

看呀， 没有主板！



设计团队如何采用
赛灵思 FPGA 实现
带SATA 的单板电脑

许多 ISM 应用都采用快速可靠的大容量存储器来保存数据采集结果，比方说固态硬盘 (SSD) 就已经成为事实上的应用标准了，因为其具有较高的可靠性，而且速度相当快。SSD 驱动器几乎全部采用串行 ATA (SATA) 接口连接。

我们不妨来看看通过高速 SATA 连接接口添加 SSD RAID 功能来扩展基于赛灵思芯片的单板计算机系统所需步骤。为此，赛灵思联盟计划生态系统结合使用了 ASICS World Service (AS ICS ws) 的高质量 IP 核技术和 Missing Link Electronics (MLE) 的可编程系统设计技术。

不过在讨论项目细节之前，我不妨进一步深入分析一下 SATA 本身。如图 1 所示，完整的 SATA 主机控制器包含多个功能层。因此，为向基于

FPGA 的可编程系统提供完整的 SATA 解决方案，设计人员远不止需要高质量 IP 核。设计方案中的某些方面往往被忽略了。

首先，我们应在 FPGA 硬件中实现物理层 (PHY)、链路层和传输层的某些部分，这就是为什么 IP 厂商在出售的 IP 中会包含这些层。ASICS World Service 提供的 SATA Host IP 核采用所谓的千兆位级收发器 (MGT) 来实现物理层，它在 FPGA 中包括类似于赛灵思应用指南 870 中所述的带外信号模块。高级传输层以及应用、器件和用户程序层最好用软件来实现，因此 IP 厂商一般不会为客户提供这些层，不过这就让系统设计团队肩负起创建上述层的任务，而且会对设计项目带来意料之外的成本。

厂商在 IP 中不包括高级传输层以及应用、器件和用户程序层的原因在于，每种架构各有不同，使用方式也不一样。在此情况下，如果提供完整的解决方案，将 IP 核与用户程序相结合，就必须实施、测试并集成分散/汇聚 DMA (SGDMA) 引擎等组件，这些组件同时包括软/硬件在内。

此外，传输层的通信通过所谓的帧信息结构 (FIS) 进行。SATA 标准定义了 FIS 类型，我们应分析一下主机和器件之间读/写操作过程中详细的 FIS 流。

如图 2 所示，主机通过寄存器 FIS 通知器件新的操作，寄存器 FIS 保存有标准 ATA 命令。就读取 DMA 操作而言，器件一准备好就发送一个（或更多）数据 FIS。器件通过寄存器 FIS 完成从器件到主机的事务处理。该 FIS 可通知操作是否成功。

我们从图 2 中还能看到主机和器件之间用于写入 DMA 操作的 FIS 流程。这里，主机还是通过寄存器 FIS 通知器件操作。器件一旦准备好接收数据，器件就发送 DMA 激活 FIS，主机则开始发送单个数据

作者: Lorenz Kolb
科技委员会成员
Missing Link Electronics 公司
lorenz@missinglinkelectronics.com

Endric Schubert
联合创始人
Missing Link Electronics 公司
endric@missinglinkelectronics.com

Rudolf Usselmann
创始人
ASICS World Service 有限公司
rudi@asics.ws

工业、科学和医疗 (ISM) 应用领域的嵌入式系统必须支持多种接口。正因如此，许多设计团队都选择将 FPGA 子卡直接插入 PC 主板，从而获得一些特殊的排一行接口。考虑到新型 FPGA 的功能非常强大，整个片上系统都能在赛灵思器件中实现。片上系统包括硬件、操作系统和软件，几乎可提供全部 PC 功能，避免了对 PC 主板的需求，从而实现更小型化、更节能、可配置的单板计算机系统。

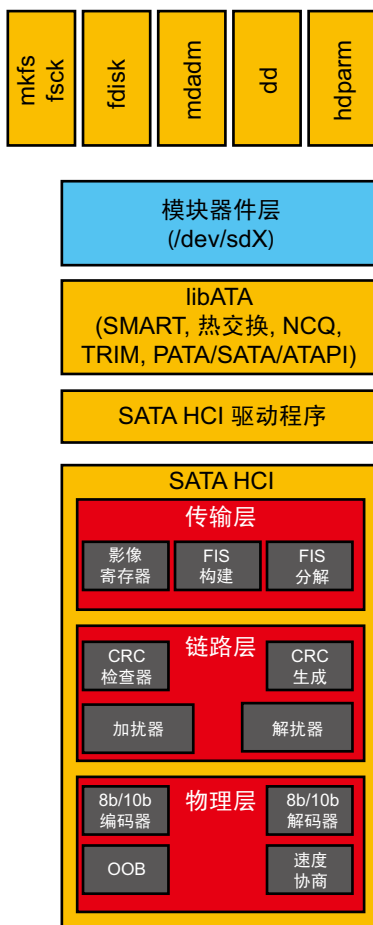


图 1 串行 ATA 功能层

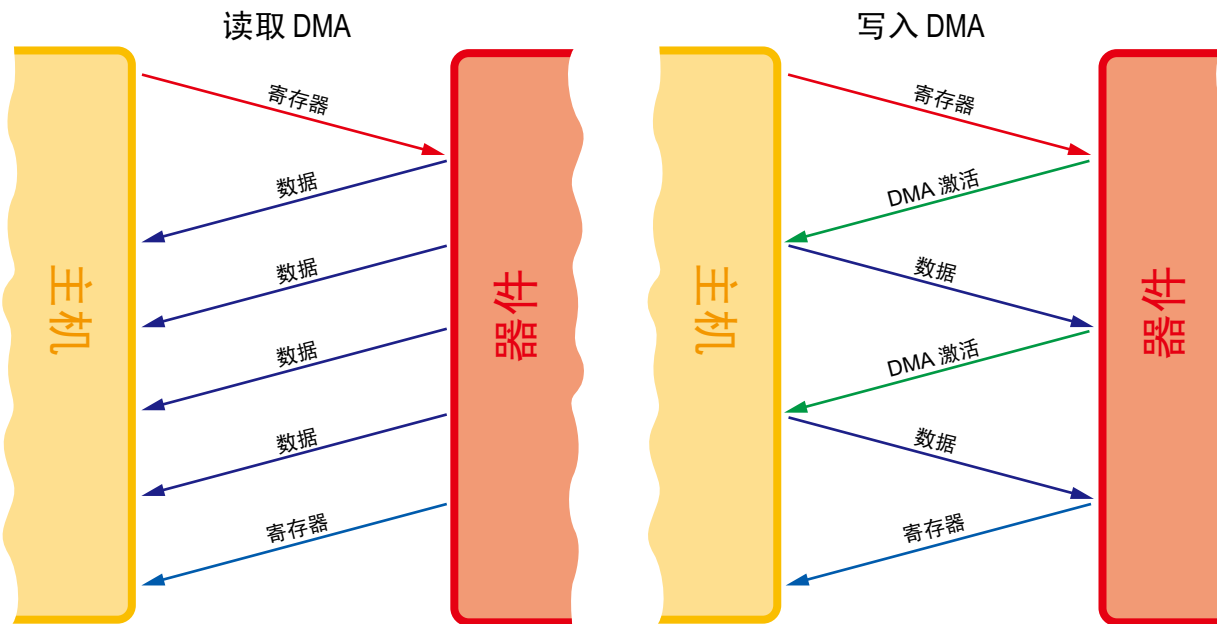


图2 主机和器件之间DMA读写操作的 FIS 流程

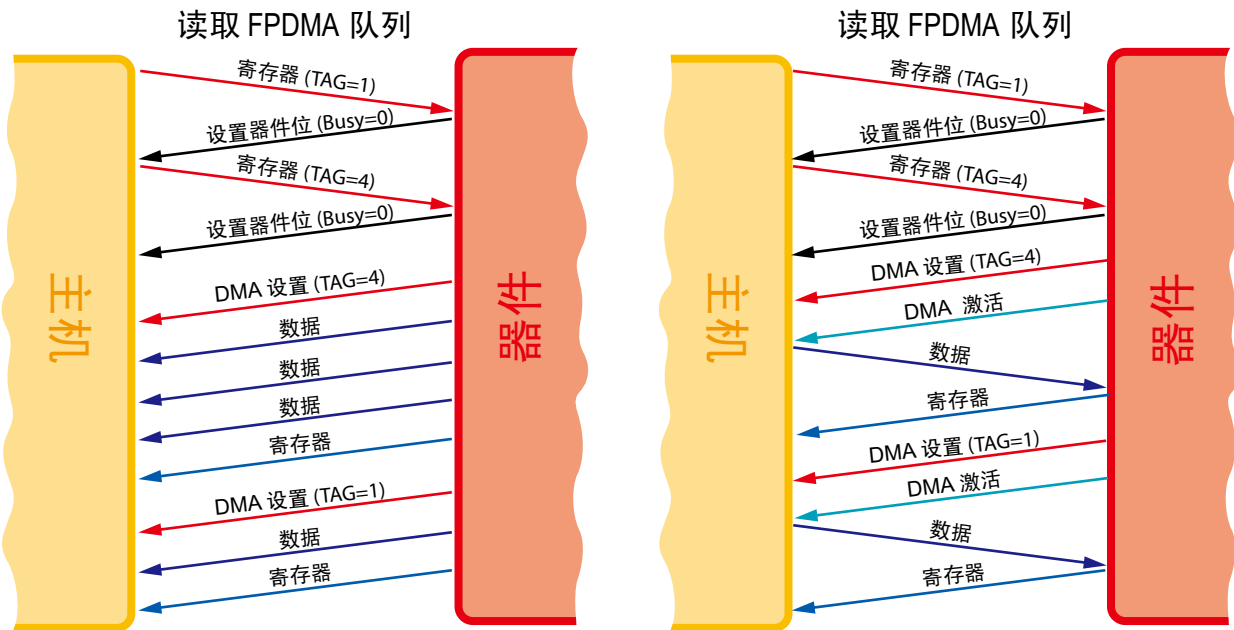


图3 主机和器件之间第一方DMA读写排序操作的 FIS 流程

FIS。器件处理 FIS 后，将等待数据，并发送 DMA 激活 FIS。上述过程的完成方式与读取 DMA 操作一样。

SATA 有一种并行 ATA 不具备的特性，就是所谓的第一方 DMA。该特性将部分 DMA 引擎控制转移到器件上。

这样，器件可缓存命令表，对其重新排序，从而优化性能，这就是所谓的原生命令队列特性。新的 ATA 命令用于第一方 DMA 传输。由于器件不必即时完成命令，而是对其进行排序，因此该操作模式下的 FIS 流程略有不同。读取第一方 DMA

命令队列的流程请见图 3 左侧。

同时，应用层上的通信采用 ATA 命令。您固然能在 FPGA 硬件中实现有限数量的命令作为有限状态机，但用软件实现要高效和灵活得多。这里的开源 Linux 内核提供已知良好的实现方案，几

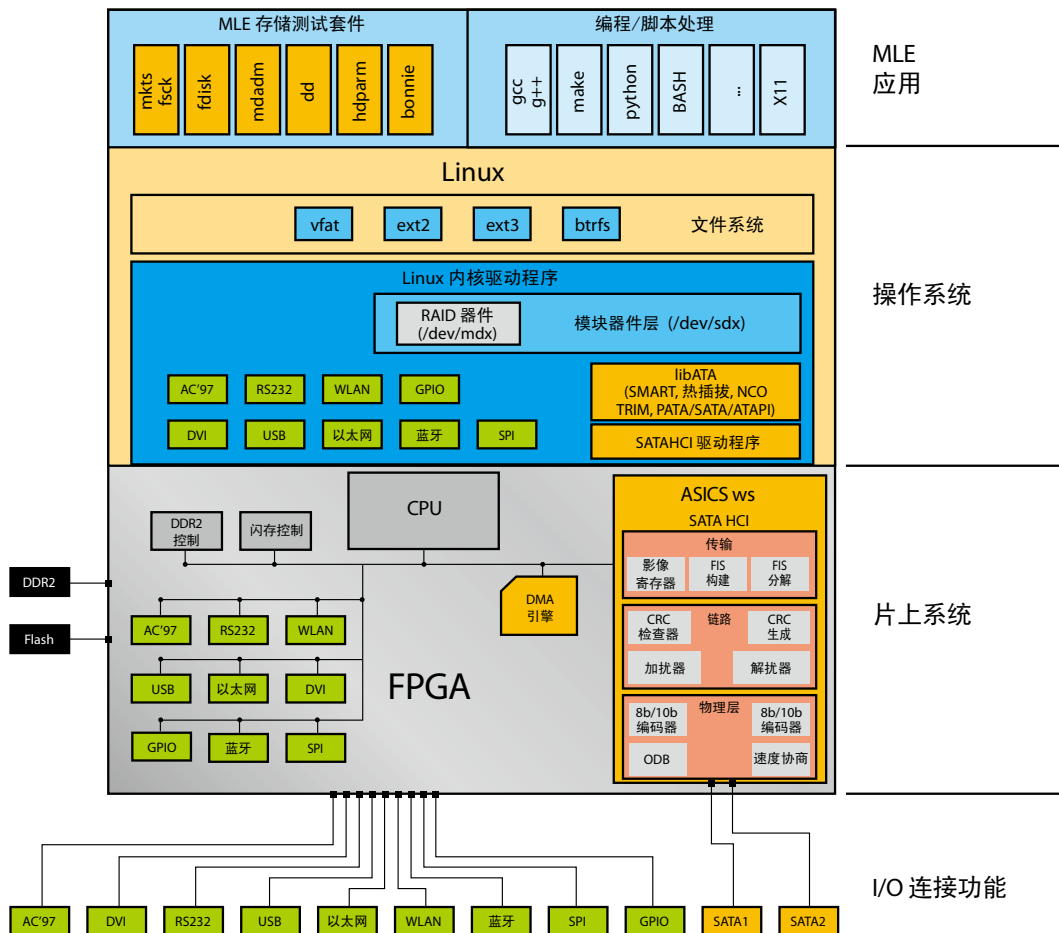


图4 完整 SATA解决方案

乎能完全符合 ATA 标准要求，同时，采用该内核的器件出货量已经超过10亿颗。

Linux ATA 库 (libATA) 能处理 100 多个不同的 ATA 命令，与 ATA 器件通信。这些命令包括数据传输，也提供自监控、分析与报告技术 (SMART) 功能，以及安全擦除和器件锁定等安全特性。

不过，要想利用该代码库，就需要另外再实现独立于硬件的软件，形式为 Linux 器件驱动程序，也就是所谓的 Linux 内核模块。如图 4 所示，Missing Link Electronics 的“软”硬件平台预装了全面的 GNU/Linux 软件协议栈，而且为 ASICS World Service 提供的 SATA 主机 IP 核提供了经测试和优化的器件驱动程序。

我们可以在 FPGA 系统中自由集成

SATA IP 核。要想实现终极整体系统性能，不仅要了解软件或硬件知识，而要全面了解软/硬件知识。软/硬件集成应结合进行。

图 5 给出了如何实现 SATA IP 核系统集成的示例。最明显的办法就是将 IP 核作为从器件添加到总线上（如 A 项），让 CPU 执行存储器和 IP 之间的数据传输。可以肯定的是，数据将通过系统总线传输两次。如果数据速率要求不高的话，这种易于实现的方法或许就足够了。不过，这时我们只能用 CPU 支持较小的应用层，因为 CPU 大部分时间都忙于数据拷贝工作了。

这时如果操作系统需要 CPU 全负荷运行时，我们就会发现性能影响变得

极大。在此情况下，我们或许要考虑添加专用拷贝引擎——赛灵思 Central DMA（也就是图中的 B 项），从而减轻 CPU 的负载。这样，我们仍能通过总线传输两次数据，不过 CPU 不会花大部分时间拷贝数据了。

不过即便如此，采用性能全开系统的性能仍远远达不到独立应用系统的水平，离理论最高值相差甚远。第三种架构选择（也就是图中的 C 项）可通过赛灵思的串流 NPI 端口和多端口存储器控制器 (MPMC) 采用简单的专用拷贝引擎降低系统总线的负载，从而改变这种局面。这样我们就能将独立应用的性能提升到理论上限的水平了。不过，这种系统的 Linux 性能仍有一定的局限性。

我们可以在 FPGA 系统中自由集成 SATA IP 核。要想实现终极整体系统性能，不是仅仅只要了解软件或硬件知识，而是要全面了解软/硬件知识。软/硬件集成应结合进行。

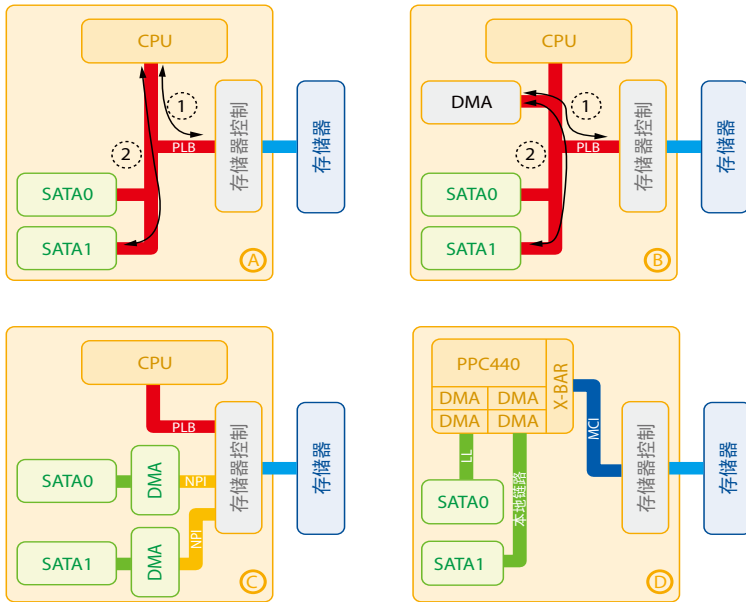


图 5 集成 SATA IP 核的 4 种架构选择

我们从独立应用角度来看，瓶颈不在于内部互联，而在于Linux 存储器管理机制。Linux采用版面尺寸处理存储器。典型系统的版面大小为 4,096 字节。就简单的 DMA 引擎和分散在 RAM 上的 4,096 字节块空闲存储器而言，每次传输只能传输 4,096 字节。而最后一种架构选择（图中的 D 项）能解决这一问题。

举例来说，Virtex®-5 FXT FPGA中采用的 PowerPC PPC440 具有支持 SGDMA 的专用引擎。这样 DMA 引擎得到存储器记录指针列表，并在表间分散/汇聚数据，从而扩大传输字节，让系统性能非常接近独立应用系统的性能。图 6 总结了不同架构的性能水平。

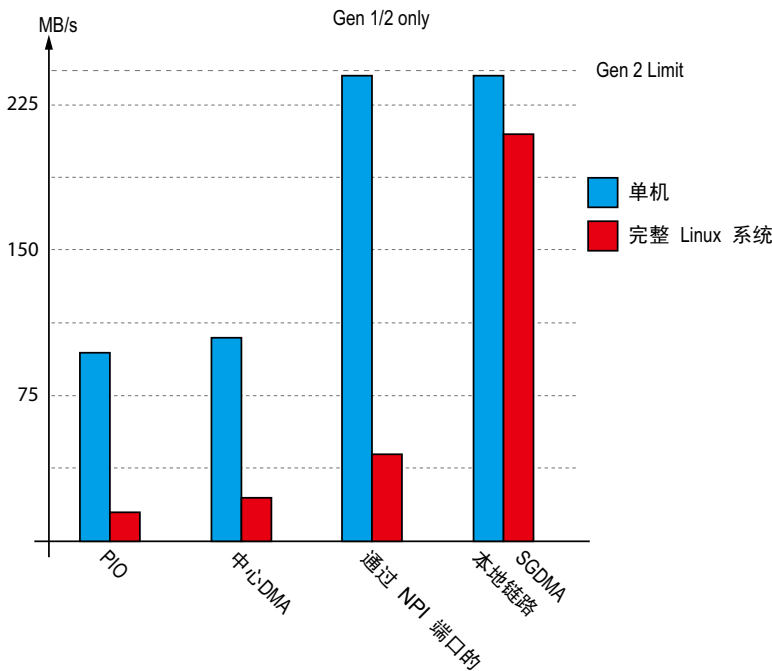


图 6 完整 SATA 解决方案的性能

如今，到底是自行设计还是购买 SATA 主机控制器的决定已经很明显了。很少有设计团队能在可接受的成本实现有效的 SATA 主机控制器。同时，设计团队要花大量时间和资金将内核集成到可编程片上系统上，为内核开发设备驱动程序，并为 IP 核运行（和测试）开发应用软件。

若没有赛灵思联盟计划合作伙伴 ASICS World Service 公司和 Missing Link Electronics 公司的积极支持，我们团队就难以联合推出解决方案。如欲进一步了解全面的 SATA 解决方案信息，欢迎访问 MLE 实时在线评估网站：<http://www.issinglinkelectronics.com/LOE>。您在此将进一步了解相关技术信息，并能应邀参加完整 SATA 系统的网上试用活动。