



WP490 (v1.0) 2017 年 3 月 10 日

# 利用赛灵思器件上的 INT8 优化开发嵌入式视觉

作者: Yao Fu、Ephrem Wu、Varun Santhaseelan、Kristof Denolf、Kamran Khan 和 Vinod Kathail

赛灵思 INT8 优化为使用深度学习推断和传统计算机视觉功能的嵌入式视觉应用提供最优异的性能和能效最出色的计算方法。与其他 FPGA DSP 架构相比，赛灵思的集成 DSP 架构在 INT8 深度学习运算上能够实现 1.75 倍的解决方案级性能。

## 概要

本白皮书探讨将 INT8 运算用于实现在赛灵思 DSP48E2 片上、使用深度学习推断和计算机视觉功能的嵌入式视觉应用，以及这种方案与其他 FPGA 的对比。与占用相同资源数量的其他 FPGA 相比，赛灵思的 DSP 架构对 INT8 乘法累加 (MACC) 运算能够实现 1.75 倍的峰值解决方案级性能。由于嵌入式视觉应用可以在不牺牲准确性的情况下使用较低位精度，因此需要高效的 INT8 实现方案。

赛灵思的 DSP 架构和库针对 INT8 运算进行了精心优化。本白皮书介绍如何使用赛灵思 16nm 和 20nm All Programmable 器件中的 DSP48E2 Slice，在共享相同内核权重的同时处理两个并行的 INT8 MACC 运算。本白皮书还阐述了要运用赛灵思这一独特技术，为何输入的最小位宽为 24 位。此外本白皮书还详细介绍了如何以 SIMD 模式使用 DSP48E2 Slice，供基本算术运算使用。另外还提供在深度学习领域或其他计算机视觉处理任务领域如何将这此功能用于嵌入式视觉的实例。

## 用于深度学习和计算机视觉的 INT8

嵌入式视觉是专业术语，指的是在嵌入式平台上实现用于现实用途的计算机视觉算法。虽然计算机视觉算法近年来有明显改进，要在降低功耗的条件下将这样复杂且高计算强度的算法移植到嵌入式平台上却是一大挑战。以更低功耗处理更多运算是一个亘古不变的需求，无论是对过滤、角点检测等传统计算机视觉算法还是对深度学习算法。

深度神经网络凭借人类级 AI 功能已推动众多应用不断演进并重新对其定义。鉴于这类算法提供的超高精度，这些网络是嵌入式设备中的主要工作负载。随着更精确的深度学习模型被开发出来，它们的复杂性也带来了高计算强度和高内存带宽方面的难题。能效要求正推动深度学习推断新模式开发方面的创新，这些模式需要的计算强度和内存带宽较低，但绝不能以牺牲准确性和吞吐量为代价。降低这一开销将最终提升能效，降低所需的总功耗。

除了节省计算过程中的耗电，较低位宽的计算还能降低内存带宽所需的功耗，因为在内存事务数量不变的情况下传输的位数减少了。

研究显示要保持同样的准确性 [参考资料 1][参考资料 2][参考资料 3]，深度学习推断中无需浮点计算，而且图像分类等许多应用只需要 INT8 或更低定点计算精度来保持可接受的推断准确性 [参考资料 2][参考资料 3]。表 1 列出了精调网络以及卷积层和完全相连层的动态定点参数及输出。括号内的数字代表未精调的准确性。

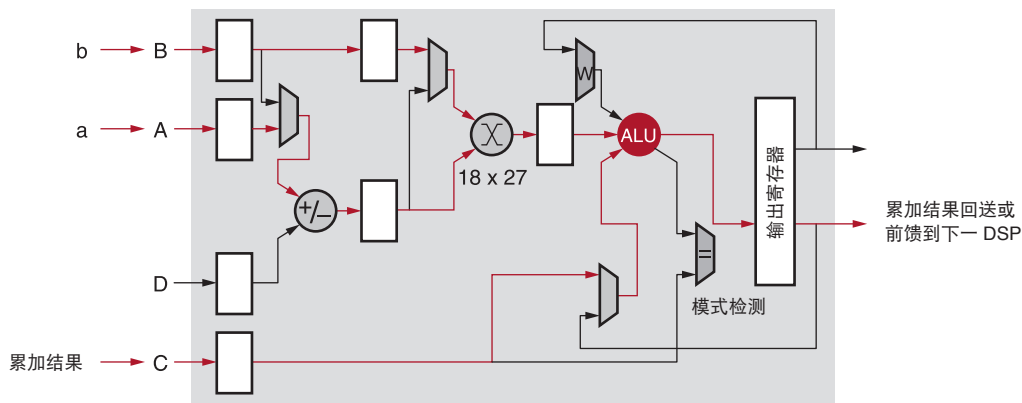
表 1：带定点精度的 CNN 模型

	层输出	卷积参数	完全相连 (FC) 参数	32 位浮点基线	定点精度
LeNet (示例 1)	4 位	4 位	4 位	99.1%	99.0% (98.7%)
LeNet (示例 2)	4 位	2 位	2 位	99.1%	98.8% (98.0%)
Full CIFAR-10	8 位	8 位	8 位	81.7%	81.4% (80.6%)
SqueezeNet top-1	8 位	8 位	8 位	57.7%	57.1% (55.2%)
CaffeNet top-1	8 位	8 位	8 位	56.9%	56.0% (55.8%)
GoogLeNet top-1	8 位	8 位	8 位	68.9%	66.6% (66.1%)

对用于深度学习的 INT8 运算的优化也直接地适用于大量传统计算机视觉功能。这些算法一般工作在 8 位到 16 位整数表达式。OpenVX [参考资料 4] 是一种近期提议的计算机视觉标准，规定了每个通道 INT8 表达式的用法。大多数计算机视觉应用需要某些程度的过滤，而过滤能够分解为一套点积运算。赛灵思 DSP48E2 Slice 上的 SIMD 运算模式为实现视觉算法涉及的运算提供额外选择。

## 赛灵思 DSP Slice 片上的 INT8 运算

UltraScale 和 UltraScale+ FPGA、Zynq UltraScale+ MPSoC（可编程逻辑）中的赛灵思 DSP48E2 Slice 设计用于完成一次乘法和加法运算，最大可在一个时钟周期内高效地实现 18x27 位相乘和 48 位累加，如图 1 所示。除了采用回送或链接多个 DSP48E2 Slice，乘法累加 (MACC) 也能使用赛灵思器件高效完成。



WP490\_01\_03

图 1：使用 MACC 模式的 DSP48E2 Slice

INT8 计算本身就能发挥宽 27 位带宽的优势。在传统应用中，预加法器一般用于高效实现  $(A+B) \times C$  类型的运算，但这类型运算不常见于深度学习和计算机视觉应用中。将  $(A+B) \times C$  的结果分解为  $A \times C$  和  $B \times C$ ，然后在独立的数据流中进行累加，使之适用于典型深度学习和计算机视觉计算的要求。

对 INT8 MACC 运算来说，拥有 18x27 位乘法器很占优势。乘法器的输入中至少有一个必须为最小 24 位，同时进位累加器必须为 32 位宽，才能在一个 DSP48E2 Slice 上同时进行两个 INT8 MACC 运算。27 位输入可与 48 位累加器相结合，实现 1.75 倍的解决方案性能提升（DSP 乘法器与 INT8 MACC 之比为 1.75:1）。其他厂商提供的 FPGA 在单个 DSP 模块中只提供 18x19 乘法器，DSP 乘法器与 INT8 MACC 之比仅为 1:1。

### 可扩展的 INT8 优化

目标是找到一种能够对输入  $a$ 、 $b$  和  $c$  进行高效编码的方法，这样  $a$ 、 $b$  和  $c$  之间的相乘结果可以轻松分解为  $a \times c$  和  $b \times c$ 。鉴于公共输入  $c$ ，这种方法可以推断为单个指令，拥有公共系数的 2 个数据。

在更低精度计算中，例如 INT8 乘法中，高位 10 位或 19 位输入用 0 或 1 填充，仅携带 1 位信息。对最终的 45 位乘积的高位 29 位来说，情况一样。这样就可以使用高位 19 位来进行另一个运算，同时不影响低位 8 位和 16 位输入。

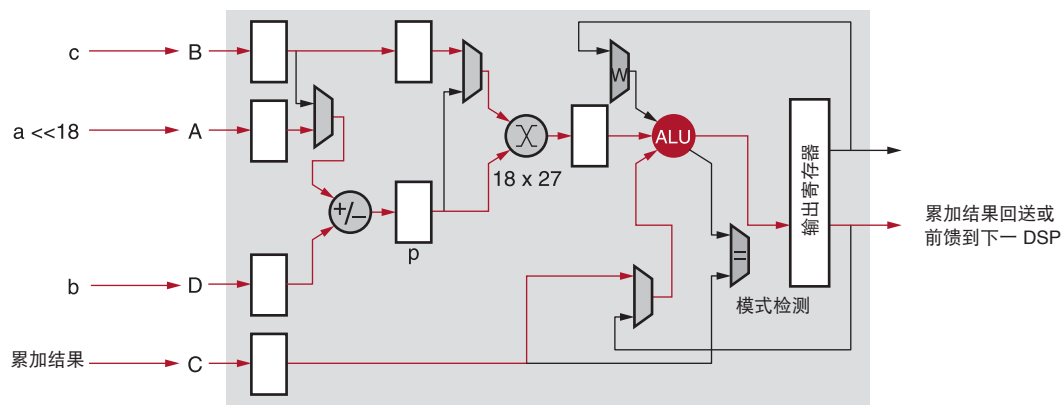
总的来说，要把未使用的高位用于另一计算必须遵循两条规则：

1. 高位不应影响低位的计算。
2. 低位计算对高位的任何影响必须可检测、可恢复。

为满足上述规则，高位乘积结果的最低有效位不得进入低位 16 位。因此高位的输入应至少从第 17 位开始。对一个 8 位的高位输入，总输入宽位至少为  $16+8=24$  位。这种最小 24 位输入大小只能保证用一个乘法器同时完成两次相乘，但仍然足以实现总体 1.75 倍的 MACC 吞吐量。

接下来的步骤是在一个 DSP48E2 Slice 中并行计算  $ac$  和  $bc$ 。DSP48E2 Slice 被用作一个带有一个 27 位预加法器（输入和输出均为 27 位宽）和一个  $27 \times 18$  乘法器的算术单元。见图 2。

1. 通过预加法器在 DSP48E2 乘法器的 27 位端口  $p$  打包 8 位输入  $a$  和  $b$ ，这样 2 位向量能尽量分隔开。输入  $a$  左移位仅 18 位，这样从第一项得到的 27 位结果中的两个符号位  $a$  以避免在  $b < 0$  和  $a = -128$  时预加法器中发生溢值。 $a$  的移位量为 18，恰好与 DSP48E2 乘法器端口  $B$  的宽度一样。



WP490\_02\_020317

图 2 : 8 位优化

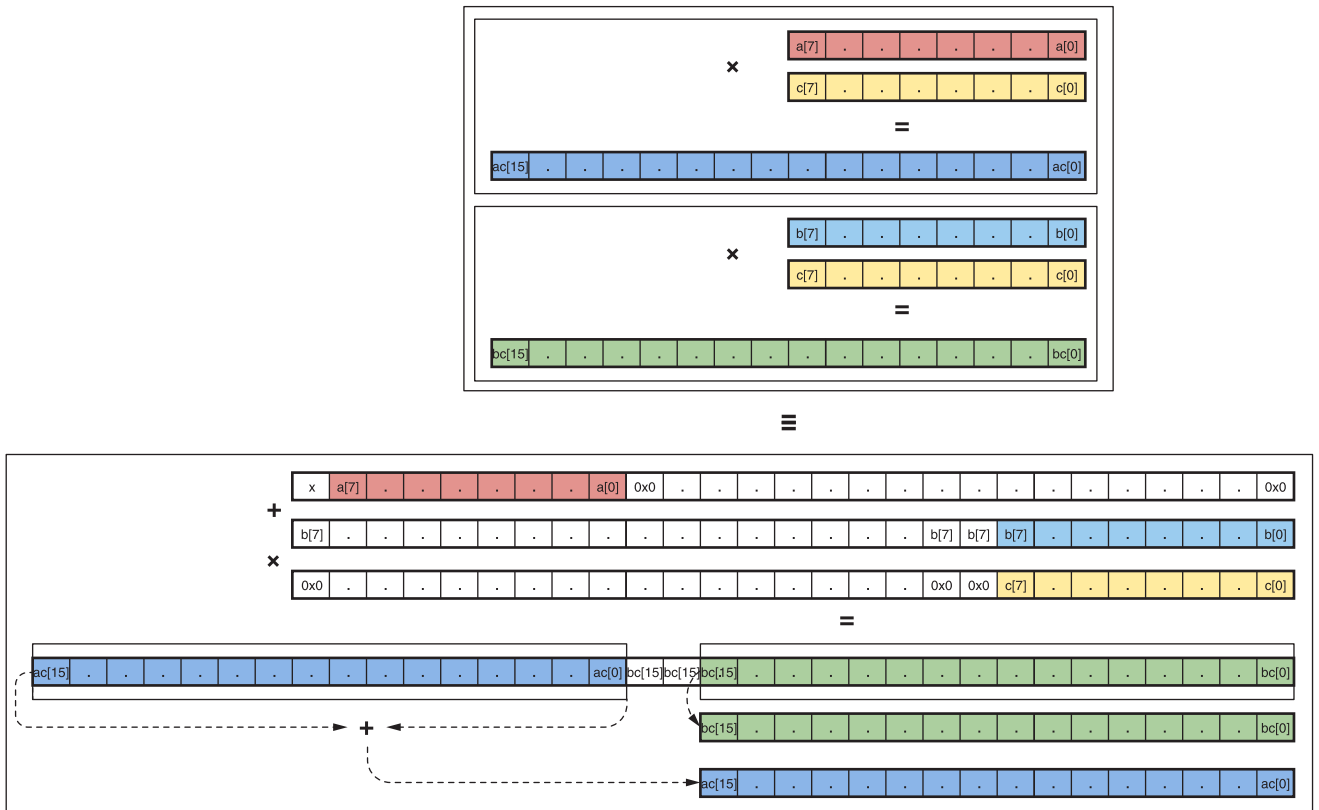
2. DSP48E2  $27 \times 18$  乘法器用于计算打包的 27 位端口  $p$  和以二进制补码格式表达在 18 位  $c$  中的 8 位系数的积。现在该 45 位乘积是二进制补码格式的两个 44 位项的和： $ac$  左移位 18 位以及  $bc$ 。

后加法器可用于累加上述包含单独的高位乘积项和低位乘积项的 45 位乘积。在累加单个 45 位积时，对高位项和低位项进行了校正累加。最终的累加结果如果没有溢值，可以用简单运算分开。

这种方法的局限在于每个 DSP48E2 Slice 能累加的乘积项的数量。由于高位项和低位项间始终保持两位（图 3），可以保证在低位不溢值的情况下累加多达 7 个项。在七个乘积项后，需要使用额外的 DSP48E2

Slice 来克服这一局限。执行 7x2 INT8 乘法 - 加法运算，与拥有相同数量乘法器的竞争型器件相比 INT8 MACC 运算的效率提升 1.75 倍。

根据实际用例的要求，这种方法有多种变化形式。使用修正线性单元（ReLU）的卷积神经网络产生非负激活，而无符号 INT8 格式产生额外一位精度和 1.78 倍峰值吞吐量提升。

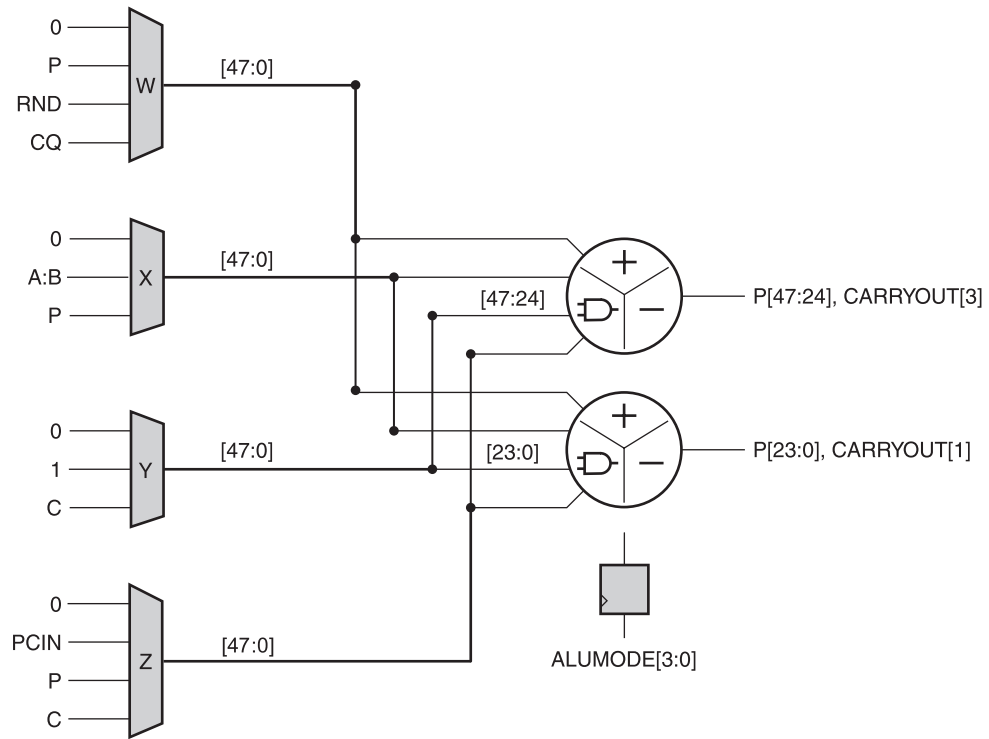


WP490\_03\_020317

图 3：用单个 DSP48E2 Slice 打包两个 INT8 乘运算

## DSP48E2 SIMD 模式

DSP48E2 Slice 的后加法器分裂成四个 12 位或两个 24 位 SIMA ALU（参见图 4），以执行并行加法、减法、累加或逐位逻辑运算。在 SIMD 模式下，DSP48E2 Slice 的预加法器和乘法器不可用。在每个周期上，ALUMODE[3:0] 控制总线选择运算，而 OPMODE[8:0] 控制总线则选择操作数 W、X、Y 和 Z。如果考虑 24 位运算，DSP48E2 Slice 的 P 寄存器能存储处理两个输入阵列的结果。对每一个阵列按顺序求和，每个周期一个元。吞吐量由此变为每个周期产生两个新结果。详细说明，请参阅《UltraScale 架构 DSP Slice 用户指南》(UG579)（关键字“SIMD”、“ALUMODE”和“OPMODE”）[参考资料 5]。

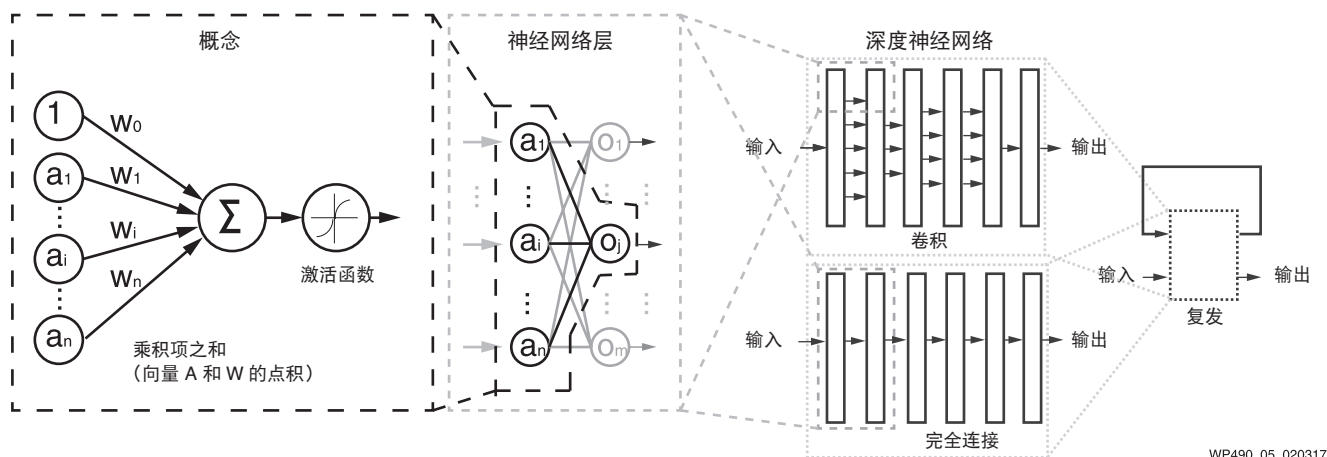


WP490\_04\_020917

图 4 : DSP48E2 双 24 位 SIMD 模式

## 映射 INT8 优化到深度学习应用

新型神经网络大部分是从这个初始概念模型 [参考资料 6] 衍生而来。见图 5。



WP490\_05\_020317

图 5 : 概念和深度神经网络

虽然从标准感知器结构开始已有相当程度的演进，现代深度学习（也称为深度神经网络 (DNN)）的基本

运算仍然是类感知器的运算，只是有更大的总体规模和更深的堆叠感知器结构。图 5 显示了感知器的基本运算。在每个典型的深度学习推断中它穿过多个层，最终重复数百万至数十亿次。如图 6 所示，在神经网络层中，计算  $m$  个感知器 / 神经元输出中的每一个输出的主要计算运算包括：

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

将全部的  $n$  个输入样本

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

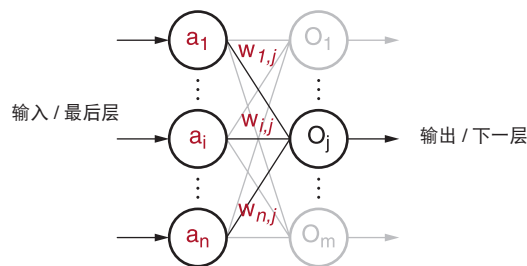
乘以对应的内核权重

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

并累加结果

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

其中： $f(x)$  可以是任何选择的激活函数。



乘积项之和： $a_1 w_{1,j} + \dots + a_i w_{i,j} + \dots + a_n w_{n,j} + w_0$

WP490\_06\_020317

图 6：深度学习中的感知器

如果  $a_i$  和  $w_{i,j}$  的精度限定为 INT8，该乘积之和是 INT8 优化方法中介绍的并行 MACC 中的第一个。

第二个乘积和使用相同输入  $a_i \ (i \in [1, n])$ ，但内核权重  $w_{i,k} \ (i \in [1, n], k \in [1, m], \text{且 } k \neq j)$  不同。

第二个感知器 / 神经元输出的结果是

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

见图 7。



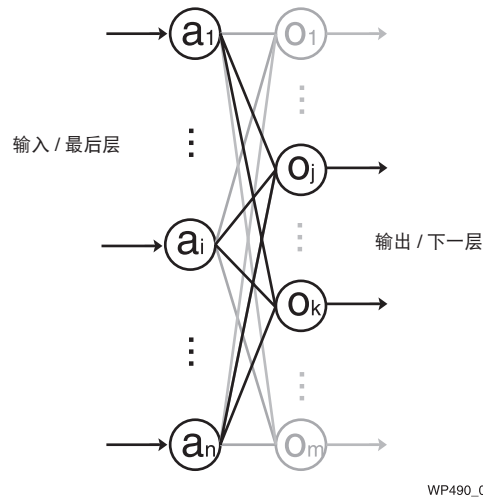


图 7：使用共享输入并行得到两个乘积项和

使用 INT8 优化方法将  $w_{i,k}$  值向左移位 18 位，每个 DSP48E2 Slice 就得出最终输出值的部分且独立的一部分。用于每个 DSP48E2 Slice 的累加器有 48 位宽并链接到下一个 Slice。为避免移位  $w_{i,k}$  饱和影响到计算，链接的模块数量被限制为 7 个，即对总共  $n$  个输入样本使用  $2n$  个 MACC 和  $n$  个 DSP Slice。

典型的 DNN 每层有数百到数千个输入样本。但是在完成 7 个项的累加后，48 位累加器的低位项可能饱和，因此每 7 个项之和就需要一个额外的 DSP48E2 Slice。这相当于每 7 个 DSP48E2 Slice 和 14 个 MACC，另加一个 DSP48E2 Slice 用于防止过饱和，从而带来 7/4 或 1.75 倍的吞吐量提升。

在卷积神经网络（CNN）中，卷积层一般主要使用同一组权重，从而形成  $a \times w$  和  $b \times w$  类型的并行 MACC 运算。因此除输入共享外，还可以使用权重共享（见图 8）。



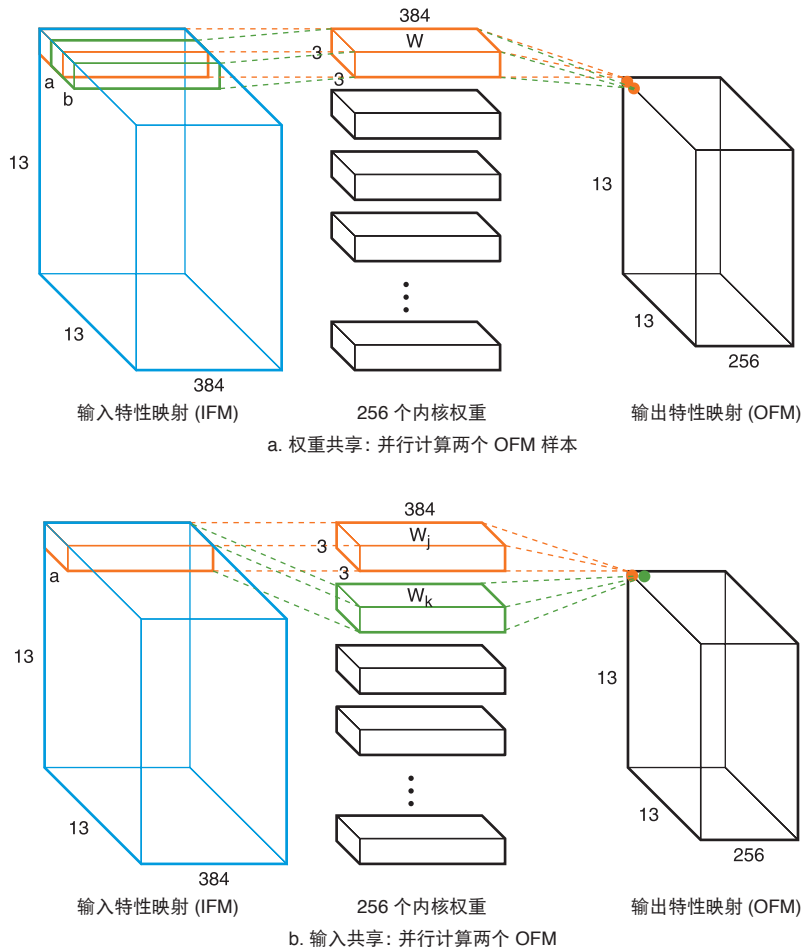


图 8 : 权重共享和输入共享比较

## 创建 INT8 链接 MACC 的其他方法

还可以使用可编程逻辑中与 DSP48E2 Slice 工作频率近似的可用 LUT（即未被设计其余部分使用的 LUT）来构建 INT8 链接 MACC。

使用可用 LUT 能显著提高深度学习性能，一些情况下可提升达 3 倍之多。在许多情况下，对于其他非 FPGA 架构而言，在计算可用深度学习运算时这些可用的计算资源并未考虑在内。

赛灵思 FPGA 和 MPSoC 中的编程逻辑是独有的，因为它能并行且高效地处理不同工作负载。例如赛灵思 FPGA 和 MPSoC 能并行执行 CNN 图像分类、网络加密和数据压缩。本深度学习性能比较分析未将 MACC LUT 考虑在内，因为一般 LUT 用于执行其他并行功能比用于执行 MACC 功能时更有价值。

# 映射 INT8 优化到计算机视觉功能

Khronos OpenVX 标准定义了一套计算机视觉处理模块，对下列用例尤为重要：面部、身体和手势跟踪；智能视频监控；高级驾驶员辅助系统 (ADAS)；对象和情景再现；增强现实；目测；机器人等。表 2 显示了 INT8 优化适用的计算机视觉相关功能。

表 2：适用于计算机视觉功能的 INT8 优化

图像 / 视觉处理 基准组件	OpenVX	OpenCV	可扩展的 INT8 可兼容	SIMD 可兼容
<b>像素级空间过滤和几何空间转换</b>				
绝对差	AbsDiff	absDiff	-	Y
累加图像	AccumulateImage	accumulate	-	Y
累加加权	AccumulateWeightedImage	accumulateWeighted	Y	-
算术相加	Add	add	-	Y
算术相减	Substract	subtract	-	Y
逐位与、或、 异或、非	And, Or, Xor, Not	bitwise_and, bitwise_or, bitwise_xor, bitwise_not	-	Y
阈值	Threshold	threshold	-	Y
箱式滤波器	Box3x3	boxFilter	Y	-
滤波器 2D (卷积)	Convolve	filter2D	Y	-
最大膨胀滤波器 (形态)	Dilate3x3	dilate	-	Y
最小腐蚀滤波器 (形态)	Erode3x3	erode	-	Y
高斯滤波器	Gaussian3x3	GaussianBlur	Y	-
高斯金字塔	GaussianPyramid	buildPyramid	Y	-
拉普拉斯金字塔	LaplacianPyramid		Y	-
中值	Median3x3	medianBlur		Y
索贝尔	Sobel3x3	Sobel	Y	-
缩放	HalfScaleGaussian, ScaleImage	resize	Y	-
光流	OpticalFlowPyrLK	PyrLKOpticalFlow	Y	-
立体视差	不可用	FindStereoCorrespondence BM	Y	-
<b>分析与识别</b>				
积分图像	IntegralImage	integral	-	Y
平均标准偏差	MeanStdDev	meanStddev	-	Y
最小 最大 位置	MinMaxLoc	minMaxLoc	-	Y
Canny 边缘检测	CannyEdgeDetector	Canny	Y	Y

表 2：适用于计算机视觉功能的 INT8 优化（续）

图像 / 视觉处理 基准组件	OpenVX	OpenCV	可扩展的 INT8 可兼容	SIMD 可兼容
FAST9 角点检测	FastCorners	FAST	Y	-
哈里斯角点检测	HarrisCorners	cornerHarris	Y	-
导向型 FAST 和旋转式 BRIEF 特性检测器	不可用	ORB::ORB	-	Y
HOG 检测 / 描述符	不可用	HOGDescriptor	Y	-
SVM	不可用	CvSVM::predict	Y	-

可扩展 INT8 优化可检查同时处理两个有共享系数的数据项的兼容性。SIMD 可检查能从 DSP48E2 Slice 中的四个操作数和算子受益的模块。数据和权重遵循 8 位限制的所有滤波器相关模块都能从这种可扩展 INT8 方法获益。大多数其他涉及基本图像算术（例如加 / 减或比较等）的模块能运用 DSP48E2 的 SIMD 运算。

## 使用可扩展 INT8 优化的定制 2D 卷积

在计算机视觉功能环境中，大部分预处理任务会涉及一定程度的过滤。因为图像主要使用每个通道 8 位的方式表达，深度学习应用中对 INT8 运算的优化能应用到图像处理中的二维过滤运算。唯一的局限是滤波器中的系数的精度必须能够用 8 位表达。这对诸如索贝尔、Scharr、拉普拉斯或其他边缘检测滤波器等常见滤波器而言，一般来说是正确的。

使用下列方法之一即可发挥 DSP48E2 Slice 内的双乘法器模式的作用：

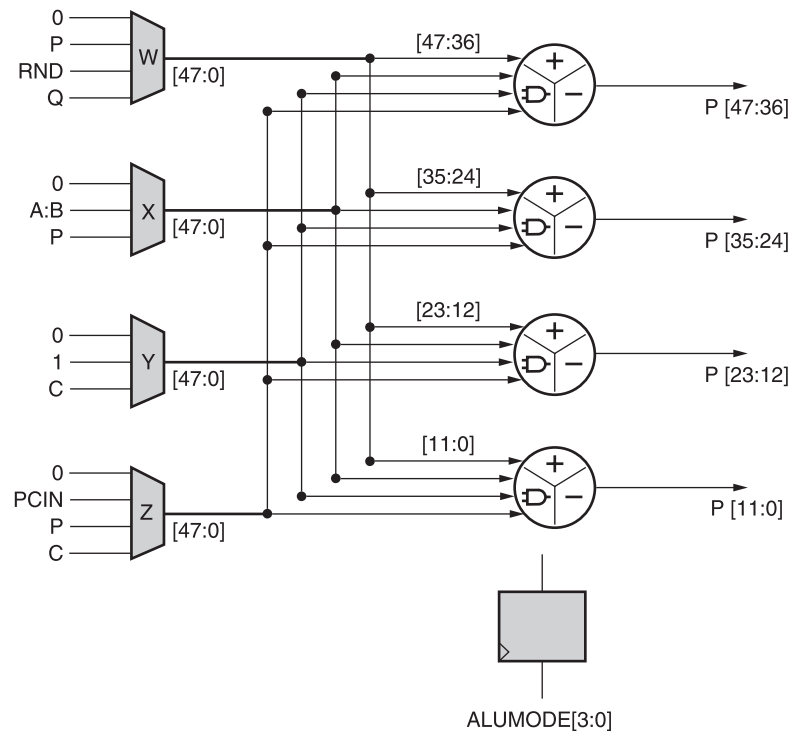
**对同一通道在多像素输出上运算：**在这种模式下可以并行计算出两个输出像素。因为在图像中滤波器系数在像素间共享，因此能同时计算位置  $(x,y)$  和  $(x,y+1)$  的像素。按照滤波器计算顺序，每个滤波器系数与两个不同输入像素相乘。这意味着在可编程逻辑中提供的资源不变的情况下，性能提高了 1.75 倍。

**对不同通道或图像的多像素输出进行运算：**如果正在处理的图像有多个通道且滤波器在不同通道间共享，对处于相同位置  $(x,y)$  的像素，滤波器的系数可在多通道间共享。相同的方法可扩展用于同时运算多个图像。

## 使用 SIMD 运算的中值滤波器

图像处理中常用的中值滤波器也用于消噪。在图像上使用中值滤波器涉及用预设大小的窗口扫描图像，计算进入该窗口的像素的中值，使用中值替换中心像素。中值计算属于高计算强度。它涉及为值排序，然后找到位于列表中间的值。排序流程是一个比较运算序列。

要使用可编程逻辑上的 DSP 实现中值滤波器，可以对算法做改动。每次比较运算可以分为减法运算及后续的符号位检查。对减法运算，DSP48E2 Slice 能够以四个 12 位或两个 24 位模式进行运算。要充分利用 DSP48E2 Slice，可以并行运算多个像素。假定每个像素为单通道，深度小于 12 位，就可以同时处理四个输出像素。对每个输出像素存在多个排序运算，这些排序运算都可以使用 DSP48E2 Slice 中的减法运算。结果的符号位可以使用最小的逻辑在 DSP48E2 Slice 之外检查。比较的总数量取决于用于排序值的算法。



WP490\_09\_020917

图 9：中值滤波器的 DSP48E2 运算模式

## 竞争分析

在竞争分析中使用英特尔（前 Altera）的 Arria 10 器件与赛灵思的 Zynq® UltraScale+™ MPSoC 对比。在进行嵌入式视觉应用计算效率比较时，选择的器件有可比的 DSP 密度和器件功耗：

- Arria 10 SoC : SX220、SX270 和 SX480
- Zynq UltraScale+ MPSoC : ZU3、ZU7 和 ZU9 器件

重点比较能用于包括深度学习和计算机视觉在内的众多应用的通用 MACC 性能。

英特尔的 MACC 性能基于运用预加法器的算子。但是这种实现方案产生的是乘积项的和，而非单独的乘积项。因此英特尔的预加法器不适用高效深度学习或计算机视觉运算。

在本计算效率分析中，每个器件的功耗使用赛灵思的 2016.4 版 Power Estimator 工具和英特尔的 16.0.1 版 EPE Power Estimate 工具进行估算，并根据下列假设得出：

1. 90% DSP 占用率
2. 英特尔器件 - 速度等级为: 2L, 最大频率下供电电压为 0.9V
3. 赛灵思器件 - 速度等级为 1L, 最大频率下供电电压为 0.72V
4. 时钟速率为 DSP Fmax 时逻辑利用率为 70%
5. 时钟速率为 DSP 最大频率的一半时,Block RAM 利用率为 90%
6. DSP 翻转率为 12.5%
7. 功耗特征：“典型功耗”

图 10 所示的即为深度学习和计算机视觉运算的能效对比。与英特尔的 Arria 10 SoC 器件相比，赛灵思器件能让深度学习和计算机视觉运算的计算效率提高 3-7 倍。

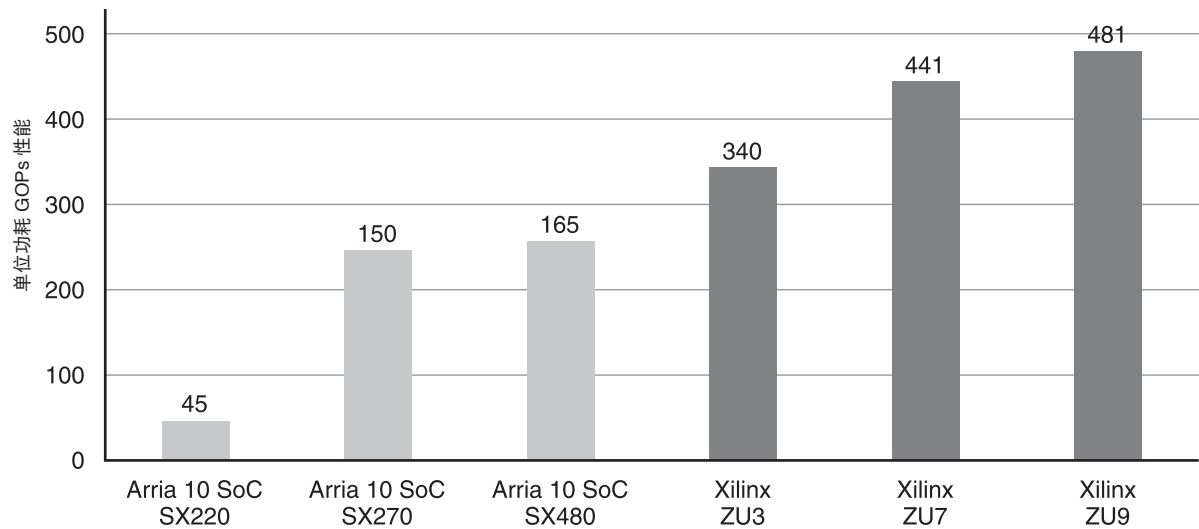


图 10 : INT8 深度学习和计算机视觉能效对比: 赛灵思对比英特尔

## 结论

本白皮书探讨了如何在赛灵思 DSP48E2 Slice 上优化 INT8 深度学习和计算机视觉运算，从而实现 1.75 倍的性能提升。赛灵思 DSP48E2 Slice 可用于在共享相同内核权重的同时处理两个并行的 INT8 MACC 运算。为高效实现 INT8，需要 24 位输入宽度，这一优势只有赛灵思 DSP48E2 Slice 支持。同样的优势还能用于计算机视觉运算，例如过滤任务及其它图像操作任务。赛灵思的 DSP48E2 Slice 的 SIMD 模式为开展四个 12 位或两个 24 位 SIMD 运算提供了新的途径。

总之，赛灵思的 Zynq UltraScale+ MPSoC 非常适用于 INT8 工作负载，使之成为为嵌入式视觉领域大量应用加速的理想选择。赛灵思不断创新新的基于软 / 硬件的方法，以加速嵌入式视觉应用领域的深度学习和计算机视觉功能。

关于赛灵思器件在嵌入式视觉方面的应用的更多信息，敬请访问：

[china.xilinx.com/revision](http://china.xilinx.com/revision)

## 参考资料

1. Dettmers, 8-Bit Approximations for Parallelism in Deep Learning, ICLR 2016  
<https://arxiv.org/pdf/1511.04561.pdf>
2. Gysel et al, Hardware-oriented Approximation of Convolutional Neural Networks, ICLR 2016  
<https://arxiv.org/pdf/1604.03168v3.pdf>
3. Han et al, Deep Compression: Compressing Deep Neural Networks With Pruning, Trained Quantization And Huffman Coding, ICLR 2016  
<https://arxiv.org/pdf/1510.00149v5.pdf>
4. Khronos Group, <https://www.khronos.org/openvx/>
5. UG579, *UltraScale Architecture DSP Slice User Guide*
6. Rosenblatt, F., The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review*, Vol. 65, No. 6, 1958  
<http://www.ling.upenn.edu/courses/cogs501/Rosenblatt1958.pdf>

## 修订历史

下表列出了本文档的修订历史：

日期	版本	修订描述
2017 年 3 月 10 日	1.0	赛灵思初始版本

## 免责声明

本文向贵司 / 您所提供的信息（下称“资料”）仅在选择和使用赛灵思产品时供参考。在适用法律允许的最大范围内：(1) 资料均按“现状”提供，且不保证不存在任何瑕疵，**赛灵思在此声明对资料及其状况不作任何保证或担保，无论是明示、暗示还是法定的保证，包括但不限于对适销性、非侵权性或任何特定用途的适用性的保证**；且 (2) 赛灵思对任何因资料发生的或与资料有关的（含对资料的使用）任何损失或赔偿（包括任何直接、间接、特殊、附带或连带损失或赔偿，如数据、利润、商誉的损失或任何因第三方行为造成的任何类型的损失或赔偿），均不承担责任，不论该等损失或者赔偿是何种类或性质，也不论是基于合同、侵权、过失或是其他责任认定原理，即便该损失或赔偿可以合理预见或赛灵思事前被告知有发生该损失或赔偿的可能。赛灵思无义务纠正资料中包含的任何错误，也无义务对资料或产品说明书发生的更新进行通知。未经赛灵思公司的事先书面许可，贵司 / 您不得复制、修改、分发或公开展示本资料。部分产品受赛灵思有限保证条款的约束，请参阅赛灵思销售条款：<http://china.xilinx.com/legal.htm#tos>；IP 核可能受赛灵思向贵司 / 您签发的许可证中所包含的保证与支持条款的约束。安全保护功能，不能用于任何需要专门故障安全保护性能的用途。如果把赛灵思产品应用于此类特殊用途，贵司 / 您将自行承担风险和责任。请参阅赛灵思销售条款：<http://china.xilinx.com/legal.htm#tos>。

## 汽车应用免责声明

汽车产品（产品部件号中标识为“XA”）不保证用于安全气囊的开发或用于影响车辆控制的应用（“安全应用”），除非在该赛灵思产品中具备故障安全保护或者额外功能，符合 ISO 26262 汽车安全标准（“安全设计”）。为安全起见，客户应在使用或分销任何集成有该产品的系统之前，对这些系统进行全面测试。在没有安全设计的安全应用中，使用产品的风险完全由客户承担，仅受有关产品责任的适用法律和法规限制。