

基于 PowerPC P2020 的嵌入式多核通信研究与实现

杨小兵, 吴德松

(中国电子科技集团公司 第三十研究所, 成都 610041)

摘要: 为满足现代嵌入式系统对高性能、实时性和可靠性的需求, 多核处理器技术成为解决复杂任务处理的关键手段; 基于 PowerPC P2020 双核处理器及 vxWorks6.9 实时操作系统, 对多核通信的实现方法进行了研究, 对 AMP、SMP 和 BMP 三种多核架构特点进行了分析, 设计了基于 SMP 模式的多核系统架构; 研究重点包括多核启动流程、任务调度策略以及核间通信机制的设计与实现; 经试验测试表明, 通过合理的任务分配、负载均衡调度以及高效的 IPI 和共享内存通信机制, P2020 双核处理器能够显著提升系统性能, 满足了实时性和稳定性的要求, 为嵌入式多核系统的设计与优化提供了理论依据和实践参考。

关键词: 多核通信; PowerPC P2020; vxWorks; SMP; 核间通信

Research and Implementation on Embedded Multi-Core Communication Based on PowerPC P2020

YANG Xiaobing, WU Desong

(The 30th Institute of CETC, Chengdu 610041, China)

Abstract: In order to meet the requirements of high performance, real-time, and reliability in modern embedded systems, it is a key means for multi-core processor technology to solve complex task processing. Based on PowerPC P2020 dual core processor and vxWorks6.9 real-time operating system, this paper studies the implementation method of multi-core communication, analyzes the characteristics of AMP, SMP, and BMP multi-core architectures, designs a multi-core system architecture based on SMP mode, focuses on the design and implementation of multi-core startup processes, task scheduling strategies, and inter core communication mechanisms. Experimental results show that through reasonable task allocation, load balancing scheduling, and efficient IPI and shared memory communication mechanisms, the P2020 dual core processor can significantly improve system performance, meet real-time and stability requirements, and provide theoretical basis and practical reference for the design and optimization of embedded multi-core systems.

Keywords: multi-core communication; PowerPC P2020; VxWorks; SMP; inter-core communication

0 引言

近年来, 随着信号处理技术的发展, 特别是人工智能与信号处理融合、大数据与信号处理结合、多模态信号处理技术以及实时信号处理技术的提升已成为信号处理技术发展的新趋势。在人工智能与信号处理融合领域, 人工智能可实现信号调制和编码优化, 如使用深度学习进行调制识别, 提高频谱效率和识别精度, 从而减少重传, 提升数据传输速率。多核信号处理系统可以并

行处理数据, 加快传输速度, 进一步提高了数据传输速率。人工智能具备实时决策能力, 对多核系统的资源进行动态分配。多核并行处理可以加速信号处理任务, 这样降低了端到端延迟, 提高了系统的实时性。可靠性方面, 人工智能可进行错误检测和纠正, 如使用机器学习模型预测网络拥塞以避免数据丢失。多核系统的冗余设计和负载均衡增强了系统的容错能力, 进而提高系统的可靠性。在大数据与信号处理结合领域, 大数据除了具有大量、高速、多样、价值、真实性五大特点之外, 还

收稿日期:2025-02-28; 修回日期:2025-04-01。

作者简介:杨小兵(1985-),男,硕士,工程师。

引用格式:杨小兵,吴德松. 基于 PowerPC P2020 的嵌入式多核通信研究与实现[J]. 计算机测量与控制, 2025, 33(7): 280-286, 294.

具有数据复杂性、数据实时性、数据来源多样性、数据易变性的特点。多核信号处理系统能够实时处理不同类型的海量数据,从而提取有用信息,确保数据的可靠性。在多模态信号处理技术领域,多模态信号处理涉及音频、视频、图像、文本等多种数据形式的融合处理。数据传输速率方面,多模态数据通常数据量大。多核通信可实现高效的压缩算法,减少传输数据量。同时,多核并行处理可以加速数据的编解码,提升传输速率。实时性方面,多模态处理需要低延迟,多核架构可以分配不同任务到不同核心,减少处理延迟。此外,中断驱动机制和快速消息传递也能提升实时性。可靠性方面,多模态数据传输需要抗干扰和纠错。多核系统可以采用冗余传输,如不同核心同时处理和传输数据,提高容错能力。另外,错误检测和自愈技术,能减少传输错误,提升可靠性。在实时信号处理领域,实时信号处理强调快速、高效地处理数据,通常用于需要即时响应的系统,多核通信场景下,多个处理器核心并行处理数据,增强了系统实时性。数据传输速率方面,多核系统采用任务分割、流水线处理、内存优化等技术可以通过并行处理提升数据传输速率。实时性方面,多核系统中的任务调度、中断管理、优先级分配等技术可以降低延迟和保证确定性处理。可靠性方面,多核系统下的冗余设计、错误检测与纠正技术可提高系统可靠性,避免数据丢失或处理错误。

信号处理技术的算法实现依托嵌入式系统平台,高效的平台最大限度地对信号处理算法普及和推广起到积极作用,二者相辅相成。新技术的发展对嵌入式系统功效、可靠性、成本、体积和功耗提出了更高要求,如何实现嵌入式系统高性能、低功耗、小型化、低成本要求成为亟待解决的问题^[1-2]。由于功耗、成本、体积等因素的影响,仅仅通过提高CPU主频的方法来实现系统性能的提升,性能提升能力往往受到限制。而单芯片多核处理器能够在同等CPU主频下提升处理器的性能,进而提升系统性能,从而规避因功耗、成本、体积带来的问题。目前多核处理器因其高性能、高集成度、低功耗、低成本等诸多优势,广泛应用于嵌入式系统。多核处理器即在一个单芯片上集成两个或多个处理器核心,每个核心执行各自的程序代码,并通过核间通信机制实现核心间的数据共享和同步^[3-4]。

多核处理器的每个核心可以同时单独读取和执行程序指令,实现程序代码的并行执行,具有较高的线程级并行性,提高了处理器的计算性能和效率,适用于快速地处理大计算量的任务。另外,多核处理器还有控制逻辑简单、易于配置、功耗低、通信延迟低等优点。多核处理器的缺点体现在:在硬件方面,由于多核处理器在一个芯片上集成了多个核心,核心之间连接紧密,对芯

片设计、布线和制造要求较高;在软件方面,多核处理器在任务调度、核间通信等方面使用的软件编程技术较为复杂和困难。

1 vxWorks 和 PowerPC P2020

vxWorks 操作系统作为一款广泛应用于嵌入式领域的实时操作系统,其模块化设计和多任务支持能力使其在多核处理器环境中表现出色。vxWorks6.9 引入了对称多处理(SMP)支持,能够有效管理多核处理器的资源分配和任务调度。vxWorks 在中断响应时间上可达到微秒级别,任务切换时间仅为1.2微秒,这使得其在实时性要求极高的场景中具有显著优势。在航空航天领域,vxWorks 被用于控制卫星的姿态调整系统,其高效的中断管理和任务调度能力确保了系统在复杂环境下的稳定运行^[5]。

PowerPC P2020 处理器架构如图1所示,PowerPC P2020 处理器是 Freescale 推出的一款双核嵌入式处理器,采用 e500v2 内核架构,主频最高可达1.2GHz。P2020 具有 DDR2/DDR3、OpenPIC 以及 DMA 等硬件外设,为多核系统通信的实现提供了基础。DDR2/DDR3 为内存类型,在多核系统中,它们作为共享内存的一部分,供多个核心访问。多核通信通常依赖于共享内存来交换数据,而内存的性能(如带宽、延迟)直接影响通信效率。在多核环境下,DDR2/DDR3 能更快地传输数据,减少核心间的等待时间。OpenPIC 为开放可编程中断控制器,在多核通信过程中主要负责中断管理、核间通信支持及多处理器协作。DMA 为直接内存访问,在多核通信过程中,作为硬件加速模块,通过绕过CPU直接进行数据传输,显著提升核间数据交互效率。三者在多核框架中协同运作,DMA 支持外设到内存、内存到外设、内存到内存、外设到外设的直接数据传输,无需CPU干预,DMA 完成数据传输后触发中断(OpenPIC 的核间中断 IPI),通知目标核处理数据。例如,Core0 通过 DMA 将任务结果写入共享内存 DDR2/DDR3 后,发送 IPI 中断唤醒 Core1。

P2020 支持 SMP 和非对称多处理(AMP)两种模式,能够根据应用需求灵活分配任务资源。在 SMP 模式下,双核共享内存和 I/O 资源,通过统一的调度策略实现负载均衡;而在 AMP 模式下,每个核可以运行独立的操作系统或任务,适用于异构计算场景。P2020 的 AMP 模式被用于同时运行实时控制任务和非实时数据处理任务,显著提升了系统的整体性能^[6-7]。

vxWorks 与 PowerPC P2020 的结合在多核处理中展现强大的协同效应。vxWorks 的 SMP 调度算法能够将任务线程动态迁移到负载较轻的核心上,从而优化资源利用率。根据 NXP 公司测试数据,在 P2020 双核处

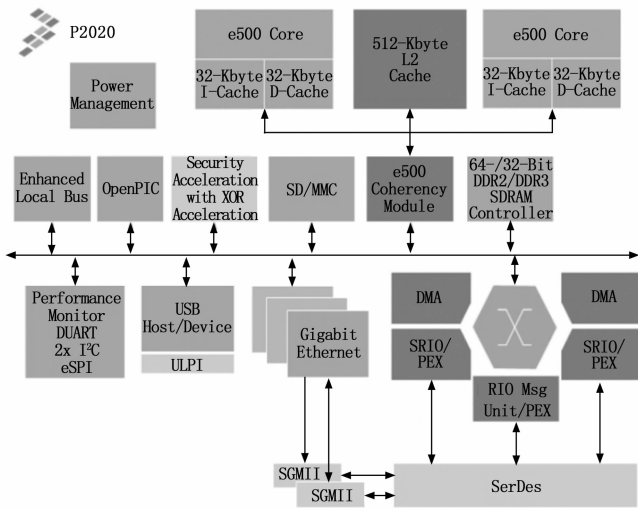


图 1 PowerPC P2020 处理器架构

理器上运行 vxWorks 时，系统的整体吞吐量比单核模式提升了约 75%。vxWorks 的核间通信机制（如共享内存和核间中断）与 P2020 的硬件特性高度契合，确保了多核间的高效数据交换^[8]。

vxWorks 与 P2020 的结合也面临一些挑战。在 AMP 模式下，核间通信的延迟和带宽限制可能成为性能瓶颈。在 P2020 的 AMP 模式下，核间通信的延迟约为 5 微秒，而在 SMP 模式下仅为 1 微秒。因此，在实际应用中，需要根据具体需求选择合适的多核模式，并通过优化任务分配和通信机制来提升系统性能。

2 多核处理器结构

从软件角度分，多核处理器平台的操作系统体系有 AMP、SMP、边界多处理结构（BMP）3 种。这些结构在任务分配、资源管理和通信机制上存在显著差异，适用于不同的应用场景。

对于 AMP 架构，如图 2 所示，每个处理器核心运行独立的操作系统实例，核心之间通过共享内存或消息传递机制进行通信。AMP 架构在资源利用率、任务调度灵活性、通信效率、可靠性等方面优缺点如下。

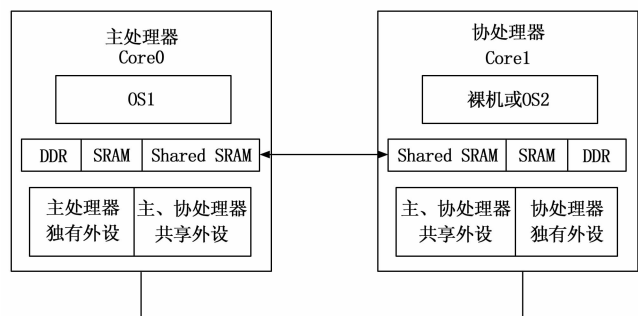


图 2 AMP 结构

资源利用率方面，AMP 架构通过静态任务分配避免资源竞争，专用核资源利用率可达 90% 以上，但核间资源无法动态共享，整体资源利用率通常低于 60%；任务调度灵活性方面，主核集中调度，任务确定性强（实时控制系统中，周期任务抖动 $< \pm 5 \mu\text{s}$ ），但从核功能固化，动态负载调整能力差；通信效率方面，AMP 架构片间通信依赖外部总线，延迟 $2 \sim 5 \mu\text{s}$ ，带宽 10 Mbps；可靠性方面，AMP 架构故障隔离性强（从核故障不影响主核），平均故障恢复时间（MTTR） $< 100 \text{ ms}$ ，但缺乏冗余机制，单点故障概率较高^[9]。

AMP 架构的优势在于任务隔离性强，适合需要高可靠性和容错能力的场景。在航空航天领域，AMP 架构被广泛应用于飞行控制系统，其中一个核心负责飞行控制，另一个核心负责故障检测和恢复。但这种架构资源利用率较低，核心之间的通信开销较大，尤其是在需要频繁数据交换的场景中。

相对 SMP 架构，如图 3 所示，所有核心共享同一操作系统实例和内存资源，任务由操作系统统一调度。SMP 架构在资源利用率、任务调度灵活性、通信效率、可靠性等方面优缺点如下。

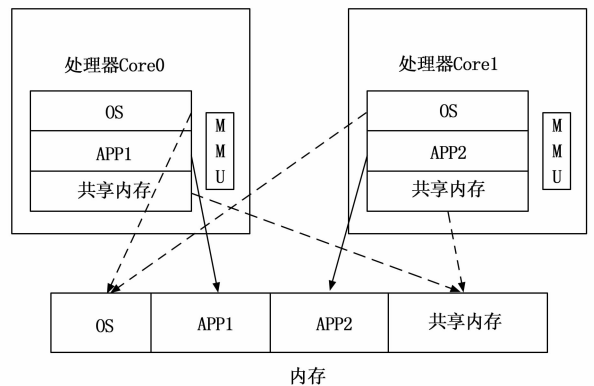


图 3 SMP 结构

资源利用率方面，SMP 架构共享内存和外设，资源利用率可达 85% 以上，但高速缓存一致性协议会导致 20%~30% 的访存开销；在任务调度灵活性方面，全局调度器支持动态任务迁移，但上下文切换开销较大；通信效率方面，SMP 架构共享内存带宽达 100 GB/s，核间通信延迟 $< 50 \text{ ns}$ ；可靠性方面，SMP 架构通过热重启和纠错编码（如 ECC 内存）实现 99.999% 可用性，但共享资源会导致故障传播风险。

SMP 架构的优势在于资源利用率高，任务调度灵活，适合计算密集型任务。通过对双核的每个核心任务进行分工，Core0 和 Core1 运行任务各有侧重，由 vxWorks 的调度策略实现双核高效并行处理。但这种架构任务之间的资源竞争可能导致性能瓶颈，尤其是在核心数量较多的情况下^[10]。

对于架构而言, 边界多处理结构 (BMP) 介于 AMP 和 SMP 之间, 核心之间部分共享资源, 同时保留一定的独立性。BMP 架构在资源利用率、任务调度灵活性、通信效率、可靠性等方面优缺点如下。

资源利用率方面, BMP 架构块内资源共享, 块内利用率 92%, 整体利用率 75%, 但跨块通信需通过片间总线, 引入 10~20 ns 延迟; 任务调度灵活性方面, 块内自主调度 (如每个块运行独立 RTOS), 块间通过邮箱通信协作, 但跨块任务迁移需要复杂同步机制, 延迟达微秒级; 通信效率方面, BMP 架构块内共享内存带宽 25 GB/s, 跨块通过 HyperTransport (HT) 总线, 延迟 80 ns; 可靠性方面, SMP 架构块级冗余设计 (如双块备份), 系统可靠性达 99.999 9%, 但故障块需整体替换, 维修成本较高^[11]。

BMP 架构的优势在于兼顾了资源利用率和任务隔离性, 适合需要平衡计算性能和可靠性的场景。但这种架构设计和调试复杂度较高, 需要精确的资源分配和通信机制设计。

在实际应用中, AMP、SMP 和 BMP 的选择取决于具体的性能需求和系统约束。AMP 架构常被用于实现高可靠性的任务隔离, 而 SMP 架构则被用于实现高效的数据处理。BMP 架构通过部分资源共享实现实时性和可靠性的平衡^[12]。

3 多核模块 SMP 架构实现

3.1 多核系统架构设计

本设计中根据信号处理和特点, 采用基于 SMP 结构的多核架构设计。如图 4 所示, Core0 和 Core1 分工协作, Core0 主要用于控制任务, 包括指令控制、设备状态管理、数据管理及任务调度的计算; Core1 专注于采集数据的运算处理, 如信号滤波、数据压缩和算法运算等, 最终处理数据结果由该核打包发

送。Core0 和 Core1 在实时操作系统 vxWorks 统一控制下对双核的任务进行动态调度, 同时双核基于 P2020 的 IPI 核间中断机制, 通过共享内存的方式完成双核间的通信, 保证双核高效、稳定运行。

通过上述设计, 基于 PowerPC P2020 的多核系统, 双核协同工作不仅可以提高任务处理效率, 还通过高效的核间通信和动态资源管理机制, 确保系统的实时性和可靠性。

3.2 多核启动流程

vxWorks 操作系统的启动流程是实现系统稳定运行的关键环节, 如图 5 所示。vxWorks 通过创建 ROM 镜像引导目标系统, 并下载运行应用程序, 确保多核系统的高效启动和稳定运行。具体而言, vxWorks 的启动流程分为以下几个阶段: 引导加载程序 (Bootloader) 初始化、内核加载与初始化、多核同步与任务分配^[13-14]。

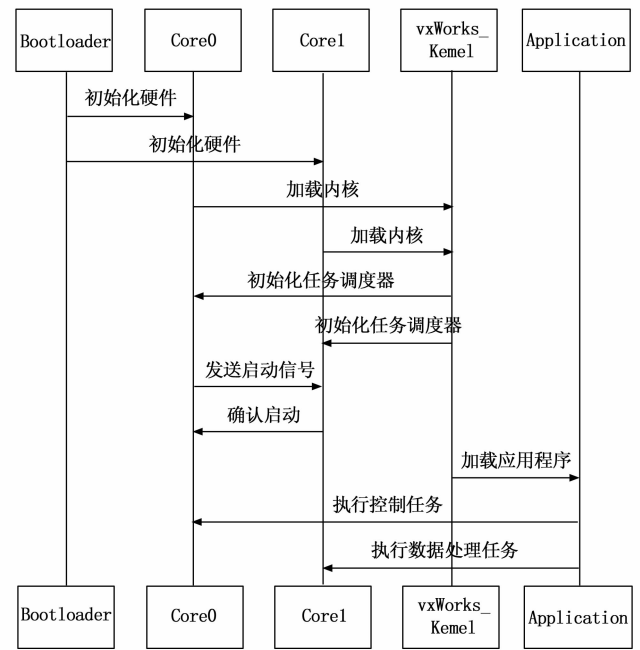


图 5 多核启动流程

在系统上电后, Bootloader 负责初始化硬件环境并加载 vxWorks 内核。Bootloader 通过读取 ROM 中的镜像文件, 将内核代码加载到内存中, 并完成必要的硬件配置, 如设置内存映射、初始化中断控制器和配置时钟频率。PowerPC P2020 的 Bootloader 设计需要特别关注双核同步问题, 以确保两个核心能够同时进入内核初始化阶段。

为实现双核同步, Bootloader 设计需从硬件初始化、内存划分、引导流程控制及同步机制设计 4 个方面入手^[15]。硬件初始化方面, P2020 的核心时钟、DDR 内存时钟等参数通过硬件上下拉电阻固定配置, 需在电

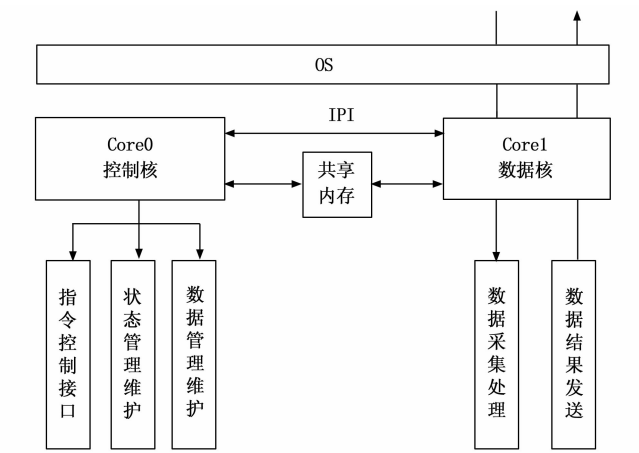


图 4 SPM 多核架构设计

路设计阶段确定, Bootloader 需确保初始化时正确识别并应用这些配置, 避免因时钟或内存参数错误导致双核启动异常; 在内存划分方面, 需要进行独立内存空间分配, 将 Core0 和 Core1 的程序分别加载到 FLASH 或 SDRAM 的独立区域。Core0 程序加载至 0xFFFF0000 (1MB Bootrom 区), Core1 程序加载至 0xF2000000 (用户存储区), 避免地址冲突。同时还需考虑共享内存区域设置, 在 SDRAM 中开辟共享内存 (0x3C000000), 用于双核间数据交换与同步标志, Bootloader 需初始化该区域, 并通过内存屏障指令确保数据一致性; 在引导流程控制方面, 包括主从核启动顺序和启动状态同步两方面, 主从核启动顺序为: Core0 (主核) 先启动, 完成自身初始化后, 加载 Core1 的程序至其内存空间一配置 Core1 的向量表起始地址—解除 Core1 的复位状态, 触发其执行。通过在共享内存中设置启动标志, Core0 在启动后写入标志, Core1 通过轮询该标志确认启动时机。结合硬件信号量或中断机制, 确保 Core1 在 Core0 准备就绪后开始执行; 同步机制设计方面, 采用硬件信号量、中断与事件通知或缓存一致性管理来实现双核的同步。

Bootloader 完成初始化后, vxWorks 内核开始加载并初始化。内核初始化阶段完成内存管理、任务调度器、中断处理机制等核心模块的配置。在 SMP 模式下, vxWorks 会为每个核心分配独立的内核栈, 初始化核间通信机制为共享内存和核间中断。vxWorks 在 P2020 平台上的内核初始化时间通常控制在 200 ms 以内, 这得益于其高效的硬件抽象层设计^[16]。

在多核同步阶段, vxWorks 通过核间中断机制确保所有核心完成初始化并进入就绪状态。Core0 作为主核心, 负责协调其他核心的启动流程。Core1 在接收到 Core0 的启动信号后, 开始执行预定义的任务分配逻辑。这种主从式启动模式在 P2020 平台上表现出较高的可靠性, 核间同步延迟通常低于 10 μ s。

vxWorks 通过下载应用程序镜像完成系统的最终配置。应用程序镜像通常存储在外部存储器中。vxWorks 的文件系统模块负责加载镜像文件, 并将其映射到内存中执行。P2020 平台在加载 10 MB 大小的应用程序镜像时, 平均耗时约为 500 ms, 这一性能足以满足大多数实时嵌入式系统的需求^[17]。

3.3 多核任务调度设计

vxWorks 操作系统的 SMP 对称多处理器结构调度, 就是将执行任务的线程 (进程) 迁移到合适的核心上, 并保持各个核心的负载均衡。SMP 的调度算法通常可以分为 3 种^[18-19]。

在需要指定某核心作专一处理时。创建线程时, 线

程默认 Core0 运行; 也可以指定亲和性, 线程指定挂载到指定核心运行。如当前任务执行队列有任务 1、2、3、4、5、6, 线程默认 Core0 运行, 任务 1、2 指定到 Core0 运行, 任务 4、5 指定到 Core1 运行, 任务 2、6 默认在 Core0 运行。这种方式可以规定某个核心专注的做某一件事或某一类事, 但 Core0 的负载会很大, 它需要调度其他核心不调度的任务。

当所有任务复杂程度相差不大, 对每个核心任务均分时。创建线程 (进程) 时, 指定核心的任务只能在该核心上运行; 没有指定核心的任务, 操作系统通过判断每个核心的任务数, 将该任务放在任务数最少的核心中。如当前任务执行队列有任务 1、2、3、4、5、6、7、8, 任务 1、3 指定到 Core0 运行, 任务 4、5 指定到 Core1 运行。任务 2、7、6、8 系统根据每个核心任务数, 均分到 Core0 和 Core1 运行。任务 2、7 分配到 Core0, 任务 6、8 分配到 Core1, 从而实现两核的任务均分。这种方式通过将任务平分给每个核心, 每个核心的负载会相对均衡, 但可能存在某个核心分配的任务轻重有差异, 导致核心任务轻的更加容易进入空闲状态, 核心任务重的, 可能会一直处于忙碌状态, 这样也会导致每个核心的负载不均衡。

在不需要关心任务具体运行到哪个核心时。创建线程 (进程) 时, 指定核心的任务只能在该核心上运行; 没有指定核心的任务, 将该任务挂载到一个总任务队列中, 当某个核心调度空闲时, 就从总任务队列中获取一个任务运行, 运行完毕之后归还给总任务队列。如当前任务执行队列有任务 1、2、3、4、5、6、7、8, 任务 1、3 指定到 Core0 运行, 任务 4、5 指定到 Core1 运行, 任务 7、2、6、8 不指定内核, 挂载总任务队列; 当 Core0 调度空闲时, 任务 8 从总任务队列中取出, 放入 Core0 运行, 此时任务 1、任务 3 挂起, 任务 8 运行结束后归还总任务队列; 同理, 任务 6 在 Core1 调度空闲时挂载到 Core1 运行。这种方式可根据每个核心的负载, 均分整个系统的任务调度, 提高了每个核心的利用率, 但调度算法较复杂。

以上 3 种调度算法, 根据不同的调度适用于不同的场景, 需要根据实际需求选择相应的调度算法。本设计选择调度算法 2 实现多核任务调度设计。

3.4 多核核间通信设计

IPI 经常被用于同一个处理器内部不同内核之间的通信, P2020 可以通过核间中断和消息寄存器来完成两个 e500v2 内核之间的通信。考虑到本系统多任务间需要频繁进行大量数据交换, 共享内存通过将同一块地址空间的内存进行共享, 能够满足数据交互需求, 提高程序的响应实时性和运行可靠性。本设计采用核间中断加

共享内存的核间通信设计, 其核间通信流程如图 6 所示。

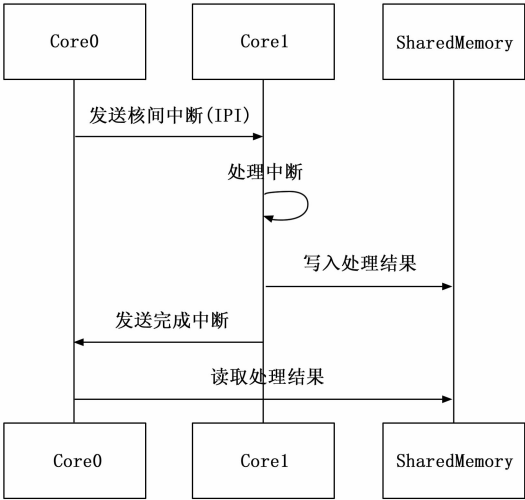


图 6 核间通信流程

在双核数据交互实现上, 如图 7 所示, 共享内存块的读写状态通过设置同步标志 flag 置 1, 置 0, 来实现 Core0 和 Core1 双方通信的同步。同时, 在多任务数据交换中, 同时运行的多个任务可能需要同一种数据资源, 为了保证共享数据操作的完整性, 通过 vxWorks 提供的互斥信号量 semTake (semMutex, WAIT _

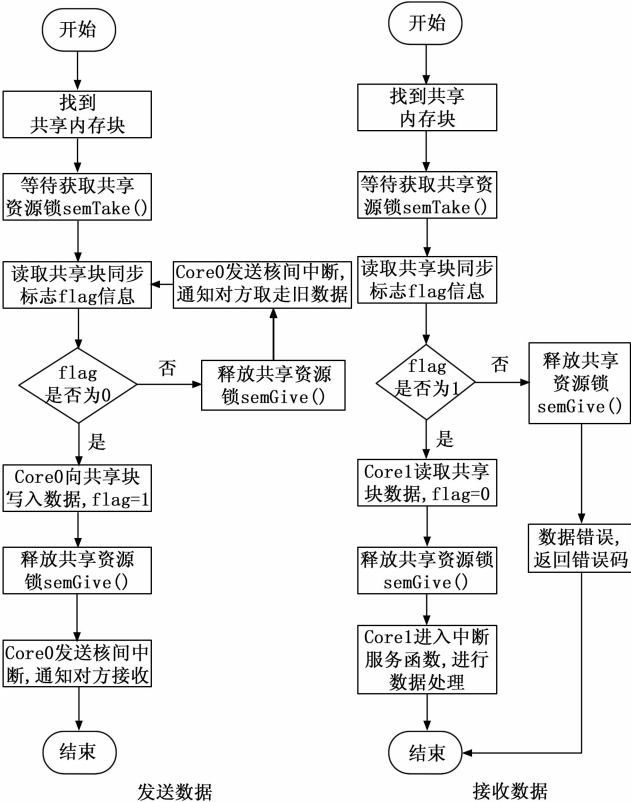


图 7 双核数据交互流程

FOREVER) 和 semGive (semMutex) 来实现对共享资源的保护。Core0 发送数据时, 首先查找共享内存块; 然后, 获取共享资源互斥锁, 循环检查 flag 状态信息, 直到共享内存块可写 (flag=0); 接着, 向共享内存块中写入消息数据, 设置共享内存块不可写 (flag=1), 释放共享资源互斥锁; 最后, 向 Core1 发送核间中断, 通知其接收消息。Core1 接收数据时, Core1 响应 Core0 中断, 首先找到共享内存块位置; 然后, 获取共享资源互斥锁, 循环检查 flag 状态信息, 直到共享内存块可读 (flag=1); 接着, 从共享内存块中读出消息数据, 设置共享内存块可写 (flag=0), 释放共享资源互斥锁; 最后, 在中断服务程序中完成数据处理^[20-21]。

4 多核系统实际运用

在某信号处理设备的实际应用中, 如图 8 所示, 基于 PowerPC P2020 双核处理器的 SMP 架构被成功部署, 显著提升了系统的实时性和计算效率。该设备主要用于通信信号处理, 要求高吞吐量和低延迟的数据处理能力。通过 vxWorks 操作系统的调度策略, Core0 被分配为控制任务核心, 负责命令控制、设备状态管理维护、数据维护等; Core1 则专注于数据处理任务, 包括信号的滤波、解调和目标检测。这种分工设计充分利用双核处理器的并行计算能力, 同时避免了资源竞争和任务冲突, 双核通信结果如图 9 所示。

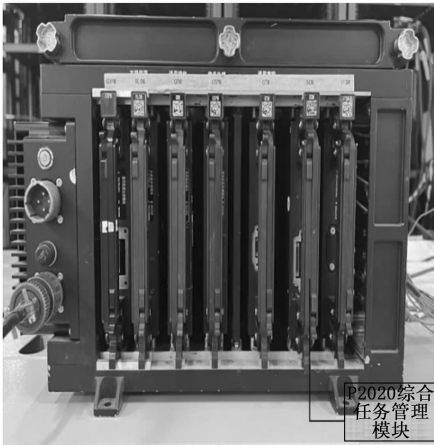


图 8 信号处理设备

在系统启动阶段, vxWorks 通过 ROM 镜像引导双核同步启动, 确保 Core0 和 Core1 在统一的内存空间和调度策略下运行。Core0 在完成系统初始化后, 通过核间中断机制通知 Core1 开始数据处理任务。核间通信采用共享内存方式, 数据缓冲区被映射到双核的同一地址空间, 避免了数据拷贝带来的额外开销。根据实际测试数据, 双核间的通信延迟仅为 1.2 微秒, 远低于传统消息传递机制的 10 微秒延迟。

```

: core0应用程序AppRun0初始化成功!
: Core1应用程序AppRun1初始化成功!
: core0向core1发送数据处理启动命令!
: core0收到core1数据处理启动命令回复!
: 数据处理任务1指定core1处理!
: core1开始数据处理任务1!
: core1将数据处理任务1数据结果发回core0!
: core0开始数据处理!
: Core0将数据处理任务1数据结果进一步处理后发回core1!
: 数据处理任务3指定core1处理!
: core1将数据处理任务3数据结果发回core0!
: core0开始数据处理!
: Core0将数据处理任务3数据结果进一步处理后发回core1!
: 数据处理任务4指定core1处理!
: core1开始数据处理任务4!
: core1将数据处理任务4数据结果发回core0!
: core0开始数据处理!
: Core0将数据处理任务4数据结果进一步处理后发回core1!
: 数据处理任务6指定core1处理!
: core1开始数据处理任务6!
: core1将数据处理任务6数据结果发回core0!
: core0开始数据处理!
: Core0将数据处理任务6数据结果进一步处理后发回core1!
: core0开始数据处理任务2!
: Core0将数据处理任务2数据结果发回core1!
: Core0开始数据处理任务5!
: Core0将数据处理任务5数据结果发回core1!
: Core0开始数据处理任务7!
: Core0将数据处理任务7数据结果发回core1!
: Core0向Core1发送数据处理停止命令!

```

图9 core0 和 core1 通信结果

在任务调度方面, vxWorks 的 SMP 调度算法根据任务优先级和负载情况动态分配计算资源。当 Core1 的 CPU 利用率达到 80% 时, 调度器会将部分低优先级任务迁移到 Core0, 以保持负载均衡。在动态调度策略下, 系统的整体 CPU 利用率始终保持在 90% 以上, 而单核模式下的利用率仅为 65%。vxWorks 的实时任务调度机制确保了关键任务的响应时间不超过 50 微秒, 满足了通信信号处理的实时性要求。

为了验证系统的任务执行时间, 设计对比试验, 在单核模式下, 系统处理 1 000 个通信信号的平均时间为 12.3 毫秒; 而在双核 SMP 模式下, 处理时间降至 6.8 毫秒, 任务执行时间大幅缩短, 性能提升了近 45%。这一结果与理论预期相符, 证明了 SMP 架构在高并发任务处理中的优势。通过分析系统日志, 双核间的任务迁移频率为每秒 120 次, 任务切换开销仅为 0.8 微秒, 进一步验证了 vxWorks 调度算法的高效性。

在设备实际运行中, 当 Core1 处理大量高优先级任务时, Core0 的控制任务可能被延迟。为了解决这一问题, 采用优化任务优先级分配策略, 将 Core0 的关键控制任务设置为最高优先级, 确保其始终能够及时响应外部事件。通过这一优化, 系统的控制任务响应时间从原来的 200 微秒降低至 50 微秒, 显著提高了系统的可靠性。

从硬件资源利用的角度来看, P2020 双核处理器的共享 L2 缓存设计为双核通信提供了硬件支持。L2 缓存的命中率达到 95%, 有效减少了内存访问延迟。通过

优化数据对齐和缓存预取策略, Core1 的数据处理吞吐量提升了 20%, 进一步释放了计算潜力。

在系统稳定性测试中, 通过模拟高负载和异常情况下的运行表现。在连续运行 72 小时的测试中, 系统未出现任务死锁或资源耗尽的情况, CPU 利用率波动范围控制在 $\pm 5\%$ 以内。这一结果表明, 基于 vxWorks 和 P2020 的双核 SMP 架构在长期运行中具有较高的稳定性和可靠性。

通过上述实际应用可看出, 基于 PowerPC P2020 的双核 SMP 架构在信号处理领域具有显著优势, 其高效的核间通信机制、灵活的调度策略以及稳定的运行表现, 为高性能嵌入式系统设计提供了重要参考。随着多核处理器核心数量的增加, 任务分配和资源管理将面临严峻挑战, 如何进一步优化任务分配和资源管理将成为未来亟待解决的问题和重要研究方向。

5 结束语

本文在 PowerPC P2020 双核多处理器架构基础上, 对多核系统的 3 种结构的工作方式进行了阐述。结合信号处理和控制特点, 对控制和数据任务分工, 分别运行于控制核和数据核, 设计并实现了一种基于 SMP 模式的多核通信方法。通过采用合理的多核任务调度策略和以核间中断加共享内存块的核间通信方式, 使任务高效并行运行, 充分发挥了多核处理器的性能, 保证了通信数据的有效性和实时性, 实现了多核处理器的高效、稳定通信。在以后研究中, 双核通信设计思路可以扩展到四核、八核以及更多核, 以满足日益复杂的应用场景需求, 对嵌入式系统在信号处理领域的功能扩展与优化具有一定借鉴意义。

参考文献:

- [1] 石伟, 张明, 郭御风, 等. 实时微处理器体系结构综述 [J]. 计算机工程与科学, 2015, 37 (5): 857-864.
- [2] 白俊峰, 邓祖朴. 多核系统的软件开发方法 [J]. 计算机工程, 2010, 23 (12): 56-57.
- [3] FREESCALE SEMICONDUCTOR CORP. P2020 QorIQ integrated processor reference manual [S]. America: Freescale Semiconductor Corp, 2011: 112-132.
- [4] 戴鸿君. 基于异构多核体系与组件化软件的嵌入式系统研究 [D]. 杭州: 浙江大学, 2007.
- [5] 杨国芳. 多核处理器核间通信技术研究 [D]. 哈尔滨: 哈尔滨工程大学, 2011.
- [6] 何翔, 任晓瑞. 支持多核的嵌入式操作系统关键技术研究 [J]. 航空计算技术, 2013, 43 (4): 86-90.
- [7] 魏智伟. 多核 DSP 间基于 SRIO 数据传输的设计与实现 [J]. 微型机与应用, 2017, 36 (4): 36-39.

(下转第 294 页)