WP506 (v1.1) 2020 年 7 月 10 日

赛灵思 AI 引擎及其应用

针对 5G 蜂窝和机器学习 DNN/CNN 等计算密集型应用，赛灵思的新型矢量处理器 AI 引擎由 VLIW SIMD 高性能处理器阵列构成，与传统的可编程逻辑解决方案相比，功耗减半，芯片计算密度提升高达 8 倍。

概要

本白皮书探讨了将赛灵思新型 AI 引擎用于 5G 蜂窝和机器学习 DNN/CNN 等计算密集型应用的架构、应用和优势。

在与过去数代技术进行比较时，5G 要求将计算密度提高 5 倍到 10 倍；AI 引擎已针对 DSP 进行优化，同时满足吞吐量和计算要求，为无线连接提供高带宽和速度提升。

机器学习在众多产品中方兴未艾，特别是在 DNN/CNN 网络中。这大幅提升了对计算密度的要求。AI 引擎专为线性代数优化，提供可满足这些需求的计算密度，并且与可编程逻辑上运行的类似功能相比，还能降低功耗高达 50%。

AI 引擎采用众多程序员熟悉的 C/C++ 语言编程。AI 引擎与赛灵思自适应标量引擎集成，旨在提供高度灵活、功能强大的整体解决方案。

赛灵思悠久的计算技术发展史

赛灵思产品用于计算密集型应用已有数十年的历史，最早可追溯到上世纪 90 年代早期的高性能计算 (HPC) 和数字信号处理 (DSP) 实现方案。赛灵思 XC4000 系列 FPGA 已经成为了商用、航空航天与国防无线通信系统实施数字前端 (DFE) 解决方案的实现技术。这些早期用户使用 LUT 和加法器实现计算单元（如乘法器），构建 DSP 功能、FIR 滤波器和 FFT。

随着客户开始选择使用赛灵思器件来处理要求严格的新型计算应用，首个“DSP 片”这样的计算密集型专用单元，跟随 2001 年 Virtex®-II 系列 FPGA 的问世被开发出来。根据摩尔定律，赛灵思已经将 LUT 的数量从 XC4000 FPGA 中的仅 400 个提升到当前器件中的超过 370 万个 LUT 和超过 12,200 个 DSP 片，可用资源的增加超过 9,500 倍。在这样的计算资源加速增长的推动下，赛灵思产品始终能为信号处理市场的最新发展提供所需的计算密度和逻辑资源。

技术进步推高计算密度

多项技术的发展正在促使对更高非线性计算密度的需求。每秒千兆赫采样率的数据转换器能直接采样 RF 信号，简化模拟系统，但需要相应数量级的更高的 DSP 计算密度。直接 RF 采样与使用多个天线相结合，例如拥有数万个天线的先进雷达系统。

热门的 5G 无线技术已酝酿数年。该技术有望通过将环境里的一切事物连接到一个网络改变人们的生活：这个网络，速度比蜂窝连接快100倍，比最快的家用宽带服务快 10 倍。毫米波、大规模 MIMO、全双工、波束成型和小蜂窝只是实现超高速 5G 网络的部分技术。速度与低时延是 5G 的两大优势，从自动驾驶汽车到虚拟现实，拥有广阔的新应用空间。这些技术带来的计算密度要求和存储器要求比 4G 高出整整一个数量级。

随着 5G 技术的发展，大规模 MIMO、多天线、多频带等新技术所导致的复杂性比 4G 高百倍。不断提高的复杂性直接推高计算密度、存储器要求和 RF 数据转换器性能。参见图 1。

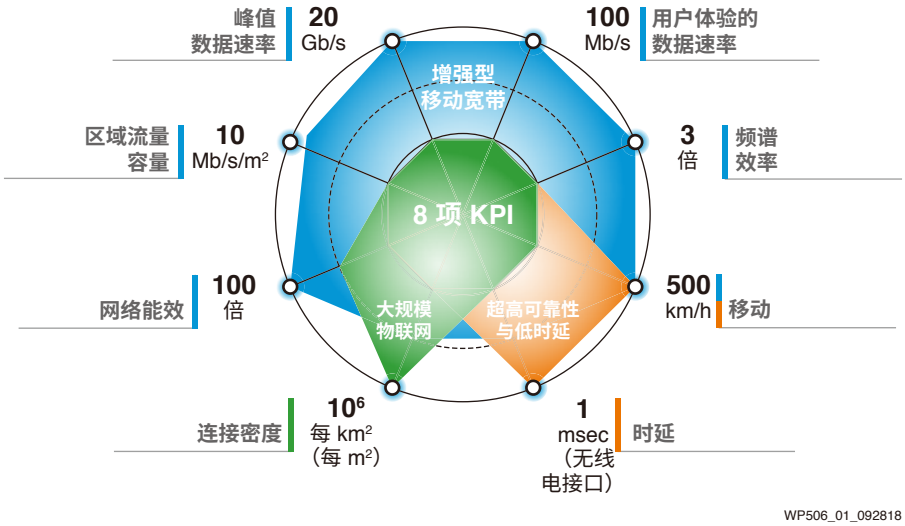

图 1： **5G 与 4G 复杂性比较**[1]

1. ETRI RWS-150029，5G 视觉与实现技术：ETRI 视角，3GPP RAN 研讨会，凤凰城，2015 年 12 月：http://www.3gpp.org/ftp/tsg_ran/TSG_RAN/TSGR_70/Docs

摩尔定律的终结

1965 年，英特尔联合创始人戈登摩尔预言，集成电路中的器件数量每两年将翻一番。1965 年，每芯片 50 个晶体管能提供最低的每晶体管成本。摩尔预言，到 1970 年每芯片晶体管数量将达 1,000 个且每晶体管成本将下降 90% 以上。摩尔后来把这一预言调整为数量每两年翻一番。这在从 1975 年到 2012 年的时间段里基本保持正确。(1)摩尔定律认为每个新的更小的工艺节点能提供更高密度、性能并降低功耗、成本。这一观点被称为“摩尔定律”，并在大约 50 年的时间里一直适用。摩尔定律原理是 IC 密度、性能和合理价格不断发展的驱动力，也是赛灵思不断推出更低成本更高性能器件所一直遵循的原理。

随着 IC 工艺节点达到 28nm 及以下，摩尔定律开始“失效”。在更小工艺节点上生产的器件不再能轻松降低功耗、成本并提高性能。在 5G 蜂窝系统的计算需求和可编程逻辑计算密度之间出现了鸿沟。第 5 代蜂窝提出的成本、功耗和性能要求超过了可编程逻辑达成系统级目标的能力。

迎来 AI 引擎

为满足新一代无线通信应用和机器学习应用对提高计算密度、降低功耗水平的需求的迅猛增长，赛灵思开始研发创新型架构，最终开发出 AI 引擎。AI 引擎结合自适应引擎（可编程逻辑）和标量引擎（处理器子系统），构成紧密集成的异构计算平台。AI 引擎为基于矢量的算法带来了高达五倍的计算密度提升。自适应引擎提供灵活的定制计算和数据传输。标量引擎提供复杂的软件支持。参见图 2。

1. Wikipedia.org，“摩尔定律”，https://en.wikipedia.org/wiki/Moore%27s_law，2018 年 8 月检索。

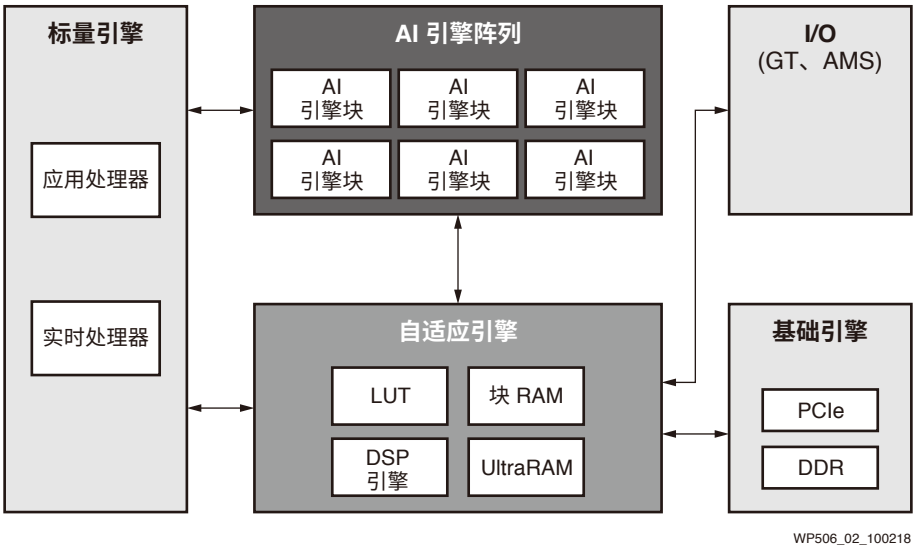

图 2： **异构计算**

图 3 所示的是 AI 引擎接口块构成的 2D 阵列。

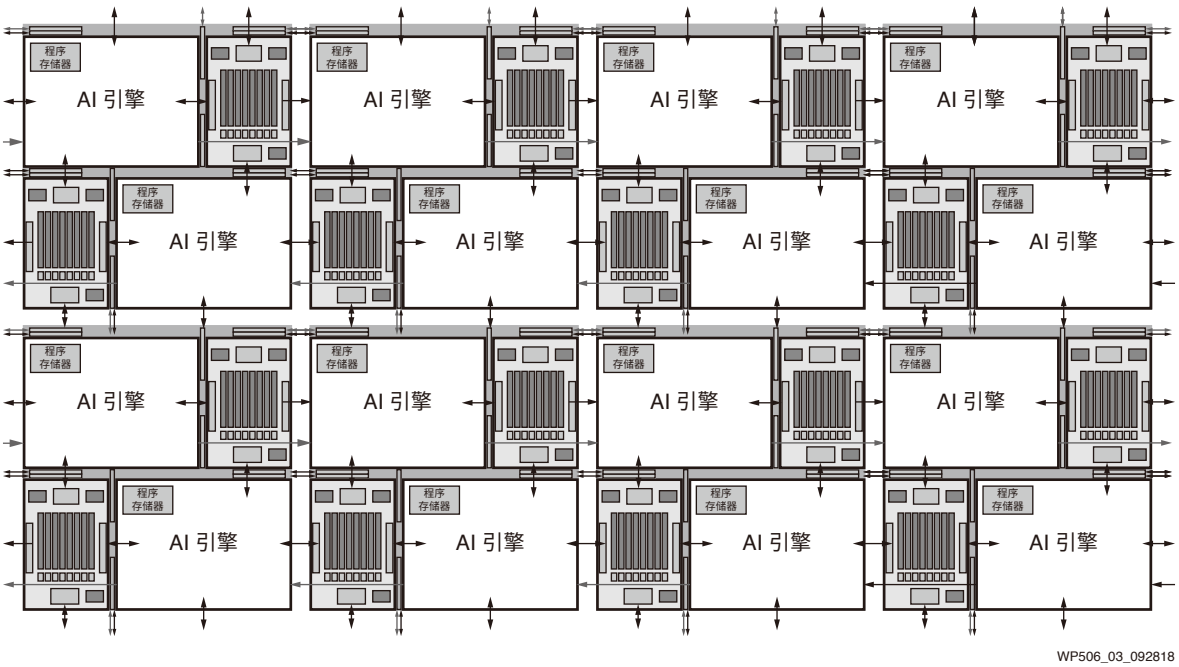

图 3： **AI 引擎阵列**

每个 AI 引擎块内置了用于定点和浮点运算的矢量处理器、一个标量处理器、专用程序和数据存储器、专用 AXI 数据传输通道以及对 DMA 和锁的支持。AI 引擎是一种单指令多数据 (SIMD)；并且是超长指令字 (VLIW)，每时钟周期提供多达 6 路指令并行化，包括两个/三个标量运算、两个矢量载荷和一个写入运算以及一个定点或浮点矢量运算。

专为实时 DSP 和 AI/ML 计算优化的 AI 引擎阵列，通过专用数据和指令存储器、DMA、锁和软件工具的结合提供确定性时序。专用数据存储器和指令存储器属于静态存储器，能消除高速缓存缺失和相关填充产生的不一致。

AI 引擎的目的和目标

AI 引擎的目的和目标由使用 DSP 和 AI/ML 的高计算强度应用决定。其他的市场需求还包括更高的开发者生产力、更高的抽象水平，这正在推动开发工具的演进发展。AI 引擎的开发旨在提供四大优势：

- 在实现计算密集型应用方面，与 PL 实现方案相比提供单位芯片面积多三到八倍的计算容量
- 与实现在 PL 中的相同功能相比，将计算密集型应用的功耗减少 50%
- 提供确定性、高性能、实时 DSP 功能
- 显著改善开发环境，助力设计人员提高生产力

AI 引擎块架构详解

要真正理解 AI 引擎的大量功能，必须对其架构和功能建立整体理解。图 4 所示的 AI 引擎块详细体现了每个块里的资源：

- 专用的 16KB 指令存储器和 32KB RAM
- 32 位 RISC 标量处理器
- 512 位定点和 512 位浮点矢量处理器（带有相关的矢量寄存器）
- 同步处理器
- 追溯与调试

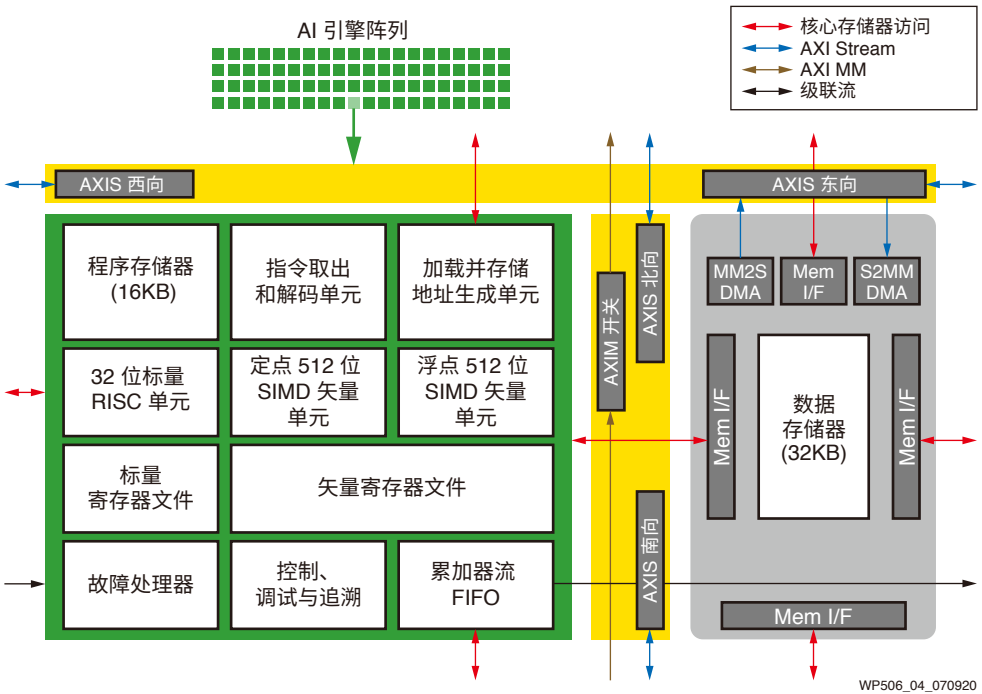

图 4： **AI 引擎块详解**

综合使用专用 AXI 总线布线和直接连接，连接到相邻的 AI 引擎块，内置指令和数据专用存储器的 AI 引擎与其他 AI 引擎块互联。在数据传输方面，专用 DMA 引擎和锁直接连接到专用 AXI 总线，实现连接、数据传输和同步。

操作数精度支持

矢量处理器由整数单元和浮点单元构成。支持 8 位、16 位、32 位操作数和单精度浮点 (SPFP)。对于不同的操作数，每时钟周期操作数数量有变化，详见表 1。

表 1： **AI 引擎矢量精度支持**

操作数 A	操作数 B	输出	每时钟周期 MAC 数
8 位实数	8 位实数	16 位实数	128
16 位实数	8 位实数	48 位实数	64
16 位实数	16 位实数	48 位实数	32
16 位实数	16 位复数	48 位复数	16
16 位复数	16 位复数	48 位复数	8
16 位实数	32 位实数	48/80 位实数	16
16 位实数	32 位复数	48/80 位复数	8
16 位复数	32 位实数	48/80 位复数	8
16 位复数	32 位复数	48/80 位复数	4
32 位实数	16 位实数	48/80 位复数	16
32 位实数	16 位复数	48/80 位复数	8
32 位复数	16 位实数	48/80 位复数	8
32 位复数	16 位复数	48/80 位复数	4
32 位实数	32 位实数	80 位实数	8
32 位实数	32 位复数	80 位复数	4
32 位复数	32 位实数	80 位复数	4
32 位复数	32 位复数	80 位复数	2
32 位 SPFP	32 位 SPFP	32 位 SPFP	8

指令与数据并行

通过指令级和数据级并行能实现多种层次的并行。

指令级并行如图 5 所示。在每个时钟周期，执行两个标量指令、两个矢量读取、一个单矢量写入和一个单矢量指令，即 6 路 VLIW。

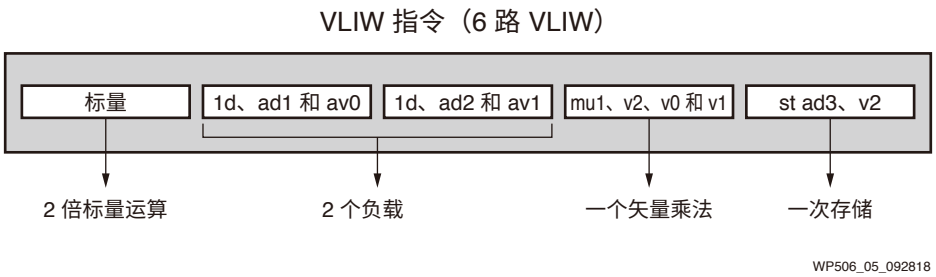

图 5： **AI 指令级并行**

数据级并行通过矢量级并行实现，即在每个时钟周期内运算多个数据集，如表 1 所示。

确定性性能与连接

AI 引擎架构专为要求确定性性能的实时处理应用开发。确保确定性时序的两项关键架构特性是：

- 专用指令和数据存储器
- 与 DMA 引擎配对的专用连接，使用 AI 引擎块间的连接实现有调度的数据传输

直接存储器 (DM) 接口提供指定 AI 引擎块与其北、南和西向 AI 引擎块数据存储器之间的直接访问。这一般用于在整个处理链产生和/或消费数据之际，向矢量处理器输入/从矢量处理器输出结果。实现数据存储器的目的是形成“交替”缓存方案，从而最大限度减轻存储器争用对性能的影响。

AI 引擎块间的 AXI-Stream 和 AXI-存储器映射连接

AI 引擎块间的数据传输的最简方式是通过直接相邻的 AI 引擎块间的共享存储器。然而，如果 AI 引擎块间距离遥远，那么 AI 引擎块就需要使用 AXI-Streaming 数据流。AXI-Streaming 连接由 AI 引擎编译器工具根据数据流图预先定义并编程。此外，这些流接口也能用来直接接口连接 PL 和片上网络 (NoC)。参见图 6。

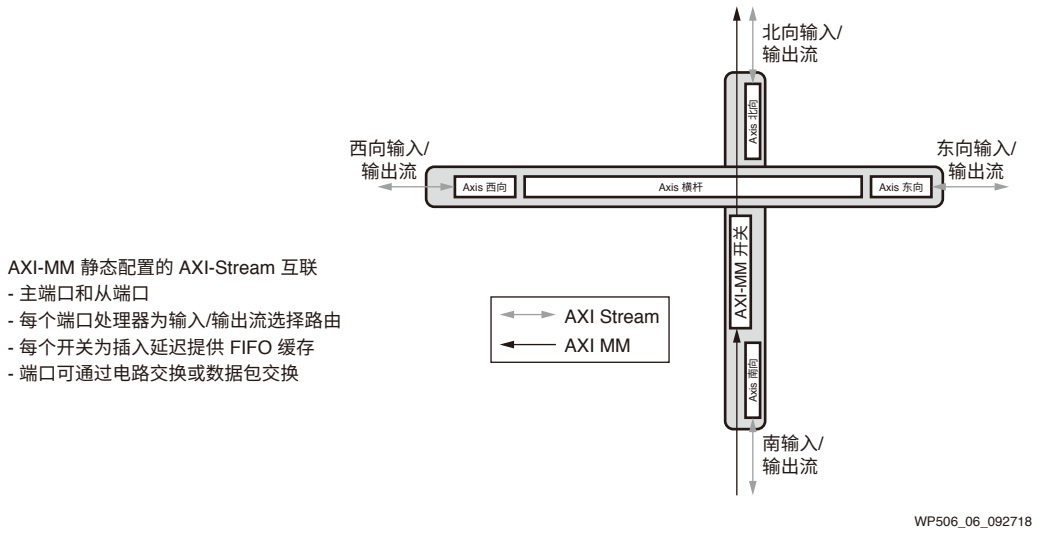

图 6： **AI 引擎阵列 AXI-MM 和 AXI-Stream 互联**

AI 引擎和 PL 连接

Versal 产品组合最突出的优势之一，是能够在自适应引擎中将 AI 引擎阵列与可编程逻辑结合使用。这样的资源结合为在最佳资源、AI 引擎、自适应引擎或标量引擎中实现功能提供了极大的灵活性。图 7 所示的是 AI 引擎阵列和可编程逻辑间的连接，也称为“AI 引擎阵列接口”。AXI-Streaming 连接存在于 AI 引擎阵列接口的每一侧，并分别延伸连接至可编程逻辑和 NoC 内。

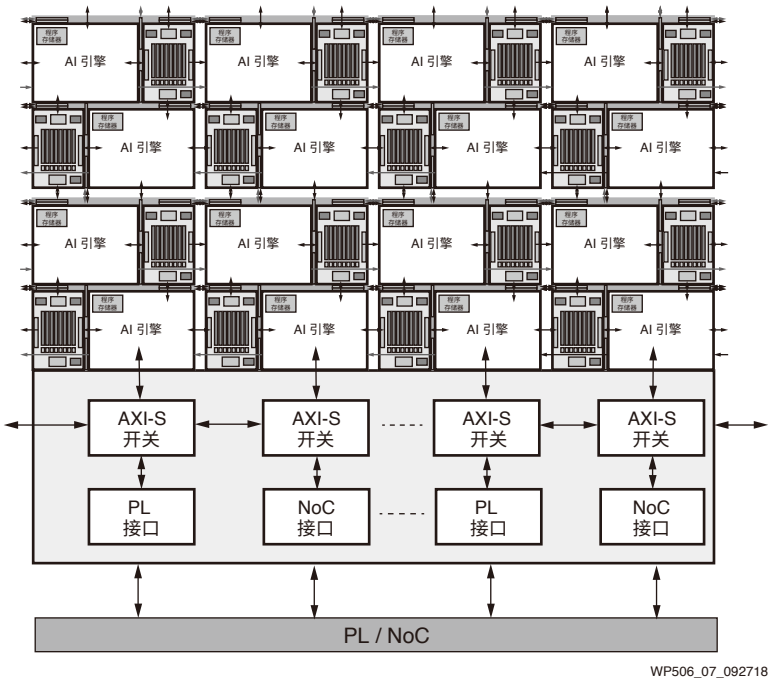

图 7： **AI 引擎阵列接口**

AI 引擎控制、调试与追溯

控制、调试与追溯功能集成在每一个 AI 引擎块里，为调试、性能监控和性能优化提供可见性。通过在 Versal 产品组合中推出的高速调试端口就能调用调试功能。

AI 引擎与可编程逻辑实现方案对比

AI 引擎的目的和目标一节介绍了评估是否满足应用需求和市场需求的指标。通过将 4G 和 5G 蜂窝分别实现在 PL 和 AI 引擎内，即可衡量该架构的效果。结果总结说明基于 AI 引擎的解决方案能够提供：

- 相对于在相同工艺节点上采用 PL 实现的相同功能，芯片占用面积降低 3 到 8 倍
- 功耗与 PL 实现方案相比降低大约 50%

对于不适合采用矢量实现的功能，AI 引擎的效率大幅降低，AI 引擎往往并非是理想选择。在这些情况下，PL 是更优秀的解决方案。AI 引擎和 PL 旨在当作对等计算单元进行使用，每个分别处理与其优势相匹配的功能。PL 非常适合数据传输、数位型功能和非矢量型运算；PL 也能为非 AI 引擎支持的操作实现定制加速器。PL 和 AI 引擎相辅相成，构成更强大的系统级解决方案。在大多数计算密集型应用中，可编程逻辑仍然是一种非常宝贵的资源。AI 引擎与 PL 的结合，能提供灵活性、优异的计算性能、高带宽数据传输和存储。

配备 AI 引擎架构的 Versal 产品组合概述

Versal 器件包括三种类型的可编程处理器：Arm® 处理器子系统 (PS)、可编程逻辑 (PL) 和 AI 引擎。每一种都提供不同的计算功能，以满足总体系统不同部分的需求。Arm 处理器一般 用于控制平面应用、操作系统、通信接口和较低级别的运算或复数运算。PL 负责数据操作与传输、非矢量型计算和接口。AI 引擎一般用于矢量实现方案中的计算密集型功能。

图 8 是 AI 引擎阵列布局在器件顶层的 Versal 器件的高级视图。AI 引擎阵列与 PL 之间的连接通过直接连接和 NoC 实现。

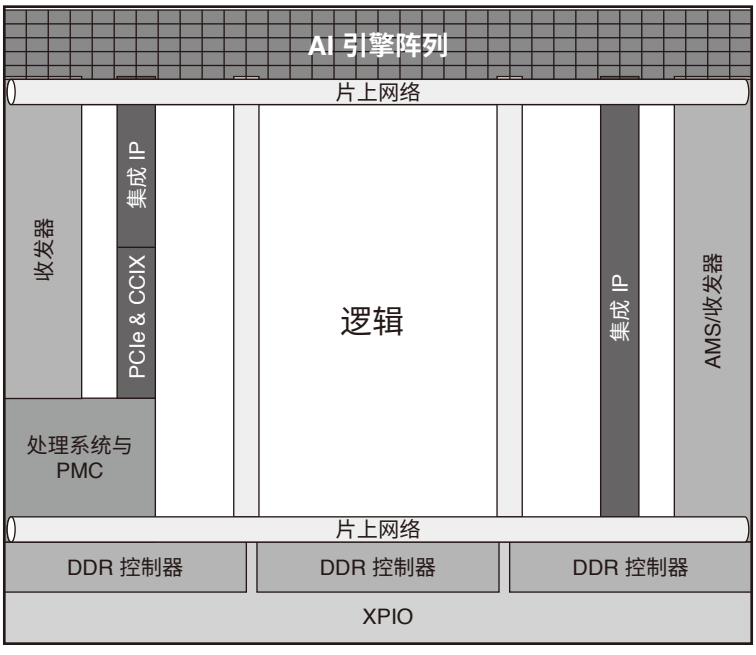

图 8： **配备 AI 引擎架构的 Versal ACAP 概略图**

AI 引擎开发环境

近年来，赛灵思高度重视使用高层次语言 (HLL) 来提升使用赛灵思器件进行开发的抽象水平。Versal 架构拥有三个完全不同的可编程单元：PL、PS 和 AI 引擎。所有三个单元都可以使用 C/C++ 编程。

使用基于 x86 的仿真环境，能对 AI 引擎开展功能仿真或周期精度仿真。对于系统级仿真，提供支持全部三种处理域的 System-C 虚拟平台。

开发环境中的一个关键元素是 AI 引擎库。该库能够为 DSP 和无线功能、ML 和 AI、线性代数和矩阵数学提供支持。这些库专为效率和性能进行优化，以便开发者充分发挥 AI 引擎功能的全部优势。

AI 引擎应用

AI 引擎专门为计算密集型应用优化，特别是数字信号处理 (DSP) 和某些人工智能 (AI) 技术，如机器学习 (ML) 和 5G 无线应用。

使用 AI 引擎开展数字信号处理

无线电解决方案验证套件

实时 DSP 在无线通信中得到了广泛的应用。赛灵思通过比较经典窄带和宽带无线电设计原则、大规模 MIMO 以及基带和数字前端概念的实现方案，证明 AI 引擎架构适用于构建无线电解决方案。

示例：100MHz 五通道 LTE20 无线解决方案

100MHz 五通道 LTE20 无线解决方案在 Versal 器件的组成部分中得到实现。五个通道的 16 位输入数据以 30.72MSPS 的速率传输并在 89 抽头通道滤波器中进行处理。信号接着使用两级半带滤波器（23 抽头和 11 抽头）四倍升频采样，得到 122.88MSPS 的采样速率。

升频采样后的流接着与直接数字综合 (DDS) 正弦/余弦波函数混合并求和。另外两个半带滤波器（47 抽头和 27 抽头）提供总共四倍采样，为峰值因数衰减 (CFR) 函数产生 491.52MSPS 输入流。41 抽头滤波器提供分数比率改变、五倍升频/四倍降频，为数字预失真函数 (DPD) 产生 614.4MSPS 输入采样率。

峰值检测器/比例尺发现 (PD/SF) 电路在 PL 中实现。491.52MSPS DUC 和混合器级的输出构成其输入之一，CFR 第二级为其提供第二个输入。PD/SF 电路如果实现在 PL 中，则资源效率高；相反，如果实现在 AI 引擎里，则资源效率低。这充分体现出如何制定架构决策，以针对设计的不同功能模块最高效地运用资源。参见图 9。

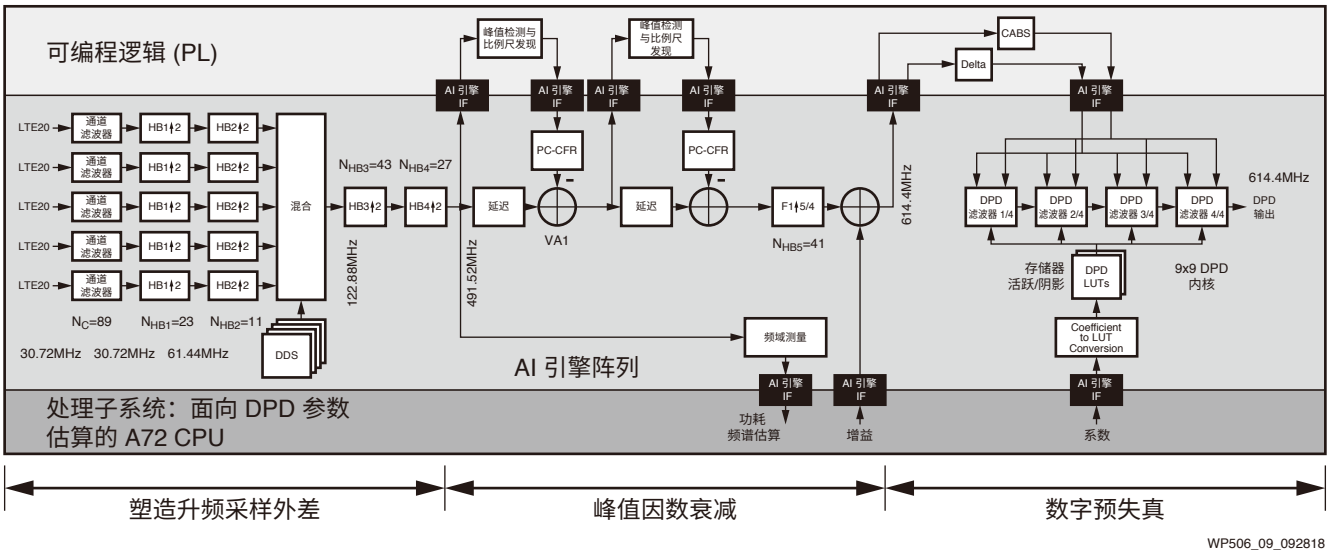

图 9： **原理图采用 DSP 的 100MHz 五通道 LTE20 无线解决方案**

DPD 功能需要周期性地重新计算系数。使用模数转换器 (ADC) 从传输数模转换器 (DAC) 的输出中采样反馈路径，并缓存。缓存的样本数据集被传递给 PS，用于每秒 10 次计算新的 DPD 系数集。新的系数集使用片上网络和 AXI 总线连接写回 DPD。

机器学习与 AI 引擎

在机器学习中，卷积神经网络 (CNN) 是一类深度前馈人工神经网络，最常用于分析视觉图像。随着计算机正在被广泛用于从自动驾驶汽车到视频监控直至图像和视频的数据中心分析等万事万物，CNN 已成为必备要素。CNN 提供的技术突破在于让视觉图像的可靠性和准确度足以用于安全地驾驶车辆。

CNN 技术目前处于初期阶段，几乎每个星期都有新突破涌现。这个领域的创新节奏令人惊叹。这意味在未来几年里就有望实现前所未有的全新应用。

然而，CNN 面临的挑战在于所需的高强度计算，通常需要数 TeraOPS。AI 引擎的优化正是为了低成本、高能效地实现这样的计算密度。

AI 引擎 CNN/DNN 叠加

赛灵思正在开发一种在 AI 引擎上构建的机器学习推断引擎，并将作为一种应用叠加应用。可编程逻辑用于高效地传输和管理数据。AI 引擎应用叠加为执行计算和实现众多流行的 CNN/DNN 网络，如 ResNet、GoogLeNet 以及 AlexNet 等所需的其他运算提供所需的界定清晰的结构。

站在用户角度来看，叠加方法拥有众多优势，包括在更加新颖的网格架构出现后进行修改的能力。AI 引擎和 PL 的可编程结合提供了一种高效且极为灵活的平台，能够随着 ML 应用空间的发展而成长和扩展。

AI 引擎 CNN/DNN 叠加既适用于加速 ML 网络推断的数据中心应用，也适用于嵌入式系统。将该解决方案实例化到用户的整体设计中就能方便地完成集成。随后使用 TensorFlow 或 Caffe 开发 CNN/DNN 神经网络，编译成在AI引擎CNN/DNN叠加上运行的可执行程序。

总结

AI 引擎是新一类高性能计算的代表。AI 引擎集成在 Versal 级别的器件中，能与 PL 和 PS 最优结合，在单个赛灵思 ACAP 中实现高度复杂的系统。实时系统需要确定性行为。对此 AI 引擎通过结合专用数据和编程存储器、DMA 与锁以及编译工具予以实现。

与传统的可编程逻辑 DSP 与 ML 实现方案相比，AI 引擎的单位芯片面积计算密度提高了3-8倍，同时在名义上降低功耗 50%。C/C++ 编程模式可提高抽象水平，有望大幅提升开发者的生产力。

从内置 30 个 AI 引擎和 8 万 LUT 的小型器件到内置 400 个 AI 引擎和近百万个 LUT 的大型器件，系统性能可扩展性通过系列器件得以实现。这些器件间封装脚位兼容，方便在产品系列内进行迁移，以满足不同的性能目标和价格目标要求。

如需了解更多信息，请参阅：

WP505，*《Versal：首个自适应计算加速平台 (ACAP)》*

WP504，*《采用赛灵思 Alveo™ 加速器卡为 DNN 提速》*

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
07/10/2020	1.1	更新图 4。
10/03/2018	1.0.2	仅编辑更新。
10/02/2018	1.0.1	仅编辑更新。
10/02/2018	1.0	赛灵思初始版本。

免责声明

关于与汽车相关用途的免责声明