畳み込みニューラル ネットワーク (CNN) では通常、畳み込み層で同一の重みが頻繁に再利用されて、a x w および b x w というタイプの並列 MACC 演算を形成します。したがって、入力の共有に代えて重みの共有も利用できます (図 7 参照)。

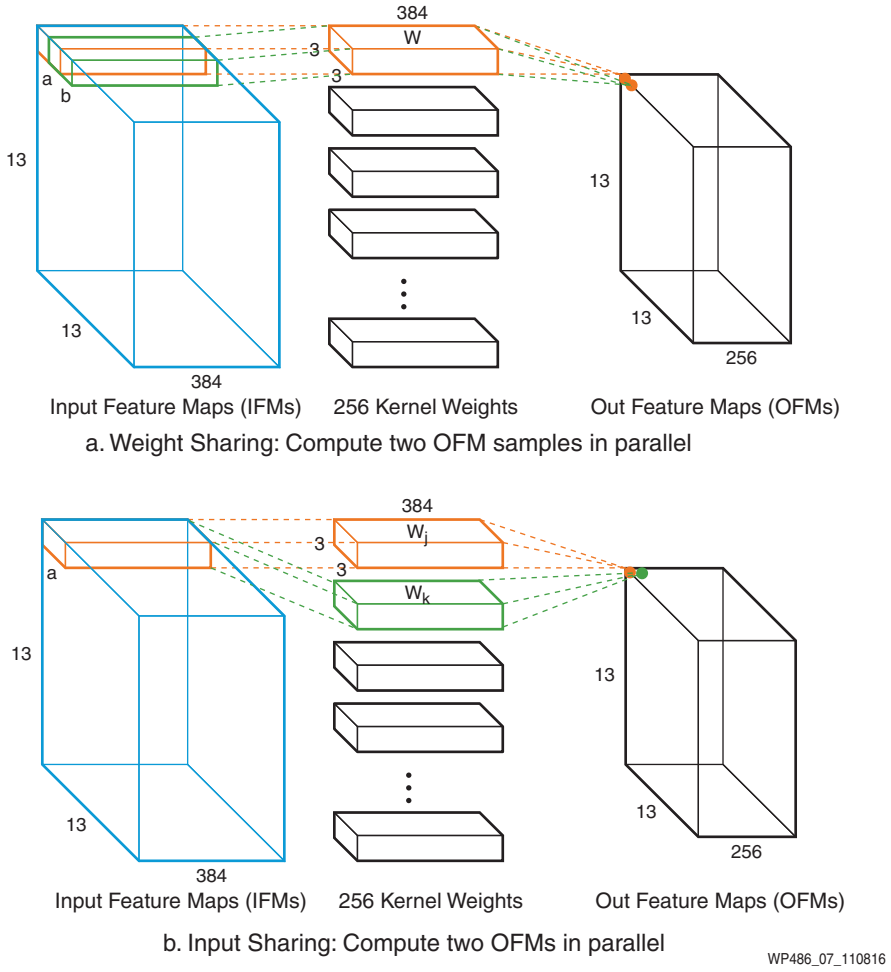

図 7: 重みの共有と入力の共有の比較

INT8 でチェーン接続した MACC を作成するその他の方法

INT8 MACC は、FPGA ファブリック内で DSP スライス同様の頻度で LUT を使用して構成することもできます。これは、FPGA の使い方によっては、深層学習パフォーマンスの大幅な向上につながり、3 倍に向上することもあります。FPGA 以外のほかのアーキテクチャでは多くの場合、使用可能な深層学習演算の実行時に、こうした利用可能な演算リソースが考慮の対象になりません。

ザイリンクスの FPGA のプログラム可能なファブリックは、さまざまなワークロードを同時かつ効率的に処理できるという点で独特です。たとえば、ザイリンクスの FPGA では、CNN 画像の分類、ネットワークでの暗号化、データの圧縮を同時に実行できます。深層学習パフォーマンスの競合分析において、MACC LUT は考慮に入れていません。これは LUT が通常、MACC 機能の実行よりも、その他の並行機能の実行に使用した方が有用だからです。

競合分析

この競合分析では、Intel 社の (かつては Altera 社の) Arria 10 デバイスと近日発売される Stratix 10 デバイスを、ザイリンクスの Kintex® UltraScale™ および Virtex® UltraScale+™ ファミリと比較しています。この演算力重視の比較では、各製品ファミリで DSP の密度が最も高い、Arria 10 (AT115)、Stratix 10 (SX280)、Kintex UltraScale (KU115)、Virtex UltraScale+ (VU9P)、および Virtex UltraScale+ (VU13P) の各デバイスを選択しました。比較では、深層学習など数多くの応用分野で使用できる汎用 MACC のパフォーマンスに焦点を当てています。

Intel 社の MACC パフォーマンスは、前置加算器を活用した演算子に基づいています。ただし、この実装で生成されるのは、積項の和であり、個々の独立した積項ではありません。したがって、Intel 社の前置加算器は深層学習演算には適していません。

Intel 社デバイスの消費電力見積もりでは、次のワースト ケースを想定して、Intel 社の EPE 消費電力解析ツールを使用しています。

1. F_{MAX} での DSP 使用率: 90 %
2. クロック レート DSP F_{MAX} でのロジック使用率: 50 %
3. 半分のクロック レート DSP F_{MAX} でのブロック RAM 使用率: 90 %
4. DDR4 が 4 つ、PCIe Gen3 x8 が 1 つ
5. DSP トグル レート: 12.5 %
6. T_J: 80°

図 8 は、深層学習演算での電力効率を比較したものです。INT8 最適化では、ザイリンクスの UltraScale および UltraScale+ のデバイスは、INT16 演算 (KU115 INT16 から KU115 INT8) と比較して、INT8 精度で 1.75 倍の電力効率を達成できます。また、ザイリンクス デバイスは、深層学習推論演算において、Intel 社の Arria 10 デバイスおよび Stratix 10 デバイスと比較して 2 倍から 6 倍優れた電力効率を実現します。

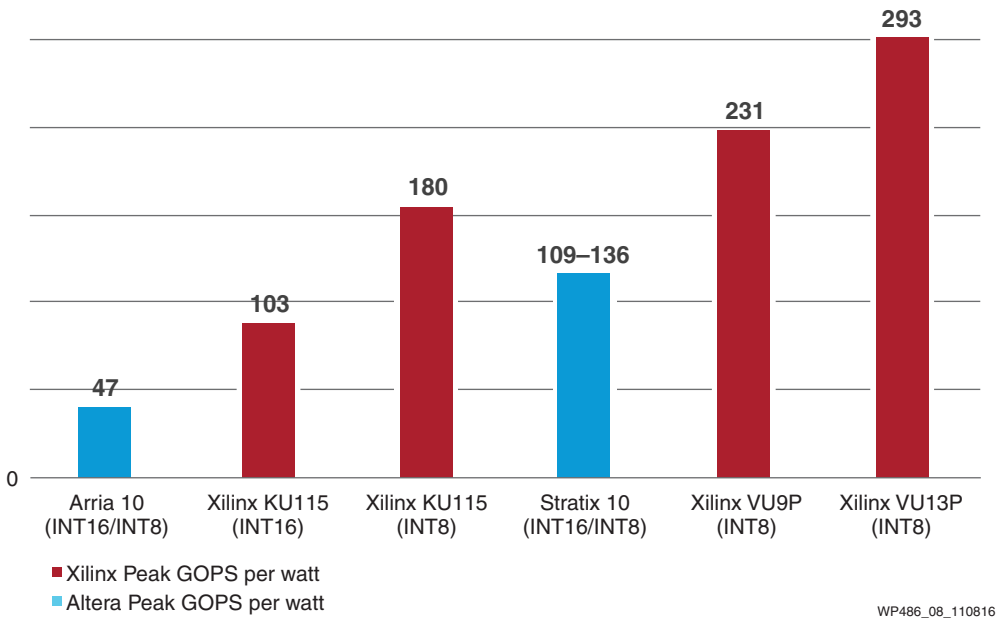

図 8: INT8 深層学習での電力効率の比較: ザイリンクスと Intel 社

まとめ

このホワイト ペーパーでは、ザイリンクスの DSP48E2 スライスが INT8 深層学習演算に最適であり、1.75 倍のパフォーマンスを達成できることについて論じました。ザイリンクスの DSP48E2 スライスを使用すると、同一のカーネル重みを共有して複数の INT8 MACC 演算を同時に実行できます。INT8 を効率良く実装するには、24 ビットの入力幅が必要です。この利点をサポートしているのは、ザイリンクスの UltraScale および UltraScale+ FPGA の DSP スライスのみです。ザイリンクス デバイスは、画像分類など、深層学習を応用した INT8 ワークロードに非常に適しています。ザイリンクスは今後も、深層学習の応用を促進するためにハードウェアおよびソフトウェアに基づく新たなメソドロジを開拓していきます。

データ センターにおける深層学習の詳細は、次のウェブサイトを参照してください。

https://japan.xilinx.com/accelerationstack

参考資料

1. Dettmers、『8-Bit Approximations for Parallelism in Deep Learning』、ICLR 2016 https://arxiv.org/pdf/1511.04561.pdf
2. Gysel ほか、『Hardware-oriented Approximation of Convolutional Neural Networks』、ICLR 2016 https://arxiv.org/pdf/1604.03168v3.pdf
3. Han ほか、『Deep Compression: Compressing Deep Neural Networks With Pruning, Trained Quantization And Huffman Coding』、ICLR 2016 https://arxiv.org/pdf/1510.00149v5.pdf
4. F. Rosenblatt、『The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain』、Psychological Review 第 65 巻、No. 6、1958 http://www.ling.upenn.edu/courses/cogs501/Rosenblatt1958.pdf

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	内容
2016 年 11 月 11 日	1.0	初版

免責事項

自動車用のアプリケーションの免責条項

この資料に関するフィードバックおよびリンクなどの問題につきましては、jpn_trans_feedback@xilinx.com まで、または各ページの右下にある [フィードバック送信] ボタンをクリックすると表示されるフォームからお知らせください。いただきましたご意見を参考に早急に対応させていただきます。なお、このメール アドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。