“一种高性价实验实时数字控制系统”的先期调研

刘冀洋

21305088

物理与天文学院 物理专业

**背景介绍：**

如今实验光学平台许多都使用外购的商业系统，其实架构封闭，功能单一独立，且价格相对昂贵，性能提升空间有限，故具体实验时，总要大量购买不同的功能模块以实现众多的控制要求，堪称“万国造”，既耗费了大量精力去整合使用又在故障发生时难以及时排解，不适宜长期的实验需要。

因此我所在本科生团队所计划的2024年大学生自主创新创业项目旨在设计一个可广泛用于任何物理实验的高性价比的实时数字控制系统（ms以内延迟），以将对实验平台的数据记录、计算、反馈输出功能进行系统整合。系统具体实现方式多样，既可以使用Real-Time Linux等实时操作系统对任务进行直接运算；也可以使用可编程门列阵（FPGA）等外部硬件设备作为受控运算系统。整体系统设计思路是：系统主体由一个核心与多个终端组成，预期可以有效处理大计算量和大数据流任务，并对实验平台进行一体控制。

鉴于具体实现的多样性复杂性,为了选择最优实现路线,我认为前期的调研很有必要。

**一、 FPGA（现场可编程逻辑门阵列）**

FPGA是一种半定制的有限编辑的专用集成电路，内部有大量逻辑闸（几万到几百万,远多于CPLD），对于信息处理能力强，且其价格相对低廉，功能拓展性很强,生态开放良好，可以快速编辑形成成品，并进行后期灵活调整，故在一些高信息密度（如视频音频通信）和快速更新的技术领域（如智能算法）领域具有商业与技术上的巨大优势,被誉为“万能芯片”。

而在其具体实现中则涉及使用硬件描述语言去设计硬件，主要使用Verilog HDL语言集成设计到模拟的全过程，且FPGA在上世纪九十年代就开始有规模地商业化，如今有数个国际厂商都有相关软件服务,如Altera的Quartus\_Ⅱ跟Xilinx的Vivado等，整个开发系统成熟。

对于本项目其主要优点是极高的性价比与高度灵活性，加上通信接口后，可作为系统的从属计算核心，完成大量高速多线工作，从而为实验平台中的大量动态受控单元提供运算服务，廉价且性能拓展空间大。而主要缺点则是，其编辑能力终究有限，运算过程无法进行分支跳转等相对复杂的整体操作，计算子核心间隔离较大，再者软体嵌入过程相对死板，需要自行开发与外部深层通信的相关机制（如与实时操作系统）。

**二、 实时操作系统（以Real-Time Linux为代表）**

实时操作系统（Real-time Operating System），简称RTOS，的主要特征是低延迟，定时内处理指定任务，对于物理实验这种需要高速动态控制的实体系统来说，这一特性至关必要。以Real-Time Linux为代表，可以结合LabVIEW Real-Time 处理端口事宜，使用第三方语言（C、C++、Python）进行软体编写，原则上，整个运算处理过程原则上均可在计算机内部运行，但也可要尝试将任务分散到FPGA里，以提高运算速度。Real-Time Linux的实时原理主要来源于“定时完成”这一限定下的优先级排布，不再将机器时间均匀分配给每一个进程，而在调度上进行优化，如允许“抢占”等，具体上通过可剥夺型内核加实时补丁实现对外部中断的快速响应。

总之，鉴于FPGA生态的丰富与实时操作系统的成熟（广泛应用于工业实验等高安全性情景），建立实时操作系统与FPGA系统间的通信与主从关系应是可行的。以前者的齐全功能弥补后者的有限性，以后者的高性价比弥补前者的低性价比，我认为

**二、 调查总结**

经调研，我们暂定选择对FPGA专用电路与实时操作系统进行结合，我们预期建立：

➡实时操作系统决策

Analog输入➡ADC转换➡Data数字空间 ➡多样输出反馈

➡FPGA专用电路运行

总之在整个系统中，任何一个环节都要低延迟，高响应，以满足整体的高速要求。且鉴于FPGA的高度灵活性，随实验展开，我们希望能自由调节其功能，如扩展其线路数量、增加特定运算模块等。

（附：在调查中，我发现有些FPGA系统已存在把CPU系统内嵌以内部运行Linux，可作为参考，在系统搭建中，可以考虑在空间结构上对该系统的进一步结合。再且系统输出反馈的具体形式，则已经实现（赋能或不赋能等），在此不加赘述。）

**参考文献：**

[What is Real-Time Linux? - GeeksforGeeks](https://www.geeksforgeeks.org/what-is-real-time-linux/)

[Intro to Real-Time Linux for Embedded Developers - Linux Foundation](https://www.linuxfoundation.org/blog/blog/intro-to-real-time-linux-for-embedded-developers)

[Intel® FPGA Basics and Getting Started | Intel](https://www.intel.com/content/www/us/en/support/programmable/support-resources/fpga-training/getting-started.html)

[How Does an FPGA Work? - SparkFun Learn](https://learn.sparkfun.com/tutorials/how-does-an-fpga-work/all)

[Programming an FPGA: An Introduction to How It Works (xilinx.com)](https://www.xilinx.com/products/silicon-devices/resources/programming-an-fpga-an-introduction-to-how-it-works.html)

https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000P6 VaSAK&l=zh-CN