WP492 (v1.0.1) 2017 年 6 月 13 日

Xilinx 全可编程器件：出色的计算密集型系统开发平台

作者：Cathal Murphy 和 Yao Fu

赛灵思 All Programmable FPGA 和 SoC 针对一系列计算密集型工作负载提供最高效、最具成本效益、时延最低、最具设计灵活性并且满足未来需求的计算平台。

摘要

为了满足不断攀升的数据处理需求，未来系统需要在计算能力上大幅改进。传统解决方案（例如 x86 处理器）再也无法以高效、低成本的方式提供所需的计算带宽，系统设计人员必须寻找新的计算平台。

FPGA 和 GPU 越来越多地被系统设计人员看好，认为它们能够满足未来需求的计算平台。

为新时代提供必要的计算效率和灵活性，本白皮书将对 GPU 以及赛灵思 FPGA 和 SoC 器件进行分析。

简介

未来系统（例如云数据中心 [DC] 和自动驾驶汽车）需要在计算能力上大幅改进，以支持不断增多的工作负载以及不断演进的底层算法 [参考资料 1]。例如，大数据分析、机器学习、视觉处理、基因组以及高级驾驶员辅助系统 (ADAS) 传感器融合工作负载都在促使计算性能能以低成本、高效的方式实现提升，并且超出现有系统（例如 x86 系统）的极限。

系统架构师正在寻找能满足要求的新计算平台。该平台需要足够灵活，以便集成到现有的架构中，并支持各种工作负载及其不断演进的算法。此外，这些系统很多还必须提供确定性的低时延性能，以支持实时系统（例如自动驾驶汽车）所需的快速响应时间。

图形处理单元 (GPU) 厂商非常积极地将 GPU 定位成新时代计算平台的最佳之选，主要依据其在机器学习训练的高性能计算 (HPC) 领域取得的成功。在此过程中，GPU 厂商针对机器学习推断工作负载修改了他们的架构。

然而，GPU 厂商还是忽视了基本的 GPU 架构的局限性。这些局限性会严重影响 GPU 以高效、低成本方式提供必要的系统级计算性能的能力。例如，在云端 DC 系统中，对工作负载的需求在一天内会发生很大变化。此外，这些工作负载的底层算法也会发生快节奏变化。GPU 架构的局限性会阻止很多今天的工作负载和明天形成的工作负载映射到 GPU，导致硬件闲置或低效。本白皮书的“GPU 架构的局限性”部分对这些局限性进行了更详细介绍。

相反，赛灵思 FPGA 和 SoC 具有众多关键属性，使它们非常适合解决未来系统要求所提出的种种挑战。这些独特属性包括：

- 针对所有数据类型提供极高的计算能力和效率
- 具备极高灵活性，能够针对多种工作负载将计算和效率优势最大化
- 具备 I/O 灵活性，能方便地集成到系统中并实现更高效率
- 具备大容量片上存储器高速缓存，可提高效率并实现最低时延

本白皮书的“赛灵思 FPGA 和 SoC 的独特优势”章节介绍了赛灵思架构的优势，并与 GPU 架构及其局限性进行对比。

GPU 起源和目标工作负载

GPU 的起源要追溯到 PC 时代，英伟达 (NVidia) 公司声称在 1999 年推出世界首款 GPU，但有很多其他显卡要先于该公司的出品 [参考资料 2]。GPU 是一款全新设计的产品，用来分担 / 加速图形处理任务，例如替 CPU 进行像素阵列的阴影和转换处理，其架构非常适合高并行吞吐量处理 [参考资料 3]。本质上，GPU 的主要作用是为视觉显示器 (VDU) 渲染高质量图像。

多年来，少量非图形的大规模并行和存储器相关工作负载是在 GPU（而非 CPU）上实现并且受益良多，例如需要大规模矩阵计算的医疗成像应用。GPU 厂商意识到他们可以将 GPU 的市场延伸到非图形应用领域，并导致 GPU 的非图形编程语言（诸如 OpenCL）应运而生。这些编程语言实际上是将 GPU 转化成了通用 GPU (GPGPU)。

机器学习

最近，能够良好映射到 GPU 实现方案的工作负载之一就是机器学习训练。通过充分运用 GPU，显著缩短了深度神经网络的训练时间。

GPU 厂商试图利用机器学习训练方面的成功来助推其在机器学习推断上的发展（部署经过训练的神经网络）。随着机器学习算法和所需数据精度的发展演进，GPU 厂商一直在调整他们的架构以保持自身地位优势。例如，英伟达在他们的 Tesla P4 产品中提供 INT8 支持。然而，即使是更低的精度，例如二进制和三进制，今天也正在被很多用户探索 [参考资料 4]。要利用机器学习及其它领域的进步，GPU 用户必须等待新硬件推出之后购买新硬件。正如本白皮书后面所述，赛灵思 FPGA 和 SoC 的用户则无需等待或购买新硬件，因为这类产品本身就具有高度的灵活性。

GPU 厂商想使自身成为这个新计算时代的首选计算平台，机器学习是他们的基础。但要弄清楚 GPU 是否适合未来系统，还要做更全面的系统级分析，需要考虑 GPU 架构的很多局限性以及系统要求如何随时间发展演进。

GPU 架构的局限性

本部分将深入研究典型的 GPU 架构，以揭示它的局限性以及如何将它们应用于各种算法和工作负载。

SIMT ALU 阵列

图 1 给出了典型的 GPU 方框图。通用 GPU 计算功能的核心是大型的算数逻辑单元 (ALU) 或内核阵列。这些 ALU 通常被认为是单指令多线程 (SIMT)，类似于单指令多数据 (SIMD)。

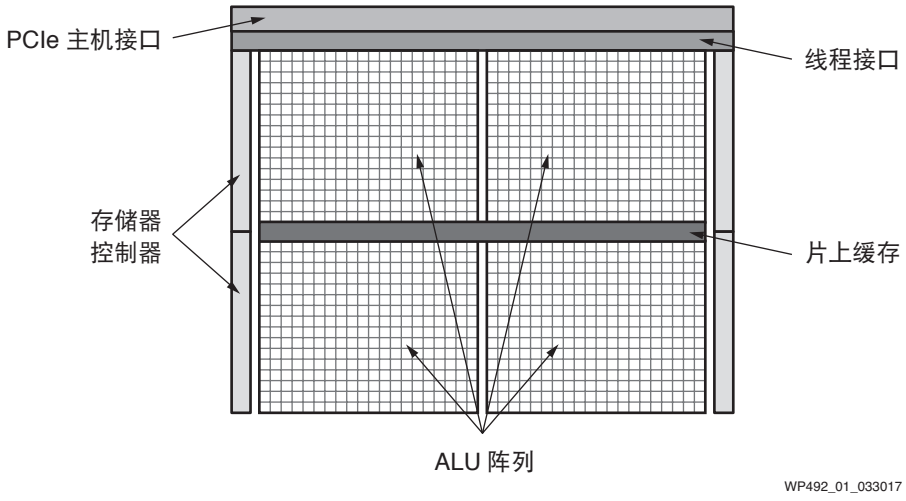

图 1： **GPU 方框图**

基本原理是将工作负载分成数千个并行的线程。需要大量 GPU 线程来防止 ALU 闲置。然后，对这些线程进行调度，以使 ALU 组并行执行同一（单个）指令。利用 SIMT，GPU 厂商能实现相对 CPU 占位面积更小和能效更高的方案，因为内核的很多资源都可与相同组中的其他内核共享。

然而，显然只是特定的工作负载（或部分工作负载）能被高效映射到这种大规模并行架构中 [参考资料 5]。如果构成工作负载的线程不具有足够的共性或并行性（例如连续工作负载或适度并行工作负载），则 ALU 会闲置，导致计算效率降低。此外，构成工作负载的线程预期要最大化 ALU 利用率，从而产生额外的时延。即使有英伟达的 Volta 架构中的独立线程调度这样的功能，底层架构也保持 SIMT，也需要大规模并行工作负载。

对于连续、适度并行或稀疏工作负载，GPU 提供的计算功能和效率甚至低于 CPU [参考资料 6]。例如用 GPU 实现稀疏矩阵计算；如果非零元素数量较少，则从性能和效率角度看 GPU 低于或等同于 CPU [参考资料 7][参考资料 8]。

有趣的是，很多研究人员正在研究稀疏卷积神经网络，以利用很多卷积神经网络中的大规模冗余 [参考资料 9]。这种趋势显然在机器学习推断领域向 GPU 提出了挑战。

稀疏矩阵计算也是大数据分析中的关键环节。[参考资料 10]

包含大量并行计算任务的大多数工作负载也包含一些连续或适度并行元素，意味着需要 GPU-CPU 混合系统来满足系统性能要求 [参考资料 11]。显然，高端 CPU 需求会影响平台的效率和成本效益，CPU 与 GPU 之间的通信也会给系统增加潜在瓶颈。

SIMT/GPU 架构的另一个局限性是 ALU 的功能取决于它的固定指令集和所支持的数据类型。

离散数据类型精度支持

系统设计人员正在探索简化数据类型精度，以此实现计算性能的跳跃式提升，而且不会使精度明显降低 [参考资料 12][参考资料 13][参考资料 14]。

机器学习推断在降低精度方面一马当先，首先是 FP16，然后是 INT16 和 INT8。研究人员正在探索进一步降低精度，甚至降到二进制 [参考资料 4][参考资料 15]。

GPU ALU 通常原生支持单精度浮点类型 (FP32)，有些情况支持双精度浮点 (FP64)。FP32 是图形工作负载的首选精度，而 FP64 经常用于一些 HPC 用途。低于 FP32 的精度通常无法在 GPU 中得到有效支持。因此采用标准 GPU 上的更低精度，除了能减少所需存储器带宽以外，作用甚微。

GPU 通常提供一些二进制运算功能，但通常只能每 ALU 进行 32 位宽运算。32 位二进制运算存在很大的复杂性和面积需求。在二值化神经网络中，算法需要 XNOR 运算，紧接着进行种群 (population) 计数。NVidia GPU 只能每四个周期进行一次种群计数运算，这会极大影响二进制计算 [参考资料 18]。

如图 2 所示，为了与机器学习推断空间的发展保持同步，GPU 厂商一直进行必要的芯片修改，以支持有限的几种降精度数据类型，例如 FP16 和 INT8。例如，Tesla P4 和 P40 卡上的 NVidia GPU 支持 INT8，每 ALU/Cuda 内核提供 4 个 INT8 运算。

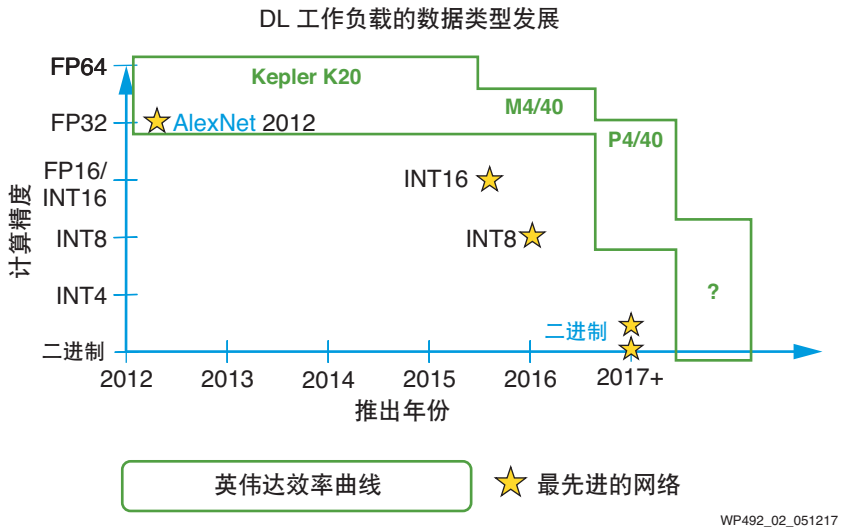

图 2： **英伟达降精度支持**

然而，英伟达面向 Tesla P40 上的 GoogLeNet v1 Inference 发布的机器学习推断基准结果表明，INT8 方案与 FP32 方案相比效率只提升 3 倍，说明要在 GPU 架构中强行降低精度并取得高效结果存在较大难度 [参考资料 16]。

随着机器学习和其他工作负载转向更低精度和定制精度，GPU 厂商需要向市场推出更多新产品，他们的现有用户也需要升级平台才能受益于这种改进。

通过严格的存储器层级实现软件定义数据路径

与 CPU 类似，GPU 中的数据流也由软件定义，并取决于 GPU 的严格而复杂的存储器层级 [参考资料 17]。典型的 GPU 存储器层级如图 3 所示。

每个线程在寄存器文件中都有自己的存储器空间，用以存储线程的本地变量。少量线程（相同的存储块中）可通过共享存储器通信；所有线程都能通过全局或片外存储器通信 [参考资料 18]。

如图 3 所示，与存储器访问有关的能耗和时延分别增加 100 倍和 80 倍以上，因为数据需要遍历存储器层级——从寄存器文件到全局存储器 [参考资料 15][参考资料 17][参考资料 19]。此外，存储器冲突不可避免，会增大时延，导致 ALU 闲置，致使计算能力和效率降低。

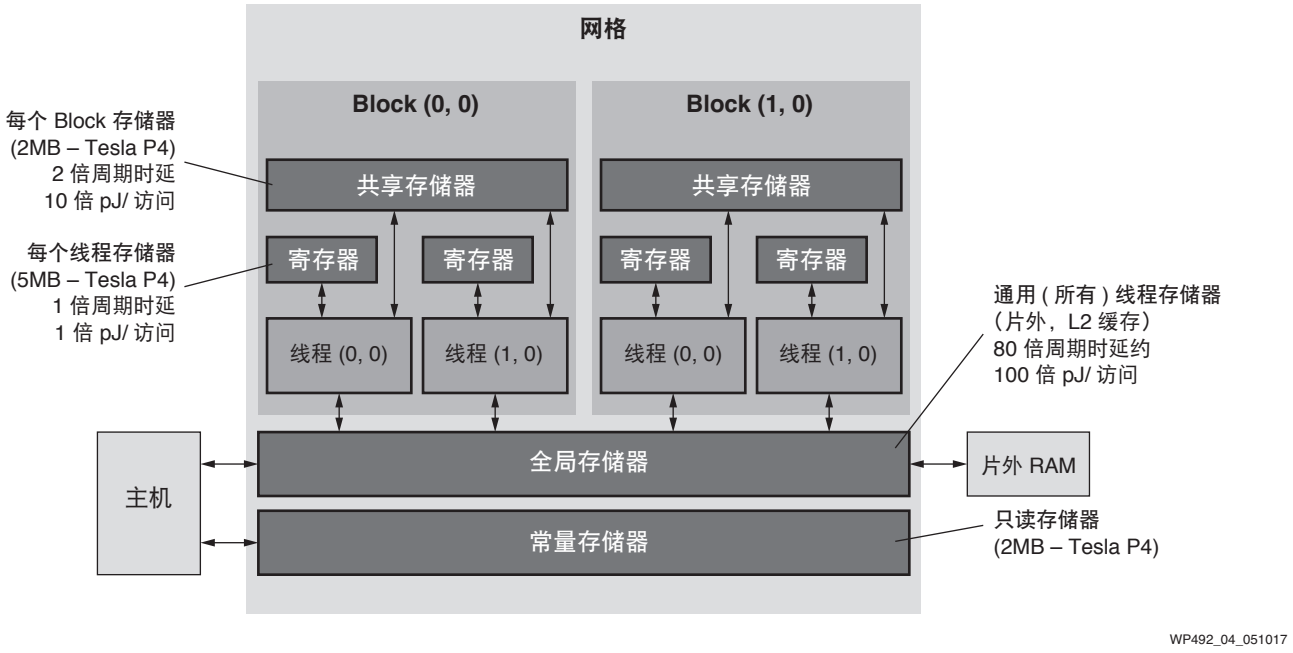

图 3： **典型的 GPU 存储器层级**

因此，如果实现 GPU 的计算和效率潜能，工作负载的数据流必须准确映射到 GPU 存储器层级。工作负载要具备足够的数据局部性，才能高效地映射到 GPU，实际上这样的工作负载很少。对大多数工作负载而言，当在 GPU 上实现时，实际的计算能力和效率会大打折扣，解决方案的时延也会增加 [参考资料 19][参考资料 20]。

机器学习推断作为量化实例，能清楚反应出这种数据流局限性。GPU 必须批处理，例如 128，以实现高效但时延更长的解决方案。最终，批处理使机器学习处理局部化，但代价是时延增加 [参考资料 21]。GoogLeNet v1 Inference 的 NVidia P40 基准测试结果可清楚地反映出该效应。对于 GoogLeNet v1，网络因 P40 存储器带宽而受计算束缚，因此与批处理有关的存储器带宽削减不会产生很大帮助。然而，P40 显然需要 128 批处理以实现 50% 的 GPU 理论性能，会给系统带来很大时延 [参考资料 16]。

有些情况下，可利用 CPU 对数据进行预处理，以便工作负载更好地映射到 GPU SIMT 架构和存储器层级，但代价是产生更多 CPU 计算和功耗，抵消了 GPU 的优势 [参考资料 7]。

有限的 I/O 选项

如“GPU 起源和目标工作负载”部分所述，GPU 的角色是作为协处理器。为了便于与主机通信，GPU 以往只有一个硬 PCIe® 接口以及几个片外 DRAM 接口（例如 GDDR5）。最近几代产品中，有些 GPU 采用硬接口实现 GPU 到 GPU 通信。仍然需要使用 CPU 来与网络进行连接以及向 GPU 分配任务，这会增加系统功耗，同时会因 PCIe 的有限带宽而带来瓶颈问题。例如，英伟达的 Tesla P40 支持 PCIe 3.0 x16，只能实现 16GB/s 带宽。

GPU 厂商已经开始构建小型 SoC，例如 NVidia Tegra X1，能够提供集成 GPU 计算、ARM® 处理器以及一些通用汽车外设（如 HDMI、MIPI、SIP、CAN 和基础以太网）。这些器件只具备少量计算能力，必须依靠额外的分立 GPU 实现必要的计算能力。然而，分立 GPU 的接口有很大局限性，例如 Tegra X1 仅支持 PCIe 2.0 x4，造成严重瓶颈。额外的 SoC 的功耗会进一步降低平台的效率。

片上存储器资源

除了时延、效率和吞吐量方面的不利影响，片外存储器的带宽要显著低于本地 / 片上存储器。因此，如果工作负载需要依靠片外存储器，不仅片外存储器的带宽会成为瓶颈，而且计算资源也会被闲置，从而降低 GPU 提供的计算功能和效率。

因此，更有利的做法是采用大型低时迟、高带宽片上存储器。再次以机器学习推断为例，GoogLeNet 共需要 27.2MB 的存储器，假设 FP32 方案，这样没能提供 GPU，这意味着需要片外存储器 [参考资料 22]。很多情况下需采用高昂的高带宽存储器 (HBM) 和批处理，以防止内核闲置。如果选择具有更大型片上存储器的器件，就能避免 HBM 成本以及额外的时延和功耗问题。

功耗范围

GPU 厂商在设计板卡和 GPU 时通常要适应 250W 功耗上限，并依靠有效热管理来调节温度。针对机器学习推断市场，英伟达开发了满足 75W 功耗范围的器件，例如 Tesla M4 和 P4。即使 75W 也远超出所允许的系统级功耗和热范围。GPU 的绝对功耗依然是阻碍 GPU 广泛使用的一大因素。

功能安全性

GPU 源自消费图形处理和高性能计算领域，不存在功能安全性要求。随着 GPU 厂商瞄准 ADAS 市场，功能安全性就变成了优先考虑和要求。器件需要全新设计，以确保实现所需的功能安全性认证等级，以便用在 ADAS 系统中。对 GPU 厂商来说这是一个长期学习过程，涉及各个方面，需要新的工具和设备。

赛灵思 FPGA 的起源

1984 年，赛灵思发明了现场可编程门阵列 (FPGA)，使用户能够在单个器件中编程（重编程）几乎无限数量的功能。以前，系统设计人员使用很多通用的分立逻辑组件或通过构建高成本的 ASIC 来实现这些功能 [参考资料 23]。

三十多年来，灵活性和可编程性仍然是赛灵思 All Programmable FPGA 和 SoC 的支柱。赛灵思提供的可编程平台能满足有线与无线通信、云计算、医疗、汽车、工业以及航空航天与国防领域中多种终端应用的核心需求。所有这些应用都需要强大的计算能力，很多还有非常严格的实时要求，例如工业自动化和 ADAS。

通常，FPGA 在使用上的挑战之一是需要利用硬件描述语言 (HDL)（例如 Verilog 或 VHDL）对其进行编程。最近，赛灵思开发出了 SDSoC ™ 和 SDAccel ™ 工具，能够将可编程器件的诸多优势提供给更广泛的用户（例如软件开发人员和系统架构师）[参考资料 24] ；并且构建了更多加速堆栈，使系统设计人员能更快速地实现赛灵思器件的优势 [参考资料 25]。

赛灵思 FPGA 和 SoC 的独特优势

原始计算能力

与 GPU 拥护者的说法不同，单个赛灵思器件能提供强大的原始计算能力，例如 Virtex® UltraScale+ ™ XCVU13P FPGA 的性能达到 38.3 INT8 TOP/s。最先进的 NVidia Tesla P40 加速卡以基础频率运行时提供相似的 40 INT8 TOP/s 原始计算能力，但功耗是赛灵思解决方案的 2 倍多 [参考资料 26]。赛灵思器件的灵活性和片上存储器能针对很多工作负载和应用显著提高计算能力（详见 All Programmable 器件灵活性和片上存储器资源）。

此外，赛灵思器件的灵活性意味着能够支持各种数据类型精度，例如 FP32、INT8、二进制和定制 [参考资料 27]。例如，针对二值化神经网络，赛灵思提供 500TOPs/s 的超高二进制计算能力（假设 2.5 LUT/ 运算），相当于 GPU 典型性能的 25 倍。有些精度最适合使用 DSP 资源，有些最适合在可编程逻辑中实现，还有些适合将二者结合起来使用。这种灵活性确保器件的计算和效率随着精度降低而调整，一直到二进制运算。

机器学习领域的大量研究都从计算、精度和效率角度来研究最佳精度 [参考资料 28][参考资料 29][参考资料 30][参考资料 31][参考资料 32]。无论最佳点在哪，对于给定工作负载，赛灵思器件的计算能力和效率都能随之调整，以实现降低精度后的所有优势。

几年来，很多 FPGA 用户实现了脉动阵列处理设计，以便针对多种工作负载实现最佳性能，包括机器学习推断。[参考资料 33][参考资料 34]。为了确保赛灵思 FPGA 和 SoC 用户能够在现有的赛灵思器件上针对此类工作负载将可实现的计算能力和效率实现最大化，赛灵思为此提供多种资源。这些资源包括 INT8 最优化以及将 DSP 阵列映射到 block RAM 和 UltraRAM 的最高效存储器层级 [参考资料 27]。如需了解有关这些资源的更多信息，敬请联系您所在地的赛灵思销售代表。

为了针对当今的深度学习工作负载提高可用的计算能力和效率，英伟达在 Volta 架构中以 Tensor Core 的形式硬化了类似功能。然而，深度学习工作负载会随时间演进，因此 Tensor Core 架构也可能需要改变，而且 GPU 用户需要等待和购买新的 GPU 硬件。

效率和功耗

从系统级角度看，计算平台必须在给定的功率和热范围之内提供最大计算能力。为满足这一需求，计算平台需要：

- 处于允许的功率范围内
- 能够在功率预算内将计算能力最大化

赛灵思提供丰富的 All Programmable 器件，这使用户能选择与功率和热范围最匹配的器件。此外，赛灵思的 UltraScale+ 器件具有低压模式 (VLOW)，能将功耗降低 30%，效率提升 20%。

如表 1 所示，赛灵思器件针对固定精度数据类型提供从原始计算角度看最高效的通用计算平台。这主要是因为赛灵思 FPGA 架构中的处理开销更低。例如，GPU 需要围绕计算资源实现更多复杂性，以便实现软件可编程功能。对于当今的深度学习工作负载的张量运算，英伟达的 Tesla V100 凭借硬化的 Tensor Core 能实现与赛灵思 FPGA 和 SoC 差不多的效率。然而，深度学习工作负载也在快节奏演进，因此无法确定英伟达的 Tensor Core 能够针对深度学习工作负载保持多久的高效性。显然对于其他通用工作负载，NVidia V100 也存在效率方面的挑战。

表 1：器件效率假设 90% 器件利用率和 80% 有效时钟周期 (1)

器件	通用计算效率	张量运算效率
NVidia Tesla P4	209 GOP/s/W(2) (INT8)	
NVidia Tesla P40	188 GOP/s/W (INT8)	
NVidia Tesla V100	72 GFLOP/s/W(3) (FP16)	288 GFLOP/s/W (FP16)
Intel Stratix 10	136 GOP/s/W (INT8)	
赛灵思 Virtex® UltraScale+ ™	277 GOP/s/W (INT8) [参考资料 27]	

注：
1. 引用数字仅用于比较。可实现的器件效率取决于最终应用和用户。
2. 每瓦特功耗每秒十亿次运算。
3. 每瓦特功耗每秒十亿次浮点运算。

鉴于本白皮书之前介绍的局限性，对于真实的工作负载与系统，GPU 很难接近表 1 中所给出的数字。

赛灵思器件的灵活性及其他优势，加之赛灵思最新软件开发堆栈，能确保基于赛灵思的解决方案为大量最终应用和工作负载实现显著提高的效率。

赛灵思器件的附加优势——例如灵活性和片上存储器——能确保赛灵思器件的效率被大量最终应用和工作负载所实现。

All Programmable 器件的灵活性

赛灵思器件经过精心设计，能满足多种高性能终端系统的计算、效率、成本和灵活性需求。赛灵思将硬件可编程资源（例如逻辑、走线和 I/O）与灵活、独立的集成内核模块（例如 DSP Slice 和 UltraRAM）组合在一起，全部构建在领先的工艺技术上，例如台积电 (TSMC) 的 16nm FinFET 工艺技术，从而达到这种平衡。

赛灵思器件具备硬件可编程性和灵活性，意味着底层硬件通过配置可满足给定工作负载的需求。随后，甚至在运行时也可通过部分重配置功能方便地重新配置数据路径 [参考资料 35]。图 4 试图捕捉赛灵思 All Programmable 器件提供的部分灵活性，但是赛灵思器件的真正灵活性无法通过单张图片来体现。内核（或用户设计元素）可以直接连接可编程 I/O、任意其它内核、LUTRAM、block RAM 和 UltraRAM、外部存储器等。

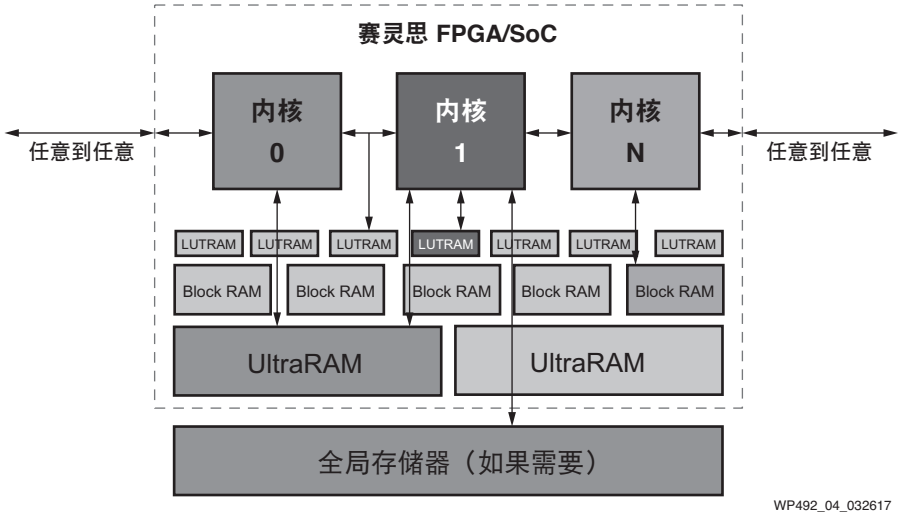

图 4： **All Programmable 数据路径和任意到任意 I/O**

赛灵思器件具有独特的硬件可编程性，意味着它们不存在特定局限性，例如 SIMT 或固定数据路径。无论是大规模并行、适度并行、流水线连续或者混合形式，都能获得赛灵思器件的计算能力和效率。此外，如果底层算法改变（例如机器学习网络的发展），则平台也能相应调整。

很多系统和工作负载中都能看到赛灵思器件发挥灵活性优势。其中一种工作负载就是机器学习推断。机器学习推断的趋势之一是向稀疏网络过渡。赛灵思器件的用户已经在利用这种趋势。英伟达公司本身就是这些用户中的一个。在最近与英伟达联合编写的关于语音辨识的一篇文章中，通过使用赛灵思 FPGA，相对 CPU 实现了 43 倍速度提升和 40 倍效率提升，相对 NVidia GPU 实现了 3 倍速度提升和 11.5 倍效率提升 [参考资料 36]。可编程数据路径还减少了赛灵思 FPGA 批处理需求。批处理是系统的时延 vs 实时性能的重要决定因素。

从大数据角度看，赛灵思器件的灵活性也展现出明显优势。赛灵思 FPGA 在处理 SQL 工作负载时非常高效和快速，包括具有复杂数据（例如可变长度字符串）的情况。百度利用基于赛灵思 Kintex® UltraScale ™ KU115 器件的加速卡实现了 25 倍以上的提速。该加速卡的功耗仅 50W。百度该解决方案的效率比 GPU 方案快 4 倍 [参考资料 37]。对于文本和图案匹配工作负载，研究表明基于赛灵思的 RegEx 方案比 CPU 方案快 14.5–18 倍，比 GPU 方案快了将近 3 倍 [参考资料 38][参考资料 39]。

基因组分析是另一个切实的实例。有人已经利用 GPU 来加速基因组分析，可相较于 Intel Xeon CPU 方案提速 6–10 倍 [参考资料 40]。不过，赛灵思 FPGA 的提速效果要高得多，相较于同等 CPU 可提速 80 倍 [参考资料 41]。

赛灵思器件的灵活性还使其成为云服务提供商的理想选择，可作为计算平台即服务的一部分。多种类型的软件即服务都可以利用赛灵思器件的优势。

最后，对于正在努力研发自动驾驶功能的汽车系统设计人员来说，赛灵思器件的灵活性能够为他们提供可扩展的平台，用以满足完全自动驾驶道路上的各种美国汽车工程师学会 (SAE) 标准。如需了解关于 SAE 标准的更多信息，敬请访问 SAE 网站。赛灵思器件可以高效处理来自各种源头的传感器数据，例如雷达、照相机和超声波，同时保持系统的实时 / 时延目标。

任意到任意 I/O 灵活性

除了器件计算资源的灵活性，赛灵思的任意到任意 I/O 灵活性能够确保器件可无缝集成到现有的基础架构，例如直接连接到网络或存储设备，无需使用主机 CPU [参考资料 42]。I/O 灵活性还允许平台针对基础架构的变化或更新进行调整。

如需了解关于赛灵思 UltraScale 架构器件的更多详情，请参阅不断扩大的赛灵思大型白皮书库。

片上存储器

如表 2 所示，赛灵思器件提供业界领先的灵活、高带宽、低时延的 500Mb 片上存储器 [参考资料 44]。如此大的片上存储器缓存意味着工作负载的很大一部分存储器要求都能通过片上存储器来满足，从而减小外部存储器访问带来的存储器瓶颈问题，以及高存储器带宽解决方案（例如 HBM2）的功耗和成本问题。例如，针对大多数深度学习网络技术（例如 GoogLeNet）的系数 / 特性图都可存在片上存储器中，以提高计算效率和降低成本。

表 2： **器件片上存储器大小**

器件	片上存储器 (Mb)
NVidia Tesla P4	80
NVidia Tesla P40	101
NVidia Tesla P100	144
NVidia Tesla V100	300
Intel Stratix 10	244
赛灵思 Virtex® UltraScale+	500

片上存储能消除片外存储器访问引起的巨大时延问题，将系统的实时性能最大化。

封装内的 HBM

针对需要高带宽存储器的情况，赛灵思在部分 Virtex UltraScale+ 器件中提供 HBM。除了封装内 HBM 堆栈的 460GB/s 存储器带宽，赛灵思 HBM 存储器控制器还增加更大的灵活性，以便将工作负载高效映射到器件和可用存储器带宽，将效率和计算效率最大化。

功能安全性

赛灵思长期以来能够满足各种功能安全性，包括工业自动化以及最近的 ADAS。赛灵思工具和器件经过重新设计，以便支持功能安全性应用，并达到相应认证等级 [参考资料 45]。

因此，多家汽车制造商在安全关键型 ADAS 应用中采用 Zynq®-7000 All Programmable SoC 量产器件。Zynq UltraScale+ MPSoC 还进一步扩大对功能安全应用的支持。

结论

系统设计人员在这个新的计算时代面对不同选择。赛灵思 FPGA 和 SoC 为系统设计人员提供最低风险，帮助其满足未来系统的核心要求与挑战，同时提供足够的灵活性以确保平台在未来不会落伍。

在深度学习领域，UltraScale 架构中的 DSP 架构内在的并行性能够针对具有可伸缩 INT8 向量点积性能的神经网络加强卷积和矩阵乘法计算量。这能为深度学习推断实现更低时延。快速 DSP 阵列、最高效的 block RAM 存储器层级以及 UltraRAM 存储器阵列可实现最佳功率效率。

现在，利用链接 https://china.xilinx.com/products/boards-and-kits.html 中的开发套件，以及多种设计输入工具，例如 HLS、SDSoC 和 SDAccel 工具，用户可发挥赛灵思器件的诸多优势。

参考资料

1. ZDNET."Vision and neural nets drive demand for more powerful chips."Accessed April 6, 2017. http://www.zdnet.com/article/vision-and-neural-nets-drive-demand-for-better-chips/.
2. NVidia Corporation. http://www.nvidia.com/object/IO_20020111_5424.html.
3. MJ Misic, DM Durdevic, MV Tomasevic."Evolution and trends in GPU computing" MIPRO, 2012 Proceedings of the 35th International Convention, 289-294. http://ieeexplore.ieee.org/abstract/document/6240658/.
4. Nicole Hemsoth."FPGAs Focal Point for Efficient Neural Network Inference."Last accessed on April 6, 2017. https://www.nextplatform.com/2017/01/26/fpgas-focal-point-efficient-neural-network-inferen ce/.
5. http://www.cse.psu.edu/hpcl/docs/2016_PACT_Onur.pdf.
6. Babak Falsafi, Bill Dally, Desh Singh, Derek Chiou, Joshua J. Yi, Resit Sendag, "FPGAs versus GPUs in Datacenters," IEEE Micro, vol. 37, no. 1, pp. 60-72, Jan 2017. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7866802.
7. Richard Vuduc, Aparna Chandramowlishwaran, Jee Choi, Murat Guney, Aashay Shringarpure. "On the limits of GPU acceleration."HotPar'10 Proceedings of the 2nd USENIX conference on Hot topics in parallelism.Pages 13-13.Berkeley, CA.June 14–15, 2010. http://hpcgarage.org/wp/vuduc2010-hotpar-cpu-v-gpu.pdf
8. Fowers, J., Ovtcharov, K., Strauss, K., Chung, E.S., Stitt, G.:A High Memory Bandwidth FPGA Accelerator for Sparse Matrix-Vector Multiplication.In:IEEE Int. Symp. on Field-Programmable Custom Computing Machines (2014). http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6861585
9. B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky."Sparse Convolutional Neural Networks."Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference. http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Liu_Sparse_Convolution al_Neural_2015_CVPR_paper.pdf.
10. Yaman Umuroglu, et al."Random Access Schemes for Efficient FPGA SpMV Acceleration." Microprocessors & Microsystems Volume 47 Issue PB, November 2016 Pages 321-332. http://www.sciencedirect.com/science/article/pii/S0141933116300023 (figure 4 - shows utilization)
11. Sparsh Mittal, Jeffrey S. Vetter."A Survey of CPU-GPU Heterogeneous Computing Techniques."ACM Computing Surveys (CSUR) Volume 47 Issue 4, July 2015 Article No. 69.
12. Xilinx white paper, "Reduce Power and Cost by Converting from Floating Point to Fixed Point." Last accessed April 6, 2017. https://www.xilinx.com/support/documentation/white_papers/wp491-floating-to-fixed-point.pdf.
13. Marc Baboulin et al."Accelerating Scientific Computations with Mixed Precision Algorithms."Computer Physics Communications 180 (2009) 2526-2533. http://www.netlib.org/utk/people/JackDongarra/PAPERS/202_2009_Accelerating-Scientific-Co mputations-with-Mixed-Precision-Algorithms.pdf.
14. Suyog Gupta et al."Deep Learning with Limited Numerical Precision."ICML'15 Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 Pages 1737-1746. https://arxiv.org/pdf/1502.02551.pdf.
15. William Dally."High-Performance Hardware for Machine Learning."Last accessed April 6, 2017. https://media.nips.cc/Conferences/2015/tutorialslides/Dally-NIPS-Tutorial-2015.pdf.
16. NVidia Corporation."NVIDIA TensorRT."Last accessed April 6, 2017. https://developer.nvidia.com/tensorrt.

17. Xinxin Mei, Xiaowen Chu."Dissecting GPU Memory Hierarchy through Microbenchmarking." IEEE Transactions on Parallel and Distributed Systems Volume:28, Issue:1, Page 72-86, Jan. 1 2017. https://arxiv.org/pdf/1509.02308.pdf.
18. NVidia Corporation."Cuda C Programming Guide" Last accessed on April 6, 2017. https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#memory-hierarchy.
19. Mark Gebhart et al."Unifying Primary Cache, Scratch, and Register File Memories in a Throughput Processor."MICRO-45 Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture, Pages 96-106, Vancouver, B.C., CANADA - December 01–05, 2012. https://research.nvidia.com/sites/default/files/publications/Gebhart_MICRO_2012.pdf.
20. Chris Edwards."Minimize Memory Moves for Greener Data Centers" Last accessed April 6, 2017. http://www.techdesignforums.com/blog/2016/06/08/dac-data-center-acclerators/.
21. Vincent Vanhouckehttps et al."Improving the Speed of Neural Networks on CPUs" Proc.Deep Learning and Unsupervised Feature Learning NIPS Workshop 2011 [online] Available: http://research.google.com/pubs/archive/37631.pdf.
22. Christian Szegedy et al."Going Deeper with Convolutions" Proceedings Computer Vision and Pattern Recognition (CVPR).Pages 1–9, 2015. http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf.
23. Funding Universe."Xilinx, Inc. History" Last accessed April 6, 2017. http://www.fundinguniverse.com/company-histories/xilinx-inc-history/.
24. Xilinx Inc. "Software Zone."Last accessed April 6, 2017. https://www.xilinx.com/products/design-tools/software-zone.html.
25. Xilinx Inc. "Acceleration Zone."Last accessed April 6, 2017. https://www.xilinx.com/products/design-tools/acceleration-zone.html#libraries.
26. ANANDTECH."NVIDIA Announces Tesla P40 & Tesla P4 - Neural Network Inference."Last accessed April 6, 2017.
27. Xilinx white paper, "Deep Learning with INT8 Optimization on Xilinx Devices."Last accessed April 6, 2017. https://www.xilinx.com/support/documentation/white_papers/wp486-deep-learning-int8.pdf.
28. Philipp Gysel et al."Hardware-Oriented Approximation of Convolutional Neural networks."ICLR 2016. https://arxiv.org/pdf/1604.03168v3.pdf.
29. Chenzhuo Zhu et al."Trained ternary quantization."ICLR 2017.https://arxiv.org/pdf/1612.01064.pdf.
30. Yaman Umuroglu et al."FINN:A Framework for Fast, Scalable Binarized Neural Network Inference."25th International Symposium on FPGAs, February 2017. https://arxiv.org/pdf/1612.07119.pdf.
31. Wonyong Sunget et al."Resiliency of Deep Neural Networks under Quantization."ICLR 2016. https://arxiv.org/pdf/1511.06488v3.pdf.
32. Nicholas J. Fraser et al."Scaling Binarized Neural Networks on Reconfigurable Logic."HiPEAC 2017. https://arxiv.org/abs/1701.03400.
33. Cl´ement Farabet, Yann LeCun, Koray Kavukcuoglu, Eugenio Culurciello, Berin Martini, Polina Akselrod, Selcuk Talay.Large-Scale FPGA-based Convolutional Networks. http://yann.lecun.com/exdb/publis/pdf/farabet-suml-11.pdf
34. C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong.Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks.ACM/SIGDA ISFPGA, pages 161–170.ACM, 2015. https://pdfs.semanticscholar.org/2ffc/74bec88d8762a613256589891ff323123e99.pdf

35. Xilinx, Inc. "Partial Reconfiguration in the Vivado Design Suite."Last accessed April 6, 2017. https://www.xilinx.com/products/design-tools/vivado/implementation/partial-reconfiguration.html.
36. Song Han et al."ESE:Efficient Speech Recognition Engine with Sparse LSTM on FPGA."International Symposium on FPGA 2017. https://arxiv.org/pdf/1612.00694.pdf.
37. Jian Ouyang et al."SDA:Software-Defined Accelerator for General-Purpose Big Data Analysis System."Hotchip 2014. http://www.hotchips.org/wp-content/uploads/hc_archives/hc28/HC28.23-Tuesday-Epub/HC28.23.80-Big-Data-Epub/HC28.23.832-SDA-BD-Analysis-JianOuyang-Baidu-v06.pdf.
38. Shreyas G Singapura et al."FPGA Based Accelerator for Pattern Matching in YARA Framework."CENG 2015. http://ceng.usc.edu/techreports/2015/Prasanna%20CENG-2015-05.pdf.
39. Yuichiro Utan et al."A GPGPU Implementation of Approximate String Matching with Regular Expression Operators and Comparison with Its FPGA Implementation."PDPTA 2012. https://pdfs.semanticscholar.org/2667/ac95d36ab63ae6eeb4b352f4c20dc46344c3.pdf.
40. BarraCUDA."The BarraCUDA Project."Last accessed April 6, 2017. http://seqbarracuda.sourceforge.net/index.html.
41. Edico Genome."DRAGEN Genome Pipeline."Last accessed April 6, 2017. http://www.edicogenome.com/wp-content/uploads/2015/02/DRAGEN-Genome-Pipeline.pdf.
42. Microsoft paper, "A Cloud-Scale Acceleration Architecture."Last accessed April 6, 2017. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/10/Cloud-Scale-Acceleration-Architecture.pdf.
43. Xilinx, Inc. "UltraScale Architecture."Last accessed April 6, 2017. https://www.xilinx.com/support/documentation/white_papers/wp470-ultrascale-plus-power-flexibility.pdf.
44. Xilinx white paper, “UltraRAM:Breakthrough Embedded Memory Integration on UltraScale+ Devices.”Last accessed April 6, 2017. https://www.xilinx.com/support/documentation/white_papers/wp477-ultraram.pdf.
45. Xilinx white paper, "Xilinx Reduces Risk and Increases Efficiency for IEC61508 and ISO26262 Certified Safety Applications."Last accessed April 6, 2017. https://www.xilinx.com/support/documentation/white_papers/wp461-functional-safety.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 06 年 13 月	1.0.1	印刷编辑
2017 年 06 年 08 月	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。赛灵思产品并非为故障安全保护目的而设计，也不具备此故障安全保护功能，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。