**基于SPARC模拟器的多核调试技术研究与实现**

**报告提纲**

1. **选题的背景及意义**

在SPARC模拟器上需要对应用程序进行调试。由于模拟的是多个BM3803核，因此需要实现针对SPARC架构的多核调试。

在多核调试过程中，为了满足调试态对运行态更接近的模拟，需要实现多核同步调试的功能；而为了控制不同处理器核的运行状态以及信息交互，需要保证调试可以异步执行。

1. **国内外本学科领域的发展现状与趋势**

现阶段，虽然多核架构芯片已成为高端嵌入式产品的首选，但国内外在嵌入式调试技术方面、尤其是基于多处理器所做的研究工作较少，较为成熟的多核硬件调试工具都是基于 JTAG 技术，并且都需要较高的成本。作为单核架构下首选的任务级调试器 GDB，虽然能在嵌入式 Linux 支持下调试多核架构下的程序，但原有单核调试功能远不能满足用户多核调试的需要。

在多核调试过程中，由于系统复杂度和系统的不确定性，多个处理器核之间通信和同步等问题，使得调试难度比单核调试增大很多。虽然高版本的GDB协议支持远程多核调试，但国内外公开的研究中，还几乎没有基于SPARC架构的很好的解决方案。

1. **课题主要研究内容**

* 多核调试协议的可扩展的交互机制研究。研究模拟器与宿主机间的交互，根据协议标准，设计易于扩展的协议处理机制，为模拟器提供传输接口。
* 基于模拟器的同步多核调试中断点处理机制的研究。断点处理是调试过程的核心，通过研究断点纪录策略和以及多种类型断点处理的机制，在没有操作系统的情况下，完成调试命令。
* 基于模拟器的异步多核调试机制研究。在完成同步多核调试的基础上，改进模拟器，支持宿主机端与模拟器进行异步交互，同时支持多个处理器核间相互独立的调试模式。

1. **拟采用的研究方法、技术路线**

* 宿主机端与模拟器通过RSP协议进行交互，通过集成flex分析器以及模拟器接口的设计，完成协议的高效转化并提供更高的扩展性。交互过程包括序列化和反序列化两部分，重点研究如何快速完成这一过程，同时通过定义交互接口，实现协议类型的快速添加，提高可扩展性。
* 在不借助操作系统的情况下，完成模拟器的断点调试功能。通过在模拟器上将调试事件集成到处理器核的调度中，对调试命令进行响应；设计相应的数据结构和断点检测方法，正确注册和判断断点的有效性；通过回调机制，保存运行现场并且完成调试命令。
* 完成基于模拟器的异步调试。在完成多核调试功能的基础上，采用事件缓存策略，允许宿主机发送多个调试命令，实现宿主机与模拟器的异步调试。同时完善断点处理机制，支持对某个处理器核下断点而不影响其它的核。在事件缓存中，将每个调试命令看作一个事件，保存到事件队列中，模拟器根据调度算法处理事件；处理事件时通过异步断点处理机制，判断事件对哪个核生效，只对相应的核进行操作。

1. **已有科研基础与所需的科研条件**

已经完成了双核SPARC模拟器的基本功能，在此基础上实现GDBServer功能。

对于GDB调试协议有一定研究，分析出了调试过程中主要的协议类型。

**6、 研究工作计划与进度安排**

**7、 参考文献**