面向FPGA的软硬件结合异步状态机接口设计与实现

**选题依据:**

随着计算器件的不断发展, 出现了各种用途高度分化的计算单元, 如 CPU, GPU, FPGA 等. 为了发挥不同计算元件的特长, 结合多种计算器件的异构计算显得越发重要, 结合 CPU 和 GPU 的 OpenGL 库已经被广泛应用于人工智能等程序的优化. FPGA 具有很强的并行运算特性, 适合编写高并行低时延的程序; 但是 FPGA 并不适合进行浮点数等运算, 这些计算最好通过 CPU 来执行. 如果可以充分利用二者的计算特长, 并且减少切换通信之间的时延, 实现软硬件程序异步执行, 从而实现 CPU 和 FPGA 之间高效并行异步计算.

**研究背景:**

CPU 和 FPGA 程序并行的主要难点在于高效的切换和通信, 已有的 CPU 调用 FPGA 的通信方式有通信协议和共享内存两种, 但都存在效率较低, 难以并行的问题, 不能充分满足高效切换的需要; FPGA 调用 CPU 方面的研究较少, 研究成果主要集中在 FPGA 调用 CPU/FPU 来弥补浮点运算效率的不足, 但缺少异步并行执行的支持.

**研究目标:**

实现 CPU 和 FPGA 高效, 封装, 可迁移的异步并行接口, 允许二者在运行过程中异步调用对方, 从而实现并行, 非阻塞的运行. 同时希望设计一套测试调试工具, 在 FPGA 调用 CPU 时, 支持查看正在执行的软件程序地址, 寄存器内存数据等方式, 方便监测进度以及调试, 并分析资源开销. 并且可以根据 CPU 计算进度, 自动选用更好的计算模块进行计算, 实现对用户透明的异构计算.

**技术方案:**

首先利用回环串口和中断, 实现 CPU 与 FPGA 之间的通信与调用; 之后在此基础上进行改进, 通过改进中断实现, 进行包含数据传输的中断分离, 降低协议等方面的通用性, 设计流水线更友好的调用方式等方法来提高调用切换效率.

在 FPGA 调用 CPU 时, 参考 gdb 的单步调试功能, 让 CPU 取消分支预测等功能, 逐步执行汇编语句, 在 FPGA 运行时将 CPU 相关运行数据通过串口发送给 FPGA, 达到实时监测, 调节负载等功能.

设计矩阵乘法等计算例程来测试计算效果, 从而和传统纯CPU /纯 FPGA 程序进行效率对比, 评判异构计算程序的时间空间消耗, 计算加速比.

将以上代码封装成高级语言库和硬件语言模块, 通过开源平台发布.