

**毕 业 论 文（设计）开 题 报 告**



**论文(设计)题目：**基于RISC-V的嵌入式实时操作系统设计

**学生姓名：** 李轩 **学 号：** 2012001059

**专 业：** 软件工程

**所在学院：** 软件工程学院

**指导教师：** 闵建

**职 称：** 讲师

2023 年 12 月 20 日

开题报告填写要求

**1．开题报告（含“文献综述”）作为毕业论文（设计）答辩委员会对学生答辩资格审查的依据材料之一。此报告应在指导教师指导下，由学生在毕业论文（设计）工作前期内完成，经指导教师签署意见及所在专业审查后生效；**

**2．开题报告内容必须用黑墨水笔工整书写或按教务处统一设计的电子文档标准格式打印，禁止打印在其它纸上后剪贴，完成后应及时交给指导教师签署意见；**

**3．“文献综述”应按论文的格式成文，并直接书写（或打印）在本开题报告第一栏目内，学生写文献综述的参考文献应不少于15篇（不包括辞典、手册）；**

**4．有关年月日等日期的填写，应当按照国标GB/T 7408—94《数据元和交换格式、信息交换、日期和时间表示法》规定的要求，一律用阿拉伯数字书写。如“2017年4月26日”或“2017-04-26”。**

**5、开题报告（文献综述）字体请按宋体、小四或五号字体书写，行间距1.5倍（英文字体可采用其它合适的字体，但正文中的英文字体应统一，否则显得凌乱）。**

**6、对每一部分要审清标题，不符合标题的内容不能写入，一定不要把各种材料拿来拼凑某部分的篇幅。**

**7、绝对不能把网上（或书中）材料简单地拷入自己的开题报告（否则一票否决），一定要进行消化整理，用自己的语言，写出相关的内容。**

**8、指称同一概念的名词，全文中使用要统一（如AJAX与Ajax、WEB与Web，只能用其中的一种），以免造成读者的疑问和报告的不规范。**

**9、注意一定用书面语、务必避免口语化倾向，要用客观描述的科技文体，不使用绝对化词语以及夸张的文学词汇及表达，不能有对某种产品的夸张性宣传。**

**毕 业 论 文（设计） 开 题 报 告**

|  |
| --- |
| 1．结合毕业论文（设计）课题情况，根据所查阅的文献资料，每人撰写不少于1500字左右的文献综述： |
| 1. 选题背景和目的   随着物联网和嵌入式技术的快速发展，对于实时性和可靠性要求越来越高的应用程序变得日益普遍。嵌入式实时操作系统作为支持这些应用程序的关键技术之一，起到了至关重要的作用。然而，目前国内在嵌入式实时操作系统设计领域仍存在一定的挑战和问题。  国内嵌入式实时操作系统市场主要依赖于国外商业操作系统，这导致了对于自主设计和研发的需求。虽然一些国内研究机构和企业在实时操作系统方面有一定的研究活动，但在自主设计和验证方面仍然存在较大的欠缺。这使得国内在嵌入式领域的技术和创新受到一定限制。并且部分国内自主研发的实时操作系统体量过于庞大，对于一些资源受限的嵌入式平台不够友好，且对于开发者来说，过于复杂会导致开发成本的提高。  与此同时，RISC-V 架构作为一种简洁、可扩展和灵活的开源指令集架构，在国内也得到了广泛关注和应用。然而，目前在 RISC-V 平台上的嵌入式实时操作系统设计和验证研究还相对较少，缺乏自主设计的解决方案。  因此，本研究旨在探索基于 RISC-V 平台的自主设计的嵌入式实时操作系统，填补国内自主设计的嵌入式实时操作系统的空白。通过深入研究实时操作系统的设计原理和算法，并结合 RISC-V 的特性，本课题将开发一个具有实时性和可靠性的嵌入式操作系统，并通过验证实验来评估其性能和可行性。  本研究的主要目的是设计和验证一种基于 RISC-V 平台的自主设计的嵌入式实时操作系统。具体目标包括：  1、分析和研究现有的实时操作系统和 RISC-V 架构的特点，包括其优点和局限性。  2、设计和实现一个针对 RISC-V 平台的自主设计的轻量级嵌入式实时操作系统，具备任务调度、中断处理、资源管理等关键功能，以满足实时应用程序的需求。  3、进行性能评估，比较新设计的实时操作系统与现有商业操作系统在关键指标（如响应时间、任务调度效率、资源利用率等）上的差异。  4、使用实际的嵌入式应用程序对新设计的实时操作系统进行验证，分析其在不同应用场景下的性能和可靠性。  5、探讨新设计的实时操作系统的局限性和改进空间，并提出未来工作的方向。  通过实现上述目标，本研究将为国内嵌入式系统领域的自主设计和创新提供有价值的参考，提升国内在嵌入式实时操作系统领域的技术实力和自主研发能力，推动国内嵌入式行业的发展。   1. 文献综述   嵌入式操作系统在硬件资源受限的嵌入式场景中发挥着重要作用。它的应用领域涵盖了工业控制、航空航天、医疗设备、汽车电子和智能家居等多个领域，为这些领域提供了实时性、可靠性和效能方面的支持。与其他操作系统相比，嵌入式操作系统具有实时性、可裁剪性、可靠性、体积小等特点[1]。  在工业控制领域，嵌入式操作系统被广泛应用于自动化生产线、机械设备和工厂控制系统。它能够实时响应和处理各种传感器和执行器的输入输出，确保生产过程的稳定性和可靠性。同时，嵌入式操作系统还可以提供高度可靠的通信机制，实现设备之间的数据交换和远程监控。  在航空航天领域，嵌入式操作系统被广泛应用于航空电子设备、导航系统和航天器控制系统。它能够处理复杂的飞行算法和导航逻辑，实时监测飞行状态并做出精确的控制决策。同时，嵌入式操作系统还需要具备高度的可靠性和容错性，以应对严苛的航空航天环境和安全要求。  此外，嵌入式操作系统还在医疗设备、汽车电子和智能家居等领域发挥着重要作用。在医疗设备中，嵌入式操作系统能够实现实时监测和控制，确保医疗设备的安全和有效运行。在汽车电子领域，嵌入式操作系统用于管理车载系统、驾驶辅助系统和车联网功能，提供安全、智能的驾驶体验。在智能家居领域，嵌入式操作系统能够实现家庭设备的联网和智能化控制，提供便利、舒适的居家环境。  目前国外主流的嵌入式操作系统主要有VxWorks、Linux、Windows CE三种。这三种操作系统在实时性能、开发环境和适用领域等方面存在差异。VxWorks在实时性和嵌入式调试工具方面表现突出，适用于高实时性要求的领域，但需要付费购买许可证且学习曲线较陡；Linux具有精简的内核和广泛的适用性，适合多种体系结构和嵌入式应用开发，但实时性能相对较弱且学习难度较大；Windows CE作为商用嵌入式系统，提供稳定性和可视化开发环境，但需要购买许可证，开发者的定制能力有限且对硬件资源消耗较大[2]。因此，国内迫切需要一款自主开发的轻量级嵌入式实时操作系统，它应当具备小体积、高实时性和易上手等优势。  当前，ARM公司的Cortex系列和A系列处理器架构是嵌入式平台上的主流芯片架构。然而，与此同时，RISC-V架构也正在迅速崛起。RISC-V是一种新兴的开源精简指令集架构,由加州大学伯克利分校在2010年首次发布[3]。RISC-V架构具有开源、免费、开放和自由的特点，它的出现是为了解决现有体系架构长期发展中出现的各种问题，并满足现代信息系统设计需求和体系结构发展的要求[4]。RISC-V采用模块化设计，并提供丰富的自定义编码空间和指令集扩展能力，具备强大的系统定制化能力。与ARM相比，RISC-V的文档内容更为精简，仅有两卷329页，包括238页的指令集手册[5]和91页的特权架构手册[6]。相较于ARM冗长的手册[7]，更易于学习和上手。  因此，基于以上优点，本课题决定基于RISC-V架构的Soc开发嵌入式实时操作系统。目前开发操作系统使用的方式主要是C语言[8]辅以少量的汇编语言。这是因为C语言可以提供较为底层的硬件控制。诸如VxWorks、FreeRTOS等知名操作系统都是由C语言开发。但C语言也存在一些安全隐患。业界表示，约70%的安全漏洞是由于内存问题[9]。这主要是因为C语言实现内存操作的方式很容易导致缓存泄漏或者控制流攻击[10]。因此，也有部分操作系统选择使用Rust语言[11]来进行编写。Rust语言是一个专注于安全和效率的开源编程语言[12]。使其适用于系统编程的原因是它没有运行时和垃圾回收机制。此外，Rust已被证明具有类型安全性和内存安全性[13]。一些由Rust编写的嵌入式操作系统有Redox、Hubris、Tock等[14]。  考虑到C语言在当下操作系统中使用的广泛性以及本人对这两种语言的了解程度，最终决定使用C语言来完成本课题的开发工作。本课题将使用RISC-V汇编语言和C语言，通过RISC-V编译工具链中的GCC将源程序翻译成目标代码。该过程直接影响了处理器系统的整体性能[15]。所以要对GCC参数进行相应的优化。   1. 参考文献 2. 姚锡忠.嵌入式操作系统的现状及发展趋势[J].中国新技术新产品,2009(23):38. 3. 刘海丹,吴砚锋.几种常用嵌入式操作系统的对比分析[J].移动通信,2012,36(S1):78-81. 4. Waterman A,Lee Y,Patterson DA.The RISC-V Instruction Set Manual,Volume I:Base User-Level ISA.Vol.1.2011. 5. 刘畅,武延军,吴敬征等.RISC-V指令集架构研究综述[J].软件学报,2021,32(12):3992-4024.DOI:10.13328/j.cnki.jos.006490. 6. Waterman A,Asanovic K.The RISC-V Instruction Set Manual,Volume I:Unprivileged ISA.Vol.1.Si Five Inc.,2021. 7. Waterman A,Asanovic K,Hauser J.The RISC-V Instruction Set Manual,Volume II:Privileged Architecture.Si Five Inc.,2021. 8. Arm®Architecture Reference Manual.Armv8,for A-profile architecture.ARM Developer.2021.http://developer.arm.com.jit.vpn358.com/documentation/ddi0487/latest 9. Kernighan BW,Ritchie DM.The C Programming Language.Englewood Cliffs,NJ:Prentice Hall,1978. 10. Thomas, G. A Proactive Approach to More Secure Code. Available online: https://msrc-blog.microsoft.com/2019/07/16/a-proactive-approach-to-more-secure-code (accessed on 22 September 2022). 11. Walls, R.J.; Brown, N.F.; Le Baron, T.; Shue, C.A.; Okhravi, H.; Ward, B.C. Control-flow integrity for real-time embedded systems.In Proceedings of the 31st Euromicro Conference on Real-Time Systems (ECRTS 2019), Stuttgart, Germany, 9–12 July 2019. 12. Klabnik S,Nichols C.The Rust Programming Language.2018.http://doc.rust-lang.org.jit.vpn358.com/book/ 13. Mozilla Welcomes the Rust Foundation. Available online: https://blog.mozilla.org/en/mozilla/mozilla-welcomes-the-rust-foundation (accessed on 22 September 2022). 14. Balasubramanian, A.; Baranowski, M.S.; Burtsev, A.; Panda, A.; Rakamaric, Z.; Ryzhyk, L. System programming in rust: Beyondsafety. In Proceedings of the 16th Workshop on Hot Topics in Operating Systems, Whistler, BC, Canada, 7–10 May 2017;pp. 156–161. 15. Culic, Ioana et al. “A Low-Latency Optimization of a Rust-Based Secure Operating System for Embedded Devices.” Sensors (Basel, Switzerland) 22 (2022): n. pag. 16. 陈欣，吴伟，陶秋铭，等.面向C4350AL处理器的GCC移植与优化[J].计算机系统应用，2015,24(9):171-175. |

**毕 业 论 文（设计） 开 题 报 告**

|  |
| --- |
| 2．本课题要研究或解决的问题和拟采用的研究手段（途径）： |
| 一、本课题研究目标  本课题旨在在RISC-V平台上设计一个高效、可靠的嵌入式实时操作系统，以满足实时应用的需求。该操作系统应具备任务调度、中断处理、内存管理、设备驱动和通信等关键功能，同时考虑资源利用效率和响应性能。   1. 关键技术和难点 2. 内存管理和保护机制：内存管理可以通过维护链表或其他数据结构来跟踪可用内存块的分配和释放。通过链表，可以有效地管理内存的分配和回收，确保内存的高效利用和避免内存泄漏。同时，利用虚拟内存技术，实现内存的隔离和保护。通过虚拟内存管理，可以限制进程之间的内存访问，防止越界访问和非法内存操作。此外，可以利用处理器提供的硬件级别的内存保护功能，来提高安全性和可靠性。 3. 中断处理和硬件接口：中断处理可以通过操作系统在S模式下设置中断向量表，将不同类型的中断映射到相应的中断处理函数。当发生中断时，处理器会根据中断向量表中的映射关系，将中断请求定向到相应的处理函数进行处理。在硬件接口的开发中，首先需要了解目标开发板的硬件架构和相关接口信息。然后，根据硬件规格和需求，开发相应的驱动程序来与硬件进行交互。 4. 实时任务调度算法：设计高效的实时任务调度算法，考虑任务优先级、调度策略和上下文切换开销等因素，以提高实时性能。通过确定任务优先级、选择适当的调度策略和优化上下文切换，实现按时响应和满足时间约束。   三、现有的研究基础  1、完成了RISC-V架构手册《The RISC-V Instruction Set Manual》的学习和研究。  2、完成了开发板的手册《D1-H User Manual》的学习和研究。  3、完成了开发板所采用的Soc的相关手册《Xuantie\_C906\_R1S0\_User\_Manual》的学习和研究。  4、深入研究xv6操作系统的源码与运行机制。xv6是一个操作系统的实现，它基于Unix第六版（Version 6 Unix）并用于教学和研究目的。由麻省理工学院（MIT）开发，xv6为学生和研究人员提供了一个权威且可靠的平台，用于深入理解操作系统的内部工作原理和核心概念。  作为Unix第六版的简化版本，xv6保留了Unix操作系统的经典特性和设计原则，并且经过精心设计和实现，以确保其权威性和可靠性。它的开发由经验丰富的计算机科学家和工程师进行，并得到了全球操作系统研究社区的认可和支持。  xv6的代码和文档都经过仔细编写和注释，以确保其易于理解和学习。它提供了清晰的代码结构和模块化设计，使学生和研究人员能够深入研究和修改操作系统的各个组件。通过使用xv6，学生可以通过实际的代码实现和实验来加深对操作系统的理解，并且可以进行自己的研究和创新。  xv6的权威性不仅体现在其代码和文档的质量上，还体现在其广泛的应用和使用上。许多知名的大学和研究机构都使用xv6作为操作系统课程的教学工具，以及操作系统研究的实验平台。它被认为是一个可靠的操作系统实现，可以作为学习和研究操作系统的基础。  5、具有国产嵌入式实时操作系统最小系统开发经验。  四、实施方案  本课题中嵌入式实时操作系统的设计大致可以分为以下三部分：   1. 底层硬件的驱动设计 2. 内核的设计 3. 系统调用设计   其具体结构关系如下图所示：  未命名文件 (20)  图 1 系统架构图   1. 硬件驱动的设计   硬件驱动设计主要讲述以下两个核心硬件的驱动部分实现：   1. UART驱动的实现 2. Timer定时器驱动的实现   UART驱动的实现主要包含发送消息和接收消息的功能实现。其中，实现发送消息的功能，首先需要检查LSR寄存器中的第5位THRE（Transmitter Holding Register Empty）是否为1。如果为1，表示THR（Transmitter Holding Register）寄存器为空，此时可以将32位的数据写入THR中，实现数据的发送功能。  而实现接收消息的功能，则需要先通过设置IER寄存器（Interrupt Enable Register）来开启UART的中断功能。这样当有可读数据时，系统会触发一个中断。在中断处理函数中，可以通过读取LSR寄存器中的第0位DR（Data Ready）来判断数据是否已经就绪。如果DR为1，则说明数据已经就绪，此时可以直接读取RBR寄存器（Receiver Buffer Register）来获取一个32位的数据消息。  通过以上步骤，实现了UART驱动的发送消息和接收消息的功能。发送消息时，判断THR寄存器是否为空，然后写入数据；接收消息时，开启中断功能并在中断处理函数中判断数据是否就绪，然后读取RBR寄存器获取数据。这样就能完成UART通信的发送和接收操作。  未命名文件 (14)  图 2 UART驱动流程图  Timer驱动的实现主要通过配置相关寄存器来完成。以下是与Timer驱动相关的几个寄存器：   1. TMR\_IRQ\_EN\_REG：与中断使能相关的寄存器。 2. TMR\_IRQ\_STA\_REG，与中断状态相关的寄存器。 3. TMR\_CTRL\_REG，与Timer控制相关的寄存器。 4. TMR\_INTV\_VALUE\_REG，与设置时间间隔相关的寄存器。 5. TMR\_CUR\_VALUE\_REG，与当前时间相关的寄存器。   在实现Timer驱动之前，首先需要通过设置TMR\_INTV\_VALUE\_REG寄存器来设置Timer的时间周期。时间周期T可以通过以下公式计算：    其中，TMR\_CLK\_SRC表示时钟的频率，TMR\_CLK\_PRES表示时钟的分频系数。假设将TMR\_INTV\_VALUE\_REG的值设置为TMR\_CLK\_SRC/1000，并将TMR\_CLK\_PRES设置为1，这样设置完成后，T的值将等于1/1000秒，即Timer的触发周期为1微秒（1us）。设置完Timer触发周期后，只需写入TMR\_IRQ\_EN\_REG寄存器来使能时钟中断即可。  **未命名文件 (15)**  图 3 Timer驱动流程图  硬件驱动部分还包括如SPI、IIC、PLIC、SD/MMC等硬件设备的驱动编写，他们的具体实现方式与前面介绍的两种驱动的编写方式大同小异，这里就不再赘述。   1. 系统内核的设计   系统采用宏内核设计，系统内核核心部分主要由以下部分构成：   1. 任务调度模块 2. 内存管理模块 3. 虚拟内存模块 4. 中断处理模块   任务调度模块采用抢占式时间片轮转调度方式，以确保高效的任务执行和资源利用，满足实时操作系统的实时性要求。  在实时操作系统中，任务调度的关键是及时响应高优先级任务。为此，采用抢占式调度机制，即当一个优先级更高的任务准备好执行并加入任务队列时，系统会立即中断当前正在执行的任务，并切换到这个优先级更高的任务上。这种机制确保了高优先级任务能够及时得到执行，以满足系统对实时性任务的响应需求。  对于优先级相同的任务，调度模块会通过考虑任务的最早截止时间来进行优先调度。最早截止时间指的是任务必须在该时间之前完成的时间点。通过按照最早截止时间进行调度，系统可以优先处理那些紧急程度更高、需要更快完成的任务，从而提高整体系统的效率和响应性。  此外，调度模块还需要考虑非实时性任务的执行。对于这类任务，采用固定的时间片轮转调度方式。每个任务被分配一个固定大小的时间片，在该时间片内执行任务操作。当时间片用尽后，系统将切换到下一个任务，以确保任务的公平执行和资源的合理分配。这种周期性执行非实时性任务的方式可以有效地满足系统对这类任务的处理需求，同时避免长时间占用处理器资源，保证其他任务也能得到适当的执行机会。  通过采用抢占式时间片轮转调度方式，任务调度模块能够灵活地响应不同任务的优先级和截止时间要求，以最大程度地提高系统的性能和实时性能。这种调度策略综合考虑了任务的优先级和最早截止时间，并合理处理非实时性任务的周期性执行需求，以实现任务的高效调度和资源的有效利用。  未命名文件 (16)  图 4 任务调度流程图    内存管理模块通过将内存空间划分为若干个大小为4k的内存页，并以链表的形式将它们串联起来，实现对内存的高效管理。当有分配内存的请求时，模块会从内存页链表的末尾取出最后一页，并将其分配给请求方。同时，该内存页会被初始化，以确保内容的正确性和一致性。而在回收内存页时，模块会对其进行格式化，并将其重新加入内存页链表的末尾，以供后续再次分配使用。  未命名文件 (5)  图 5 内存管理形式  这种基于内存页的管理方式具有一定的灵活性和效率。通过将内存空间划分为固定大小的内存页，模块可以更加高效地管理内存资源，避免内存碎片的产生。链表的结构使得内存页的分配和回收操作可以在常数时间内完成，提高了内存管理的效率。  然而，当内存页的数量不足以满足分配请求时，系统将进入一种异常状态，称为系统panic。这种情况表示系统无法继续正常运行，因为没有足够的内存页可用于分配。系统panic是一种保护机制，它通常会触发错误处理程序或系统崩溃，以避免对无效的内存空间进行访问和操作。  内存管理模块的设计旨在提供高效、可靠的内存分配和回收机制，以满足系统对内存资源的需求。通过合理地划分和管理内存页，并采取适当的异常处理策略，模块可以有效地避免内存碎片和内存不足的问题，从而提高系统的可靠性和性能。  未命名文件 (17)  图 6 内存管理流程图  虚拟内存的实现通过使用RISC-V架构中的SV39地址转译模式，将物理地址映射到虚拟地址空间。这一过程涉及创建并使用页表，将页表的首地址写入SATP寄存器，以完成虚拟内存的启用。  在SV39地址转译模式下，虚拟地址空间被划分为多个页表级别，其中最底层的页表级别为页表页（Page Table Page，简称PTE）。每个PTE对应一块物理内存页，并存储了虚拟地址到物理地址的映射关系。通过多级页表的层次结构，可以有效地管理大型的虚拟地址空间。  在实现过程中，首先需要创建并初始化页表。通过将物理地址映射到适当的虚拟地址，页表建立了虚拟地址与物理地址之间的对应关系。然后，将页表的首地址写入系统控制寄存器SATP（Supervisor Address Translation and Protection），以告知处理器虚拟内存的起始地址和页表结构。  通过使用SV39地址转译模式和页表，虚拟内存的实现实现了虚拟地址空间与物理地址空间之间的映射，提供了更大的地址空间和更灵活的内存管理能力。这种机制可以实现内存的分页管理，并允许操作系统对内存进行动态分配和保护，提高了系统的安全性和效率。  IMG_256  图 7 Sv39虚拟地址组成    图 8 Sv39页表项组成  未命名文件 (9)  图 9 Sv39页表组成  Sv39的从虚拟地址映射到物理地址的逻辑为：   1. 通过VPN[2]中的数值找到一级页表中对应的页表项，从而定位到下一个二级页表的地址。 2. 通过VPN[1]中的数值找到先前找到的二级页表中对应的页表项，从而定位到下一个三级页表的地址。 3. 通过VPN[0]中的数值找到先前找到的三级页表中对应的页表项，从而定位到包含物理地址信息的页表项。 4. 将包含物理地址信息的页表项通过右移8位再左移12位最后再加上虚拟地址的offset值的方式定位到最终的物理地址。   从物理地址映射到虚拟地址的方式与上面所描述的相反。  未命名文件 (18)  图 10地址映射流程图  中断管理模块的实现基于RISC-V的中断控制模块。在程序启动时，系统处于机器模式（Machine Mode），这是RISC-V特权级别中的最高级别。当中断或异常发生时，默认情况下会切换到机器模式下执行，但这可能存在安全问题。为了解决这个问题，本课题希望将控制权转移到监管者模式（Supervisor Mode）下执行。通过设置mdeleg（Machine Exception Delegation）寄存器和mideleg（Machine Interrupt Delegation）寄存器，可以实现这个目标。    图 11 medeleg寄存器    图 12 mideleg寄存器  其中，mdeleg寄存器和mideleg寄存器中的每一位分别对应一个异常或中断，将对应位置设置为1表示将该异常或中断转移到监管者模式下执行。  在完成异常和中断的委派设置后，还需要开启监管者模式下的异常中断（exception interrupt）、时钟中断（timer interrupt）和软件中断（software interrupt）。通过设置sie寄存器中的SEIE、STIE、SSIE位来实现这一目的。  图 13 sie寄存器  完成上述设置后，还需要设置sstatus寄存器中的SIE位，以启用监管者模式下的中断监听。    图 14 sstatus寄存器  当一个中断信号触发后，首先需要判断该中断是由异常还是设备触发的。通过读取scause寄存器可以获取相关信息来进行判断。    图 15 scause寄存器  当中断为设备触发的中断信号时，scause寄存器的Interrupt位会被置为1，表示中断是由设备触发的。如果中断是异常引起的，则Interrupt位为0。  通过读取Exception Code字段，可以进一步了解中断的具体类型和原因。Exception Code提供了更详细的信息，包括不同类型的异常和设备中断的编码。    图 16 Exception Code  当中断类型为时钟中断时，该中断由CLINT（Core Local Interruptor）处理。而当中断类型为外部中断时，该中断由PLIC（Platform-Level Interrupt Controller）处理。  当PLIC处理中断时，首先会跳转到stvec（Supervisor Trap Vector Base Address）寄存器所指向的中断向量处进行处理。stvec寄存器用于存储中断向量表的基地址，其中包含了处理不同中断类型的中断处理程序的入口点。    图 17 stvec寄存器  PLIC（Platform-Level Interrupt Controller）通常会提供一组寄存器，其中包含了外部中断的相关信息，包括中断号、中断使能等。通过读取这些寄存器，可以获取当前触发的外部中断的中断号。  一旦获取到中断号，可以根据中断号的映射关系，将控制权转移到相应的中断处理函数，进行中断的具体处理操作。中断处理函数可以是预先定义的函数，用于处理特定中断类型的逻辑。  未命名文件 (19)  图 18 中断处理流程图   1. 可行性分析 2. RISC-V指令集架构在过去几年中取得了显著的发展势头。由于其开源性质和灵活性，吸引了众多开发者和研究者的关注和参与。在全球范围内，涌现出了许多RISC-V社区、论坛和开发平台，提供了丰富的相关资料、教程和技术支持。开发者可以通过这些平台获取最新的开发工具、编译器、调试器和文档。 3. 随着RISC-V的发展，市场上出现了众多厂商提供的RISC-V架构的中央处理器（CPU）。这些CPU产品涵盖了不同的性能级别、功耗要求和应用场景。从嵌入式系统到高性能计算，从物联网设备到服务器，开发者可以根据项目需求选择适合的RISC-V CPU。这种多样性为实时操作系统的设计和实现提供了更大的灵活性和可选择性。 4. RISC-V的发展受益于强大的开源社区。在这个社区中，开发者可以分享他们的经验、解决方案和代码，并从其他人的工作中获得灵感和帮助。开源社区的协作精神鼓励了知识的共享和创新的加速，对于实时操作系统设计与实现项目来说，可以获得更多的资源和支持。 5. 实时操作系统在诸多领域中都有广泛的应用需求，并且这些领域的发展趋势显示出对实时操作系统的需求将持续增长。例如，工业自动化领域需要高性能和可靠性的实时操作系统来管理和控制生产设备；物联网领域需要实时操作系统来处理大量的数据和实时通信；嵌入式系统领域需要轻量级且低功耗的实时操作系统来支持各种终端设备。这些市场需求提供了实时操作系统设计与实现项目的发展机会。   5、RISC-V架构的一个显著优势是其可扩展性和定制化能力。实时操作系统设计者可以根据具体应用的需求，选择和定制合适的指令集扩展和硬件功能，以提高系统的性能和效能。这种灵活性使得实时操作系统能够更好地适应不同的应用场景，并满足特定的实时要求。 |
|  |

**毕 业 论 文（设计） 开 题 报 告**

|  |
| --- |
| **指导教师意见：** |
| 1．对“文献综述”的评语：  2．对本课题的深度、广度及工作量的意见和对论文（设计）结果的预测：  3.是否同意开题：□同意 □不同意  　　　　　　　　　　　　　　　　指导教师：  202 年 月 日 |
| 所在专业审查意见：  　　　　　　　　　　　　　　　　负责人：  202 年 月 日 |