

数字处理系统中基于 DMA 的数据存储的 FPGA 设计

盛 磊¹
SHENG Lei

摘 要

针对数字处理系统在高吞吐量场景下数据存储效率低、网络传输与本地存储协同性差的问题,文章提出了一种基于 AXI-DMA 技术的结合 SD2.0 的 TF 卡存储,并结合 UDP/TCP 以太网传输的 FPGA 设计方案。首先,深入分析 AXI-DMA 协议的传输机制与数据交互细节,明确其在高速数据搬运中的优势;其次,结合 SD2.0 协议下 TF 卡的存储容量、读写速率特性,以及 UDP/TCP 协议在网络传输中的实时性、可靠性差异,构建多模块协同的数据流架构;在此基础上,优化数据路径设计,实现“本地高速缓存-实时网络传输”的无缝衔接,解决数据在存储与传输环节的瓶颈问题。实验与验证结果表明,该设计有效优化了系统数据管理机制,提升了数据处理效率与传输安全性,具有重要的实际应用价值。

关键词

AXI-DMA; TF 卡存储; UDP 以太网; FPGA 设计; 高速数据传输

doi: 10.3969/j.issn.1672-9528.2025.09.032

0 引言

在具有 AD/DA 或者射频收发器 SOC 的信号处理系统与设备中,常存在需要将某些收集到的连续状态信息或者数据导出存储的情况;反之也存在需要将连续数据或者信号信息导入系统的情形^[1]。结合当前硬件技术发展趋势与设计指导思想,借鉴主流硬件平台化设计方案,采用以 FMQL/ZYNQ 复合 SOC 为主控芯片的架构,基于 DMA 技术搭建并实现了一套复合型数据存储系统。

FMQL 平台是一款与进口 Xilinx ZYNQ 平台功能相似的完全国产化综合型 SOC 平台,其核心架构融合了 FPGA 与 ARM 处理器。该平台既具备 FPGA 可编程逻辑带来的灵活芯片设计能力,又拥有 ARM 处理器支持的便捷、高效软件迭代开发优势,实现了硬件可编程性与软件开发灵活性的有机结合。

本文针对某型号雷达信号处理系统的数据存储需求,提出了一种基于 AXI-DMA 和 TF 卡存储以及 UDP 以太网传输的 FPGA 数据存储方案。该系统采用复旦微公司的 FMQL (ZYNQ) 器件作为核心处理器,其中可编程逻辑部分(PL)采用基于 AXI 总线的 DMA IP 核与 TF 卡控制器 IP 核实现数据与 TF 卡之间的高速传输^[2],可编程软件部分(PS)采用嵌入式 FreeRTOS 操作系统,通过挂载 TF 卡,并利用 UDP 协议与上位机进行数据交互。简而言之,充分利用了 FPGA 的并行处理能力和 AXI 总线的高带宽特性,同时采用 TF 卡

作为存储介质,有效解决了数据存储容量不足的问题。此外,系统将数据通过以太网传输至上位机,便于后续的数据分析和处理。

1 关键系统技术设计

1.1 数字系统架构设计

系统架构如图 1 所示。

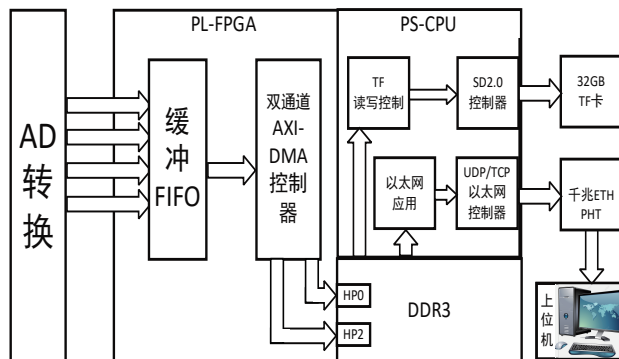


图 1 基于 SD2.0 和 UDP 以太网的数据存储 FPGA 系统架构图

本设计中,基于 DMA 的数据存储 FPGA 模块的数据来源以架构中的 AD 转换芯片为例:该芯片首先对高频模拟信号进行数字下变频与模数转换处理,将其转换为零中频数字信号;所得数字信号随后进入一组缓冲 FIFO,完成 FPGA 内部的时钟域匹配与位宽转换;之后,通过双通道 AXI-DMA 子系统,数据被送入 CPU 子系统的高速数据接口(HP0 与 HP2 端口),并借助两组独立的高性能 AXI 端口,以 PL 端(FPGA 控制器)的操作界面直接写入 PS 端(处理器系统)

1. 贵州航天电子科技有限公司 贵州贵阳 550009

的 DDR3 大容量存储空间。当数据写入 DDR3 后, PL 端会向 CPU 发送指示信号; CPU 接收到信号后, 通过软件控制流程调用子程序, 将 DDR3 中缓存的采样数据分两路处理: 一路转存至 SD2.0 协议的 TF 存储卡中实现本地存储, 另一路通过 UDP 协议传输至上位机完成数据上传。

关于本设计的具体细节, 下文将分为“DMA 传输系统设计”与“以太网/TF 卡数据传输系统设计”两部分展开详细讨论。

1.2 DMA 传输系统设计

直接存储器访问(direct memory access, DMA)是一种高效的内存访问和数据传输技术, 在无需 CPU 干预的情况下, 实现外设与内存之间的独立双向数据传输^[3]。在 FPGA 系统设计中, DMA 控制器通常集成于片上系统的互连架构, 如 AMBA AXI 总线。Xilinx FPGA 器件提供了多种现成的 DMA IP 核, 如 AXI DMA、AXI CDMA、AXI VDMA 等, 可满足不同应用场景下的数据传输需求。

系统中所设计的 DMA 传输架构如图 2 所示。

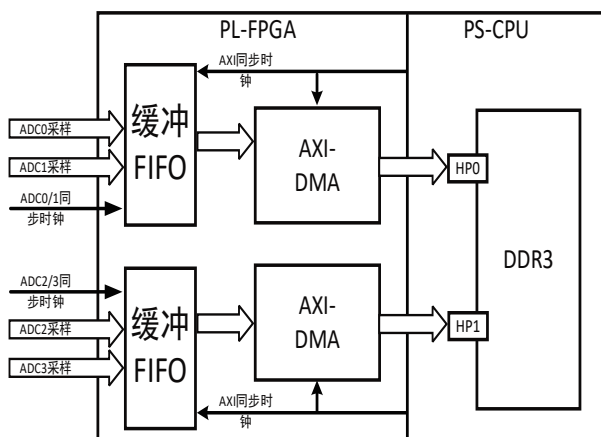


图 2 DMA 传输系统设计架构图

输入的 ADC 采样数据通过两组完全独立并行的 DMA 传输数据通道进入 DDR3 缓存器, 该数据通道包含: 独立的缓冲 FIFO、独立的 AXI-DMA 硬件控制器以及独立的 CPU 端高性能数据传输端口 HP0 和 HP2。其中独立的 FIFO 缓冲来保证完成数据从 ADC 到 CPU 系统之间的时钟域转换, 完成对数据结构的重新排列和重新组织。独立的 AXI-DMA 硬件控制器则在 CPU (PS) 进行配置之后, 按照配置地址独立地对 DDR3 内存进行数据读写。

如图 3 所示, 在 CPU (PS) 端的高速通道接口须选用 HP0-HP2 或 HP1-HP3 两个配对, 才能保证设计为并行的 CPU 高性能端口; 如选用 HP0 和 HP1, 会因两个端口同属于一个总线, 从而导致传输过程中产生仲裁, 严重影响传输效率, 降低 PL 到 PS 系统之间的传输带宽^[4]。

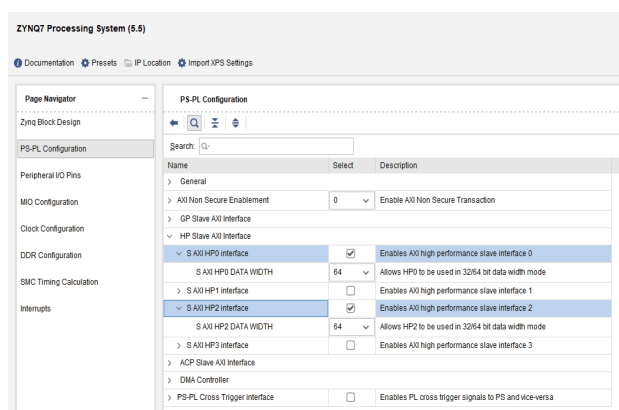


图 3 CPU 端选择高性能端口配置图

1.3 UDP/TCP 和 SD2.0 文件传输系统设计

文件传输系统的程序架构如图 4 所示, 控制应用程序运行在 ZYNQ 的 ARM 核心上, 采用 freeRTOS 实时操作系统。考虑到 DDR3 存储器容量有限, 系统设计了两种数据转存方案: TF 卡存储和 UDP 网络传输。TF 卡的大容量和低成本适合大容量存储的需求; 而以太网传输方面, UDP 作为一种无连接的传输协议, 具有速度快、资源开销小等优点, 适用于实时数据传输。而 TCP 作为一种面向连接的可靠传输协议, 可以保证数据的完整性和正确性。用户通过上位机配置来选择用 TF 卡存储还是以太网传输; 中间层程序负责将之前 PL 逻辑通过 DMA 直接写入 DDR3 缓冲区的 ADC 采样数据取回, 然后发送给需要发送数据的 TF 卡驱动层接口或者 UDP 以太网协议栈数据接口, 从而完成数据交换。

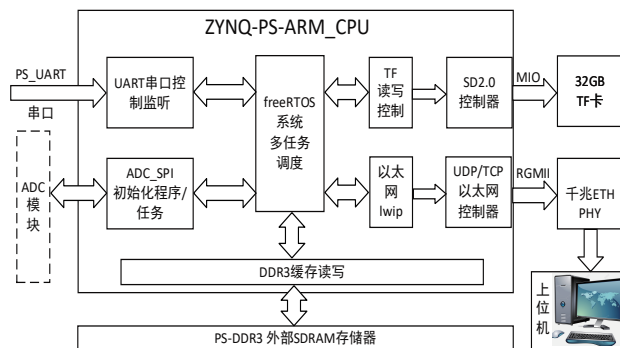


图 4 以太网和 TF 卡传输设计架构图

在本系统的以太网传输系统中, UDP/TCP 控制器通过 SGMII 接口与 PS 相连, PS 软件内部集成了 UDP/IP 和 TCP/IP 网络协议栈。为进一步优化 UDP 传输的性能, 在 PS 应用程序中采取以下措施: 一是增加环形缓冲区的长度来一次传输更多的数据包; 二是增大接收缓冲区的大小, 避免数据包在环形缓冲区中堆积; 三是软件设计上, UDP/TCP 传输使用 Xilinx SDK 提供的 lwIP 协议栈。通过配置 EMAC 和 GMII 接口, 实现千兆以太网的初始化和连接。

SD2.0 文件传输系统则使用 SD2.0 规格的 TF 卡连接到

处理器系统 (PS) 侧的专用 MIO 接口, TF 卡其最大理论传输速率可达 20 MB/s。实际 TF 卡的写入速度与其规格、数据长度以及数据写入时的寻址方式等因素有关。

此外, 对于固定长度的数据, 采用 FAT32 格式化的 TF 卡时, 其写入速度高于 exFAT 格式约 20%。对于本文系统而言, 采用 FAT32 格式可获得更优的 TF 卡写入性能^[5]。TF 卡的写入方式也会影响写入速度。使用块写入的方式, 以 512 字节对齐的方式写入, 其写入速度最高, 而按照任意字节地址写入时, 速度会有所下降。经测试, 对于 1 GB 的数据, 按照 512 字节对齐写入速度为大于 20 MB/s。因此, 本系统中的 TF 卡写入地址均按照 512 字节边界对齐, 可以获得最佳的存储性能。在 CPI 端, 使用 Xilinx SDK 提供的 FatFs 文件系统库来管理 TF 卡上的文件。通过配置 SD2.0 控制器和 MIO 管脚, 实现了 TF 卡的初始化和挂载。然后通过 FatFs 库的 `f_write` 函数将 DDR 缓存的数据写入 TF 卡。

2 FMQL/ZYNQ 系统设计

2.1 ZYNQ 核心软件架构设计

FMQL/ZYNQ 系统采用基于 ZYNQ 的 PS 与 PL 的异构结构, 充分发挥了 ZYNQ 器件的高性能和灵活性。基于 ZYNQ 系统的设计可以使用 BlockDesign (BD) 辅助来进行, 本系统中的 BD 包含有两路完全并行的 FIFO-AXI_DMA 数据传输系统, 该传输系统由外部 PL 子系统连接到 ZYNQ 的 CPU 处理硬件核心。该传输系统的数据源来自经过 AD 采集转换得到的数据, 数据宽度为 64 bit, 数据速率为 50 MHz。为提高系统性能, 减轻 PS 端 CPU 的负担, 提高数据吞吐量, 系统采用 AXI-DMA 直接存储器访问的方式, 将并行数据迁移到 PS 端的 DDR3 存储器中, 设计双通道 DMA, 即同时数据位宽为 128 bit, 最大传输速率可达 9.6 Gbit/s。

为保证数据的高效存储和传输, 设计了两个 DDR 数据缓存区, 每个缓存区大小为 1 GB。当一个缓存区存满后, 系统自动切换到另一个缓存区继续存储数据, 同时唤醒 PS 端的数据存储任务, 将已存满的缓存区数据通过 SD2.0 接口存储到 TF 卡或者通过 UDP 网络传输。SD2.0 接口采用 4 bit 宽度, 时钟频率 50MHz。UDP 网络传输采用千兆以太网接口, 包含协议开销的理论传输速率可达 125 MB/s。

部署数据存储和网络传输任务的应用程序, 通过轮询方式查询缓存区的数据状态, 当有缓存区存满数据时, 立即启动数据存储或网络传输任务。对于网络传输, 系统采用多线程并发传输的方式, 每个线程独立传输一个数据包, 数据包大小为 32 kB。多线程并发可充分利用千兆以太网的带宽, 提高数据传输效率。

2.2 FMQL/ZYNQ 中数据控制流程设计

控制软件采用了多级流水线的架构设计, 其中 DMA 控制器负责 DDR3 存储器与 PL 侧 IP 核之间高速数据传输。其中, DMA 控制器通过 AXI 总线与 PS 连接, TF 卡控制器通过 SDIO 接口与 PS 连接, UDP/TCP 网络传输控制器通过 SGMII 接口与 PS 连接。整个系统的数据处理流程图如图 5 所示。

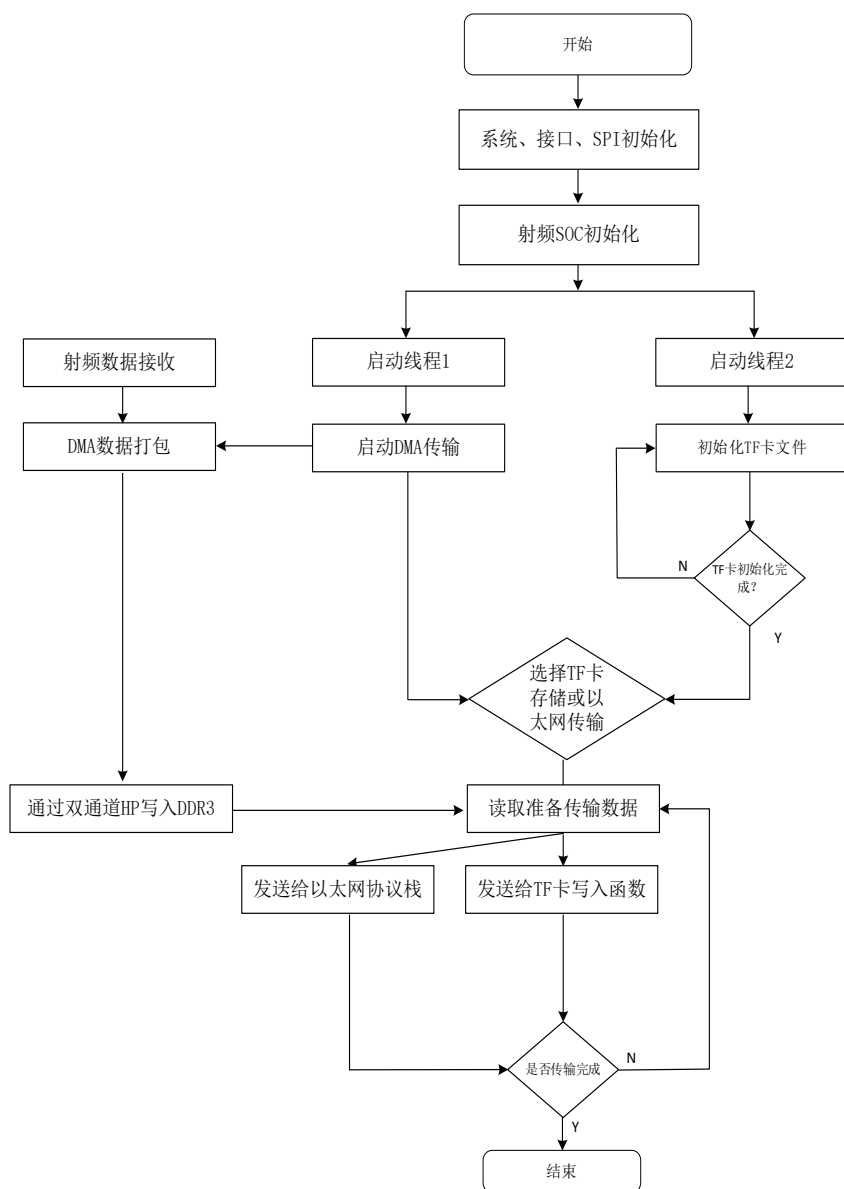


图 5 基于 SD2.0 和 UDP 以太网的数据存储 FPGA 数据流程示意图

传感器或其他数据源产生的原始数据首先写入 PL 侧的 FIFO。当 FIFO 中数据量达到设定的阈值后,就会向 DMA 发出数据传输请求。DMA 收到请求后会开始一个突发传输事务,将 FIFO 中的数据批量搬移写入到 DDR3 存储器的地址。这个过程可以在 CPU 不介入的情况下完成,极大地提高了数据传输效率。

当用户选择通过使用 TF 卡存储数据,通过设计控制数据存储容量的中断机制,可以实现采样数据向 TF 卡的自动转存。具体而言,当 DDR3 内存的存储空间被 PL 写入达到一定阈值时,就会触发一个中断给 PS 的 CPU 程序。该程序通过 SDIO 接口将 DDR3 指定地址范围内的数据写入 TF 卡。整个转存过程可以与数据采集搬移和预处理并行执行,从而提高了系统的实时性能^[6]。使用的 TF 卡可提供最高 32 GB 的存储空间,可以满足大多数应用场景的数据存储需求。测试结果表明,在 64 MB 数据量下,写入速度可达 21.6 MB/s,已经达到 TF 卡的理论写入速度。

对于第二种 UDP 数据转存方案,当用户选择通过以太网发送数据,应用程序将数据从 DDR3 缓存区读取发送到千兆 ETH-PHY 芯片然后传输到网络上。与 TF 卡存储方案的流程类似,UDP/TCP 传输也采用中断触发机制:当 DDR3 存储器数据量达到阈值时自动启动网络传输任务。应用程序将驱动 DMA 把网络传输的数据包迁移到内存中的环形缓冲区(rx_ring),然后由中断处理程序通知内核协议栈处理接收的数据包,最后应用程序通过 socket API 从接收缓冲区中读取数据^[7]。测试结果表明,在 64 MB 数据量下,UDP 传输速度可达 112 MB/s, TCP 传输速度可达 86 MB/s,虽然相较于理想的 UDP/TCP 的连接吞吐量有一定的性能损失,但还是达到了较高的传输速度。

3 基于 DMA 的数据存储系统测试

如图 6 所示的实验搭建环境,在本实验中,使用了基于 ZYNQ 器件系统平台,通过千兆以太网线连接测试板卡与测试 PC 计算机,通过信号发生器产生一组输入射频 AD 芯片的单音信号进行采集测试。

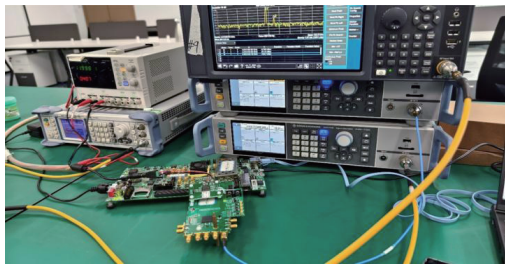


图 6 以太网连接测试板卡与计算机、信号源测试环境图

4 结语

本文提出并设计实现了一种基于国产 ZYNQ/FMQL 的数

字处理系统,采用了 AXI-DMA、TF 卡存储和 UDP 以太网传输相结合的数据存储方案。并对各关键模块进行了性能测试与分析。通过合理配置并调用其中各子系统,实现高速、高效、大容量的数据存储和传输。该系统架构具有较强的实用性和可扩展性,可以在数字信号处理、图像处理、高速数据采集等领域得到广泛应用。

不过,本系统在实际应用中仍存在问题有待进一步研究和改进。例如,还需要考虑数据的完整性和可靠性。对于 TF 卡存储,可以在写入数据后,读取并校验数据,确保写入的数据无误。对于千兆以太网的 UDP 传输,通过上位机配合在数据包中添加序号和校验和字段,通过重传机制来保证数据的可靠到达^[8]。以及进一步提高 TF 卡的读写速度,优化 UDP 网络传输的效率和可靠性^[9],以及如何将该系统与其他数字处理模块进行更紧密地集成等。此外,随着数据处理需求的不断增长,还需要研究如 PCIE 等更高性能、更大容量的数据存储方案,以满足更加苛刻的应用需求。

参考文献:

- [1] 高尚尚,王新宇,王小丫,等.基于以太网传输的图像处理系统设计与 FPGA 实现[J].计算机测量与控制,2022,30(7):213-218.
- [2] 郭宇,郑国辉.基于 FPGA 的 SD 卡高速数据存储系统设计与实现[J].电声技术,2017,41(6):39-42.
- [3] 齐志强.基于 FPGA+COM Express 的基带数字信号处理平台设计[J].计算机测量与控制,2019,27(1):255-258.
- [4] 杨海学,吴凡,李林,等.XCZU9 的高速大容量数据采集存储系统设计[J].单片机与嵌入式系统应用,2019,19(12):84-87.
- [5] 王子懿,沈三民,杨峰,等.基于 FPGA 的高速大容量存储与传输系统[J].电子测量技术,2021,44(13):150-155.
- [6] 芦夜召,张会新.基于 FPGA 的多路数据同步采编存储系统设计[J].现代电子技术,2021,44(4):13-16.
- [7] 郑骏,徐海纳.基于 Zynq-7100 数据存储模块的设计[J].电脑知识与技术,2020,16(23):45-46.
- [8] 张昊宇,甄国涌,陈建军.基于 FMQL 的千兆以太网传输系统设计[J].单片机与嵌入式系统应用,2022,22(2):76-80.
- [9] 李健.应用于 SoC 的千兆以太网 MAC 控制器设计与实现[D].西安:西安电子科技大学,2023.

【作者简介】

盛磊(1993—),男,甘肃酒泉人,硕士研究生,助理工程师,研究方向:雷达信号数字处理算法与设备。

(收稿日期:2025-04-16 修回日期:2025-09-04)