精读报告

> DONGLE: Direct FPGA-Orchestrated NVMe Storage for HLS

研究背景

近来许多近存储 FPGA 已被证明在处理大数据方面具有很大优势。然而，当前的高级综合（HLS）开发环境不允许 HLS 代码直接访问NVMe 存储。带来了如下困难

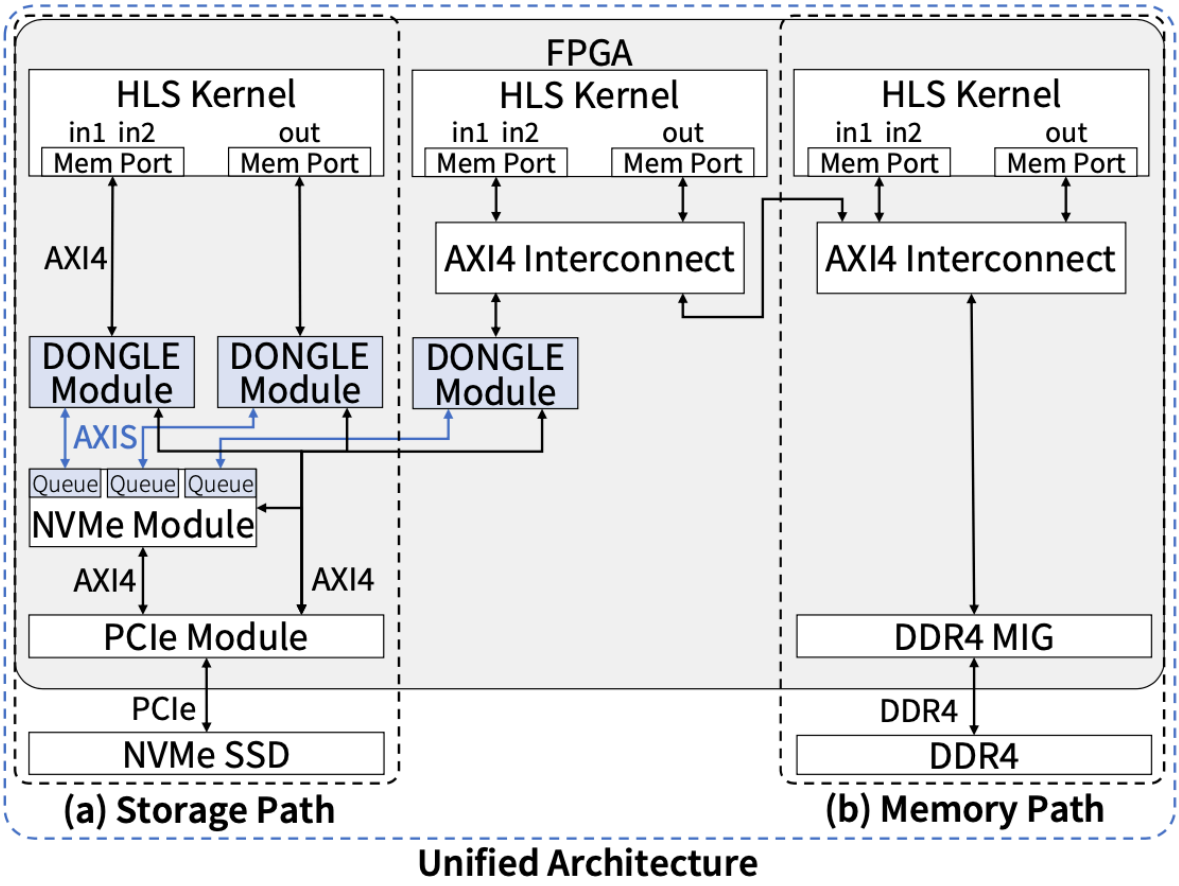
1. 用户必须经常在HLS和主机代码之间切换以访问存储中的数据，并且这个过程需要繁琐的通讯来确保功能正确性。
2. 由于 HLS 代码访问 DRAM 和 NVMe SSD 的访问模式完全不同，针对 DRAM 平台的 HLS 代码移植到基于 NVMe 的平台非常困难，导致代码的可移植性和重用性受限。
3. 频繁地挂起 HLS 内核和 CPU 与 FPGA 之间的同步引入了显著的延迟开销，并需要复杂的调度机制来隐藏延迟。

解决的问题

DONGLE 为 NVMe SSD 和 DRAM 两种不同的存储介质提供了统一的编程接口，允许一个单源 HLS 程序使用相同模式访问多个内存/存储设备，从而使代码库更清晰、可移植和高效。

主要思路

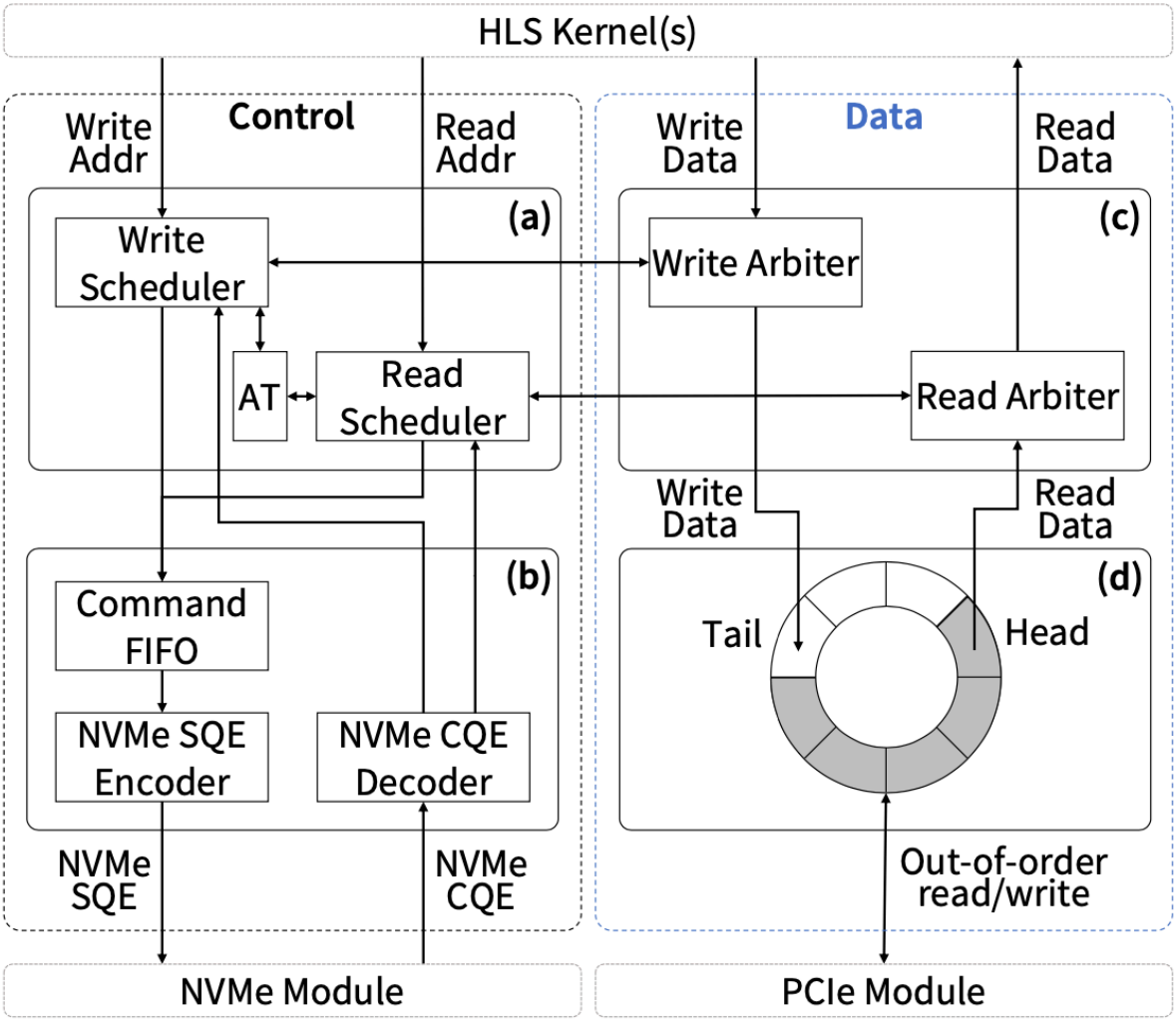
文章中使用了单独的抽象层 DONGLE module，基于块设备 NVMe SSD 提供和 DRAM 一致的基于字节的访存接口。用户可以编写一个单源 HLS 程序，其中只涉及到算法实现，而无需指定底层的内存或存储。然后使用链接指令将 HLS 内存端口分配给内存和存储。DONGLE链接器根据用户的链接指令将 HLS 内核集成到 DONGLE 存储体系结构中，以完成整体的FPGA设计，再使用 Vivado 进行布局和布线，生成用于编程FPGA的比特流。用户在高层次上对底层存储介质没有感知，并且工作流程和现有的 HLS 综合完全一致，没有任何干扰。



该架构可以与现有的 HLS 基础设施相结合，创建一个可以同时访问 NVMe 和 DRAM 的统一架构。因为 DONGLE module 独立于 DRAM 运行，所以可以在不冲突的情况下同时使用 NVMe 和 DRAM。多个 HLS 核心可以运行在同一个 FPGA 上，以同时利用 NVMe 和 DRAM。现有的 HLS 基础设施，如 AXI4 互联，可用于将 HLS 核心连接到 NVMe 和 DRAM，使用单个内存端口同时访问两者。此外，HLS 内存映射变量（例如数组）的位置可以在运行时配置，提高了灵活性，可以实现实现更大的容量或更好的性能。

用户也可以在单个 NVMe SSD 上使用多个 DONGLE module，以利用 NVMe 天生的并发访问特性。这样，单个 NVMe SSD 可以连接到多个内存端口，而无需 AXI4 互联。通过使用多个 DONGLE module 和 NVMe I/O 队列，不同 HLS 端口的访问模式对于 NVMe SSD 来说是透明的，从而提供更高的顺序性能。用户还可以选择将多个变量映射到单个 DONGLE module 和 NVMe I/O 队列，使用单个 HLS 内存端口或 AXI4 互联来节省资源。

为了桥接底层块设备接口与上层基于字节的访存接口，DONGLE module 使用了 Schedule unit、Command unit、Arbitration unit 和 Reorder buffer 四大组件。



- Schedule unit 从 HLS 内核接收访存地址，并调度 Command unit 和 Arbitration unit 的执行，确保操作的正确顺序以保证数据完整性。为了跟踪不同的事务，Schedule unit 将访存地址存储在地址表 (Address Table, AT) 中，并使用对应行号作为调度ID。调度ID被发送给 Command unit 以构造 NVMe 命令。当命令完成返回时，Command unit 会向 Schedule unit 回复ID，Schedule unit 据此操纵 Arbitration unit 返回数据。
- Command unit 负责处理 NVMe 特定的逻辑。它构造 NVMe 命令，将其发送到 NVMe 模块，并解码从 NVMe 模块接收到的 NVMe 回复。当被 Schedule unit 调用时，Command unit 将根据操作类型、块地址、调度ID和传输大小构造一个 NVMe 命令。为了处理 Backpressure 的情况，Command unit 使用一个 FIFO 队列缓存命令，直到 NVMe 模块准备好。当 NVMe 命令完成后，Command unit 通知 Schedule unit 对应的调度 ID。
- Arbitration unit 控制 HLS 内核与 Reorder buffer 之间的数据传输。当被 Schdule unit 调用时，Arbitration unit 决定访存的粒度，并在访存完成后向 Schedule unit 发出信号以安排进一步的操作。
- Reorder buffer 用作 NVMe 传输的 DMA 缓冲区。此外，它还在 HLS 和 NVMe 之间进行数据重排序。其实现为环形缓冲区，向PCIe模块暴露一个内存地址和数据接口进行 DMA，并在内部向 Arbitration unit 暴露另一个数据接口，以将数据传递给 HLS 内核。

缺点

DONGLE module 的性能依赖于 FPGA 平台提供的可配置内存接口的实现，这里以 AMD 为例，它提供的是 AXI4 接口，此接口在非性能导向的低功耗 FPGA 开发板上无法提供 DONGLE 所需的并发访存以及其他特性，泛用性有待提升。

启发

这是 FPGA 加速与大数据处理的交叉领域中相当杰出的工作，通过统一抽象层解决了长久以来不同存储介质访存模式大相径庭带来的困难。硬件加速器能够以最少的编程工作直接访问海量 SSD 存储，而无需通过主机 CPU，从而释放了近存储计算的潜力。 统一抽象层的概念可以应用到许多其他领域，高层次抽象为开发人员提供统一接口，低层次适配带来良好的可移植性和代码复用性。