 DOI: https://doi.org/10.12349/tie.v3i8.11532

Design and Implementation of a Distributed Data Acquisition System Based on the Heterogeneous Architecture of RISC-V and FPGA

Zhenhong Zhao

Shenzhen Kezhao Technology Co., Ltd., Shenzhen, Guangdong, 518000, China

Abstract

In industrial IoT device condition monitoring projects, we frequently encounter challenges such as a large number of channels, stringent synchronization requirements, and limited on-site power supply. To address these issues, we developed a distributed acquisition system based on an FPGA-RISC-V architecture: the FPGA controls 16 ADC channels for simultaneous sampling and performs basic data preprocessing, while the RISC-V processor (BK7258) packages the data and transmits it via Wi-Fi or Ethernet. Experimental results demonstrate a synchronous sampling rate of 1 MSPS across all 16 channels, a channel-to-channel synchronization error below 1 μs, and a packet loss rate under 0.02% during continuous operation for 72 hours. To illustrate this architecture, we conducted a comparative analysis with our previous smart instrument project for two-wheeled vehicles (using the BK7258). These results confirm that RISC-V is fully capable for real-time acquisition and control applications.

Keywords

RISC-V; FPGA; distributed data acquisition; heterogeneous architecture; Industrial Internet of Things

基于 RISC–V 与 FPGA异构架构的分布式数据采集系统设计与实现

赵镇洪

深圳市科昭科技有限公司，中国 · 广东 深圳 518000

摘　要

在工业物联网设备状态监测项目中，我们经常遇到通道数多、同步要求高、现场供电受限的情况。为此，我们设计了一套基于“FPGA+RISC-V”组合的分布式采集系统：FPGA负责控制16路ADC同时采样，并做简单的数据预处理；RISC-V处理器（BK7258）则负责把数据打包并通过Wi-Fi或以太网上传。实测下来，16通道同步采样率达到1MSPS，通道间同步误差小于1μs，连续运行72小时丢包率低于0.02%。为了把这套架构讲清楚，我们还拿之前做的一个两轮车智能仪表项目（基于BK7258）来对照分析，事实证明RISC-V在实时采集和控制场景中完全够用。

关键词

RISC-V；FPGA；分布式采集；异构架构；工业物联网

1 引言

去年我们在给一家设备制造商做产线振动监测时，发现传统的采集板卡要么通道数太少，要么同步精度不够，而且功耗偏高，现场只能用电池供电，撑不了几天。当时我们就在想，能不能用 FPGA 做并行采集，再用一个低功耗的处理器做网络传输？

RISC-V 是加州大学伯克利分校 2010 年发布的开源指令集架构，和 ARM、x86 不同，它允许用户自由修改和扩展指令。对我们来说，这意味着我们可以基于 RISC-V 做定制化设计，而不用担心被某家商业 IP 绑定。

所以，我们决定尝试把 FPGA 和 RISC-V 结合起来，FPGA 负责采集前端，RISC-V 负责协议处理和通信，看看能不能在性能、功耗和成本之间找到一个好的平衡点。

2 系统总体架构

2.1 整体方案

我们把系统分成两个处理单元。FPGA 端（选的是 Lattice 的 ECP5 系列）负责直接控制 16 路 ADC 的采样时序，同时做简单的数字滤波（比如均值滤波），然后把数据整理好放到共享缓存里。RISC-V 端用的是博流智能的 BK7258

【作者简介】赵镇洪（1986–），男，中国广东江门人，本科，从事电子技术研究。

芯片，它内置一个 RISC-V 核心，主频够用，而且集成了 Wi-Fi 和蓝牙，省去了外挂无线芯片的麻烦。RISC-V 端跑的是 FreeRTOS，上面跑了 LWIP 协议栈和 MQTT 客户端，负责把数据封装成报文上传到服务器。

两者之间通过双端口 RAM（DPRAM）交换数据，接口是 QSPI，速度能达到几十兆 bps。控制命令则通过 APB 桥接器由 RISC-V 写寄存器来配置 FPGA 的工作模式。这样做的好处是，FPGA 只管“采”，RISC-V 只管“传”，各司其职，不会互相抢资源。

2.2 多通道同步采集设计

同步采集的关键是让 16 个 ADC 在同一时刻开始采样。我们采用“统一时钟＋同步脉冲”的做法：用一个低抖动的时钟 Buffer 把 50MHz 参考时钟分发给所有 ADC，同时 FPGA 产生一个 SYNC 脉冲，经过等长布线送到每个 ADC 的触发引脚。所有 ADC 在 SYNC 上升沿同时锁存当前采样值。

不过，实际 PCB 上走线总会有微小差异，所以我们还在 FPGA 内部做了一个动态相位监测模块，每次上电时自动调整 SYNC 脉冲的延迟量，确保各通道的采样时刻对齐。

2.3 处理器与 FPGA 通信接口

双端口 RAM 我们用的是片外的一块 SRAM，FPGA 和 RISC-V 都能访问。FPGA 把 16 通道的数据按固定格式写入 RAM 的不同区域，写满一个缓冲区后，给 RISC-V 发一个中断。RISC-V 收到中断后，启动 DMA 把数据从 RAM 搬到自己内存里，同时 FPGA 切换到另一个缓冲区继续写——这就是经典的 Ping-Pong 缓冲，可以保证数据不丢失。

控制命令走的是 APB 总线，RISC-V 通过 APB 桥接器直接读写 FPGA 内部的配置寄存器，比如设置采样率、开启或关闭某个通道等。

3 关键技术实现

3.1 RISC–V 处理器软件架构

我们把软件分成了四层，这样以后换其他 RISC-V 芯片也容易移植：硬件层：处理芯片启动、中断向量表、外设驱动（GPIO、UART、QSPI、DMA 等）。中间层：封装了芯片架构差异和板级配置（比如不同型号的 ADC 对应的引脚定义）。组件层：集成了 LWIP 协议栈、FatFS 文件系统、MQTT 客户端和 JSON 解析库。应用层：实现采集任务、数据处理任务、网络发送任务和命令行交互任务。

数据流方面，我们用了上面提到的双缓冲机制。

3.2 分布式同步机制

如果以后扩展成多个采集节点，就需要节点间的时间同步。我们参考了 IEEE 1588 的思路，但做了简化：主节点通过 Wi-Fi 广播一个带时间戳的同步帧，从节点收到后记录本地时间，然后回复一个确认帧，主节点根据往返时间计算出时钟偏移并告诉从节点。这样在局域网环境下，同步误差能控制在 20 μs 以内，对大多数工业监测场景来说足够了。

4 以两轮车智能仪表项目为例的系统实现

前面几节讲的都是通用架构，那在实际产品中这套思路怎么落地？我们正好有一个基于 BK7258 的两轮车智能仪表项目。我们仔细分析过它的代码和任务调度，发现它的数据流管理方式和我们采集系统非常相似，所以拿出来做对比，也能验证 RISC-V 的实用性。

4.1 项目背景与系统概述

这个两轮车仪表方案用的是博流 BK7258（内置 RISC-V 核心）搭配 BK3515NS 蓝牙芯片，主要功能是：通过 Wi-Fi 接收手机投屏的图像，显示在 1024 × 600 的 LCD 上；同时支持蓝牙 A2DP 播放音乐和 HFP 免提通话。系统还跑了一个 LVGL 图形界面，用于显示车速、电量等信息。

从软件分层来看，它和我们前面说的四层结构几乎一样：

硬件层：BK7258 核心、LCD 屏、触摸屏、音频 Codec 等。

中间层：启动代码、QSPI LCD 驱动、UART 蓝牙通信、GPIO 按键等。

组件层：LVGL 图形库、Wi-Fi 协议栈、蓝牙协议栈（A2DP/HFP）、TCP/IP 协议栈。

应用层：Wi-Fi 图传切换、蓝牙播放控制、界面管理。

4.2 高速数据流处理机制

这个项目中，图像数据从 Wi-Fi 网络收到后，要经过解码才能在 LCD 上显示。整个过程和我们采集系统中“数据接收→预处理→输出”非常像，所以下面重点看看它是怎么管数据的。

4.2.1 帧缓冲区队列管理

项目定义了两个队列：

frame_buffer_queue：管理 UVC 摄像头采集的原始图像或 H264 压缩码流，用于编码和存储。

media_data_process_queue：管理从 Wi-Fi 接收到的导航 / 图传图像，用于实时显示。

这两个队列都提供了统一的 API：生产者（网络接收任务）把数据放进队列，消费者（解码显示任务）从队列取数据，互相不阻塞。这和我们用双端口 RAM 做生产者 - 消费者解耦的思路如出一辙。

4.2.2 图像数据接收与解码流程

具体流程如下：调用 video_data_process_open 启动一个接收线程，监听网络端口。当网络数据包到达时，回调函数 video_data_receive_complete 被触发，根据协议类型（TCP/UDP）解析数据。通过 media_frame_queue_* 系列函数把零散的小包拼成一幅完整的 JPEG 帧。启动 JPEG 解码管理器 av_server_jpeg_decode_manager_turn_on，解码后的图像直接刷新到 LCD，同时暂时冻结 LVGL 界面更新，以免画面撕裂。关闭时，按相反顺序停止线程、关闭解码器、释放队列。

这个流程完全可以套用到我们的采集系统——FPGA 来的数据就是“网络包”，经过预处理后，由 RISC-V 打包并

发送到云端。

4.3 并发任务调度与资源管理

这个项目跑着好几个任务，它们通过消息队列和信号量来协同工作。

4.3.1 多任务并发架构

Wi-Fi 网络任务：负责连接管理、TCP/UDP 收发包、图像数据接收。显示管理任务：控制 LVGL 界面和图传画面的切换，处理触摸事件。蓝牙协议任务：处理 A2DP 音频流（含 SBC 解码）、HFP 双向音频、AVRCP 遥控命令。用户命令任务：通过串口解析 ap_cmd 命令行，动态切换模式或调试参数。

4.3.2 进程间通信机制

我们特别留意了蓝牙音频的处理方式：当音频数据回调 bt_audio_sink_media_data_ind 被触发时，它并不直接解码，而是发送一个 BT_AUDIO_D2DP_DATA_IND_MSG 消息到主任务队列。主任务从队列取出消息后，根据编码类型（如 SBC）计算帧长，解码成 PCM 后写入音频播放缓冲区。这样就把中断上下文和业务处理分开了，不会阻塞中断。

这种消息驱动模式，正好对应我们采集系统中“FPGA 中断通知 → RISC-V DMA 读取 → 协议栈封装发送”的流程，本质上都是异步解耦。

4.4 系统配置与调试方法

项目用 Kconfig 做了很多可配置项，方便针对不同硬件版本或需求进行裁剪：CONFIG_MEDIA_DEMO_MODE_TCP：选择 TCP 还是 UDP 传图。

调试方面，项目留了一套 CLI 命令，我们在开发采集系统时也借鉴了这种做法，留了几个调试命令，现场排查问题方便多了。

4.5 对本数据采集系统的启示

通过分析这个两轮车项目，我们总结了几点可以直接用到采集系统里的设计思路：数据流用队列解耦：不同类型数据分队列管理，生产者和消费者独立。任务按功能拆分：网络、显示、音频、命令各归各的任务，响应更快。硬件差异通过配置管理：用 Kconfig 和板级文件隔离变化，代码复用率高。留调试接口：CLI 命令在开发阶段和现场运维时都是救命稻草。计算能力验证：BK7258 能跑 JPEG 解码和音频编解码，说明它的 RISC-V 核心处理能力不弱。

5 系统测试与结果分析

5.1 测试环境

我们搭了一个测试台：16 个通道同时接入 1kHz、2Vpp 的正弦波信号，ADC 分辨率 16 位，采样率设为 1MSPS。数据通过有线以太网和 Wi-Fi 两条路分别上传到本地服务器，跑了整整 72 小时。

5.2 测试结果

测试项目　指标要求　实测结果（有线 / 无线）　结论

采样率　≥1MSPS　1MSPS　达标

通道同步误差　≤5 μ s　≤0.8 μ s　优于指标

数据丢包率（72h）　<0.1%　<0.02%　达标

Wi-Fi 实际吞吐量　-　满足 1080P 图传需求　满足要求

系统待机功耗　-　微安级（基于 BK7258）　优于预期

关于同步误差，我们特意做了反复测试：把 16 个通道都接同一个信号，用 FPGA 内部的逻辑分析仪抓采样时刻，计算各通道相位差。结果最大偏差只有 1 个采样周期（1 μ s），而且断电重启了十几次，偏差值都没变。

5.3 方案对比

对比维度　本方案（RISC-V+FPGA）　传统 ARM+FPGA 方案　优势说明

处理器授权成本　零　较高　RISC-V 开源优势

系统功耗　极低　较高　适合电池节点

无线集成度　高（可选内置 Wi-Fi/BT 芯片）　需外挂　简化 BOM

软件生态与复用度　可采用分层架构与组件化设计　成熟但封闭　灵活性与可移植性高

6 结论与展望

这次设计让我们切身体会到，FPGA+RISC-V 的组合在工业数据采集上确实有搞头。FPGA 的并行性保证了多通道高速同步采集，RISC-V 的灵活性则让我们可以方便地定制协议栈和通信方式。而通过分析那个两轮车智能仪表的代码，我们更确认了：无论是图像数据还是传感器数据，数据流管理和多任务调度的底层逻辑是相通的，RISC-V 完全能扛住这些任务。

从这个项目里，我们最大的收获不是“验证了可行性”这种空话，而是实实在在踩过的一些坑和解决经验。比如刚开始 QSPI 通信时钟相位没调对，FPGA 和 RISC-V 之间传数据老是错位，后来在 FPGA 里加了动态延迟校正才搞定。这些经验对我们后续做同类产品很有用，也让我们相信 RISC-V 在工业测控领域是可以落地的。

下一步，我们打算把多个采集节点组成分布式网络，把 IEEE1588 同步协议移植到 Wi-Fi 或者有线以太网上，让节点间的时间同步精度再提一个档次。另外，BK7258 有足够的算力，我们计划在 RISC-V 端部署一个轻量级的异常检测模型（比如孤立森林），直接在采集节点上做初步的故障判断，这样能减少上传的数据量，也能及时预警。

参考文献

[1] 陈龙震,徐康民,徐天骅,等.RISC-V架构的交叉调试系统设计[J].单片机与嵌入式系统应用.2023,23(12).

[2] 叶朝辉,张仁刚,赵腾浩,等.基于RISC-V的SOPC电子系统设计实验研究[J].实验技术与管理.2023,40(11).

[3] 王媛,孙立宏,胡孔阳.一种基于ALU运算单元的译码分配电路设计[J].中国集成电路.2022,31(1).