

异构处理器多操作系统协同技术研究^①

冯瑞青, 张 激, 赵俊才

(华东计算技术研究所, 上海 201808)

通讯作者: 冯瑞青, E-mail: marblevon@163.com

摘 要: 随着嵌入式设备应用场景日趋复杂的变化, 异构多核架构逐渐成为嵌入式处理器的主流架构. 目前, 多核处理器主要采用的单操作系统模式在实际应用中存在诸多局限性. 为了充分发挥异构处理器的多核特性, 针对异构处理器不同核部署相应的操作系统并实现多操作系统协同处理技术至关重要. 本文对异构多核处理器 (ARM+DSP) 操作系统进行了研究, 在异构多核平台上成功移植了嵌入式 Linux 和国产 DSP 实时操作系统 ReWorks; 为实现 ReWorks 与 Linux 操作系统协同处理, 本文对核间通信的关键技术进行分析研究, 并以 TI 公司的 AM5718 为例, 设计了一系列多核异构通信组件. 经测试, 本文设计的异构通信组件实现了在 ARM 上对 DSP 核进行 ReWorks 操作系统和应用程序的动态加载、Linux 与 ReWorks 核间消息收发、以及 Linux 与 ReWorks 的协同计算等功能.

关键词: 异构多核处理器; 协同技术; 核间通信; ReWorks; AM5718

引用格式: 冯瑞青, 张激, 赵俊才. 异构处理器多操作系统协同技术研究. 计算机系统应用, 2018, 27(12): 90-95. <http://www.c-s-a.org.cn/1003-3254/6698.html>

Cooperative Technology of Multi-Operating System on Heterogeneous Multi-Core Processor

FENG Rui-Qing, ZHANG Ji, ZHAO Jun-Cai

(East China Institute of Computing Technology, Shanghai 201808, China)

Abstract: As application scenarios become more and more complex, heterogeneous multi-core architecture becomes the mainstream architecture of embedded processors. At present, there are many limitations in the application of the single operating system model mainly used in multi-core processors. In order to give full play to the heterogeneity of multi-core processor features, it is essential to deploy the corresponding operating system for different cores of heterogeneous processors and realize the collaborative processing technology of multiple operating systems. In this study, the operating system of heterogeneous multi-core processor (ARM+DSP) is investigated, and embedded Linux and ReWorks of domestic DSP real-time operating system are successfully transplanted on heterogeneous multi-core platform. In order to realize the collaborative processing between ReWorks and Linux operating system, this study analyzes the key technologies of inter-core communication, and takes AM5718 of TI as an example, designs a series of multi-core heterogeneous communication components. It has been tested that the heterogeneous communication components can dynamically load the DSP kernel with the ReWorks operating system and applications, send and receive messages between Linux and ReWorks cores, and perform collaborative computing between Linux and ReWorks.

Key words: HMP; multi-core communication; coprocessing; ReWorks; AM5718

① 基金项目: “核高基”国家重大科技专项 (2017ZX01041201)

Foundation item: National Science and Technology Major Program (2017ZX01041201)

收稿时间: 2018-06-01; 修改时间: 2018-06-27; 采用时间: 2018-07-03; csa 在线出版时间: 2018-12-03

随着集成电路技术的快速发展,嵌入式处理器早已从单核过渡到多核时代^[1].在移动通信、雷达信号处理、航天、工控、机器人等领域,不仅需要高性能、低功耗的处理器,而且要求单个芯片能够处理各种场景的应用,如系统控制、信号处理、通信技术、图形、图像显示等.同构多核处理器功能单一,如Cortex-A9^[2],无法应对复杂多样的场景,异构多核处理器(Heterogeneous Multi-core Processor, HMP)应运而生.典型的异构多核处理器有IBM公司的Cell Broad-Band Engine^[3]、ARM公司的big.LITTLE^[4]、TI公司的Sitara™AM57x系列^[5]、华为公司的麒麟970等.

国内外针对某种特定构架(如ARM、MIPS、X86)的嵌入式操作系统比较多,典型的有RT-Thread、ReWorks、DeltaOs、VxWorks、QNX、RT-Linux、SYS/BIOS等.目前,异构多处理器大多采用单一的嵌入式操作系统.为充分发挥异构处理器的性能,需要在不同的核上运行不同的操作系统,因此多操作系统的通信协同技术非常关键.目前,对异构多核处理器的操作系统、片上通信技术等方面的研究还并不是十分成熟.由于异构多核处理器的特殊性,实现不同核上任务调度以及不同核之间的实时通信和调度,利用不同类型处理器的各自性能特点,才能充分发挥异构多核处理器的性能优势^[6].

本文对异构多核处理器的发展以及异构多核的核间通信机制进行研究;着重研究分析了“ARM+DSP”片上基于共享总线的Mailbox异构多核间通信机制;以TI公司的Sitara™AM571为例设计实现了ARM核上Linux与DSP核上ReWorks之间的多核通信组件、DSP核的动态加载程序、协同计算等功能.

1 相关工作

1.1 ARM+DSP 异构多核处理器

“RISC+DSP”架构作为一种典型的异构多核架构在嵌入式领域应用非常广泛.“RISC+DSP”架构将通用处理器与专用处理器融合,集合了通用处理器的通用性好、任务管理能力强和专用处理器的计算能力强的优点,在功耗有限的前提下极大提升了系统性能^[7,8].其中,“ARM+DSP”架构是最常见的“RISC+DSP”架构之一,见图1.

本文以TI公司的AM5718为实例完成了具体的设计与实现.TI公司在2015年推出的是一款为工业自

动化应用包括人机交互(HMI)、工业通信、自动化和控制,机器视觉以及其它通用应用设计的28nm集成工艺的高性能处理器.ARM+DSP异构多核处理器融合了可编程的视频处理功能与高度集成的外设,旨在提供集集成度、可拓展性和外设于一体的芯片^[9],见图2.

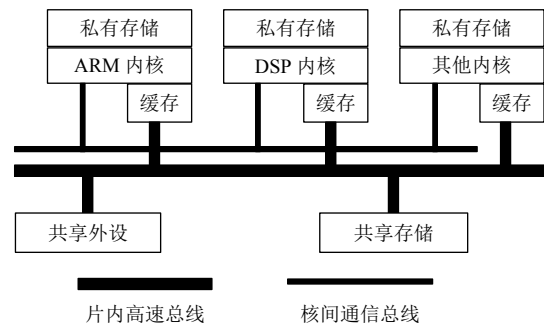


图1 ARM+DSP 异构多核处理器结构图

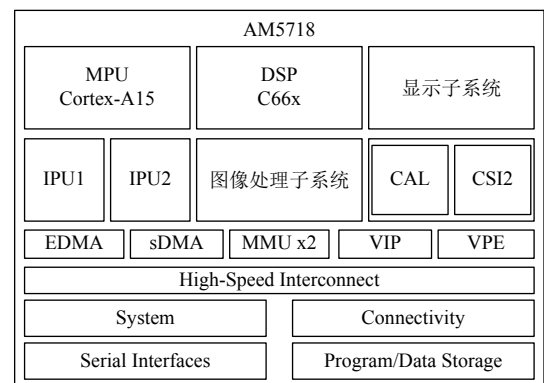


图2 AM5718 框图

1.2 嵌入式操作系统

目前异构多核处理器大多只有主核运行操作系统,其它核作为协处理器.这种架构的特点是主核功能较为复杂,负责全局任务、资源的管理和调度,从核功能单一接受主核的管理,负责执行由主核分配的任务.这种操作系统结构简单便于设计实现.如ARM+DSP异构处理器在ARM核上运行Linux,由ARM核通过指令或共享存储设备实现协处理器的运算.ARM负责事件管理和I/O接口等功能,DSP负责计算加速.这种多核协同加速处理的模式在大部分情况下可以同时满足通用性和计算能力的要求^[10].上述模式存在DSP核无法直接连接外设、DSP计算程序固定且修改程序的工作量大等缺陷.如何避免ARM核的多余过程并且实现动态加载DSP程序成为关键问题.在主核的操作系统

上对从核进行程序动态加载,并在从核上部署操作系统,通过核间通信组件可以简化上述模式流程,更好发挥从核的协同处理能力,灵活应对复杂应用场景。

本文在 AM5718 的 ARM 上运行 TI 官方提供的嵌入式 Linux 内核,在 DSP 核上运行锐华操作系统。嵌入式 Linux 操作系统是一款成熟稳定的开源操作系统,适用于不同的 CPU,支持如 X86, ARM, MIPS 等多种体系架构,具有内核小、稳定性高、资源丰富、可裁剪性强和结构灵活等特点。

在 AM5718 的 DSP 核上运行“锐华”ReWorks 操作系统。“锐华”操作系统是中国电子科技集团公司第三十二研究所(华东计算技术研究所)自主研发的嵌入式实时操作系统,提供了面向主流 DSP 芯片的高性能国产基础软件解决方案。操作系统内核采用了“十二五”核高基成果--“锐华”ReWorks 嵌入式实时操作系统。ReWorks 内核可抢占,中断可嵌套,具备高效的中断管理机制和任务调度、上下文切换算法,具有强实时性;采用微内核的体系架构和面向对象的设计方法,具有良好的可裁剪性;支持 C/C++ 编程语言,提供 POSIX 接口,具有较强的可移植性;提供故障检测、隔离和恢复框架,具有高可靠性;提供核心扩展接口,具备良好的系统可维护性;此外作为自主可控操作系统具有可持续性等特点^[1]。

锐华 DSP 操作系统软件组成,如图 3 所示,主要包括三部分:底层 CSP(芯片级支持包)/BSP(板级支持包)、操作系统核心和扩展功能组件。

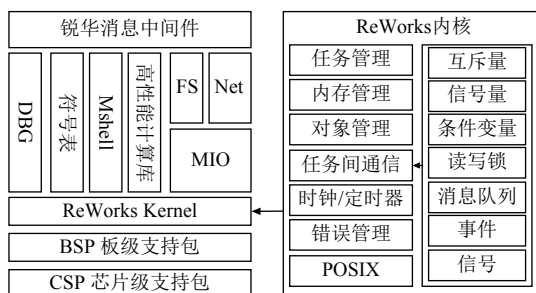


图3 “锐华”DSP操作系统软件组成

“锐华”DSP 实时操作系统在系统裁剪、实时性、易用性、性能、标准化方面具备一定的优势,与 TI 的 DSP 操作系统 SYS/BIOS 相比,ReWorks 内核除了支持任务管理、信号量、消息队列的功能外还拥有条件变量、任务读写锁等 SYS/BIOS 不支持的功能点。经

对比测试,“锐华”在相同的测试平台上多项功能、实时性优于 SYS/BIOS,满足嵌入式实时操作系统的各项性能指标要求。在下一章中,本文将设计一系列核间通信组实现上述两种嵌入式操作系统在不同核上的部署以及核间的协同处理。

2 方案设计与实现

2.1 总体方案

首先在 ARM、DSP 核上分别完成 Linux 和 ReWorks 的适配移植工作。设计一种核间通信组件基于 MailboxMessage(邮箱消息)机制实现片上处理器之间通信。其中,Mailbox 中包含 ARM 和 DSP 的共享数据寄存器组以及各自的控制/状态寄存器。ARM 和 DSP 通过访问共享寄存器和共享内存来完成数据交互,通过访问控制/状态寄存器来完成状态同步。多核通信组件使用队列邮箱中断机制允许两个不同的处理器之间通过一系列的寄存器和关联中断信号发送和接收消息来建立通信通道,总体框架见图 4。

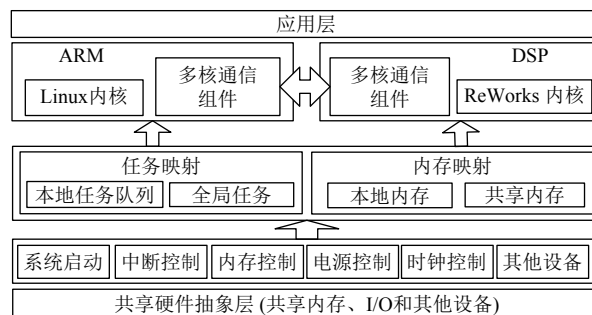


图4 Linux与ReWorks异构系统总体架构图

AM5718 执行两种邮箱类型:一种是 IVA 邮箱,数量为 1 个,用于图像加速子系统与其它核之间的通信;另一种是系统级邮箱,共有 13 种,用来实现 ARM 与 DSP、IPU 等子系统之间的通信。每个邮箱模块支持如下特性:可配置参数,包括:用户数,邮箱消息、队列数,每个消息队列(FIFO,先进先出)的消息数;32-bit 的消息宽度;使用中断完成消息接收通知和队列未滿的通知;支持 16/32-bit 寻址空间^[12]。

总体的设计思路:ARM 核作为主核先启动并且运行 Linux 操作系统,设计 ImageLoader 动态加载工具,在 Linux 运行 ImageLoader 将 ReWorks 镜像文件动态加载到 DSP 核上。DSP 上 ReWorks 操作系统等待任务消息;Linux 将任务数据写入共享内存,并发送任务消

息给 ReWorks. 收到消息后, ReWorks 从共享内存读取任务数据, 并执行相应的操作, 发送 MailboxMessage 通知 ARM 任务完成. 接下来的章节将详细介绍具体实现方法.

2.2 Linux 侧组件设计

TI 公司将 AM5718 的 ARM 核设计为主核, 上电后 ARM 核最先启动, 首先对 ARM 核上 Linux 侧的 MailboxDriver 模块和 ImageLoader 动态加载工具进行设计.

设计 Linux 系统下 MailboxDriver 模块, MailboxDriver 模块包括: DSP 子时钟系统初始化模块、DSP 子系统启动模块、邮箱设备初始化模块、MPU/DSP 消息管理模块、邮箱中断模块、邮箱管理模块、共享内存管理模块等子模块. MailboxDriver 模块初始化主要流程为: 注册邮箱字符设备创建 dsp_mailbox, 配置 MPU 和 DSP 寄存器, 分别为 ARM 和 DSP 创建一个共享内存区域查找表, 初始化 DSP 子系统时钟、唤醒 DSP, 初始化邮箱中断, 模块初始化完成.

在 MailboxDriver 模块加载完成后, 通过创建的 dsp_mailbox 设备可将 ReWorks 镜像文件动态加载到 DSP 核, 设计 ImageLoader 动态加载工具将 ReWorks 镜像加载到 DSP 核.

ImageLoader 动态加载工具的关键技术在于 elf(coff) 格式文件分析与加载; 根据 elf 文件格式解析镜像文件信息, 获取代码段、数据段等段的数据内容与相应地址、段长度等信息, 将数据写到相应的内存地址. 程序计数器获取入口地址, 程序从入口开始执行.

创建 dsp_mailbox 设备句柄, 将镜像入口地址传入 dsp_mailbox 设备并赋值给 mailbox. ImageLoader 的主要流程为先执行 elf_load 函数, 得到入口地址 entry; 再执行 start_dsp 函数, 镜像文件加载完成. ImageLoader 工具允许用户自定义镜像文件和目标核, 此设计同样适用于其他异构多核平台.

Linux 启动后先安装侧完成 MailboxDriver 模块安装, 创建 mailbox 字符设备并创建设备号等, 运行 ImageLoader 工具加载镜像创建一个计算任务并将任务数据写入共享内存特定地址, 检查 DSP 邮箱状态; 当 DSP 邮箱可用时, 向 DSP 发送消息; 收到 DSP 发送来的消息后, 读取信息并从共享内存的指定地址读取结果, 完成计算任务并释放邮箱, Linux 侧流程图见图 5.

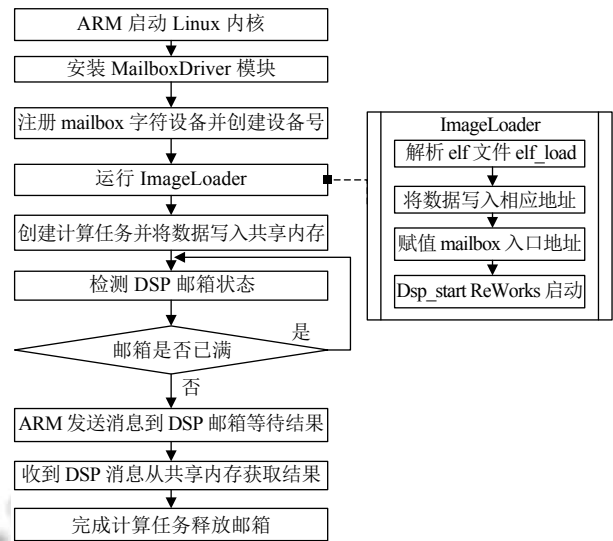


图 5 Linux 侧流程图

2.3 ReWorks 侧组件设计

设计在 DSP ReWorks 添加邮箱中断模块, ReWorks 嵌入式实时操作系统从 c_int00 开始执行, 逐步完成操作系统运行环境初始化 (例如处理器寄存器初始化、堆栈初始化等). 然后跳转到 ReWorks 内核的入口函数 main 中完成操作系统启动. 设计在 ReWorks 启动过程的中断模块初始化 (int_module_init) 增加 Mailbox 中断模块, 具体流程见图 6.

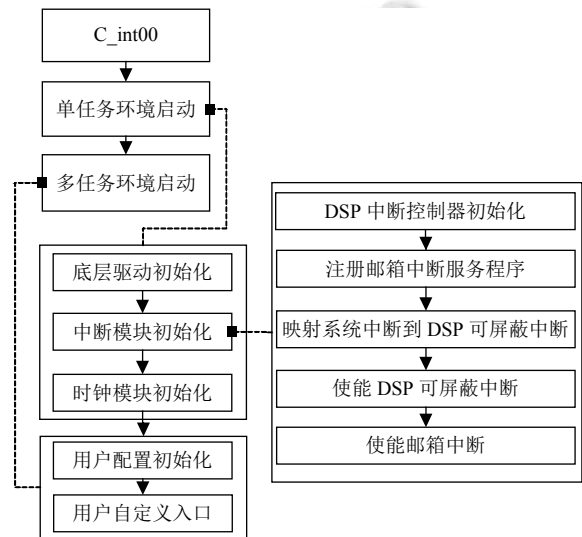


图 6 ReWorks 邮箱中断模块初始化流程图

DSP 核 ReWorks 启动后创建 idle 任务挂起. ReWorks 收到消息后从共享内存特定地址读取数据并处理, 将计算结果写入共享内存特定地址; 检测 ARM 邮箱状态, ARM 邮箱可用时, 向 ARM 发送消息; 使能

中断, 释放邮箱并退出中断, 继续挂起. DSP 侧流程图见 图 7.

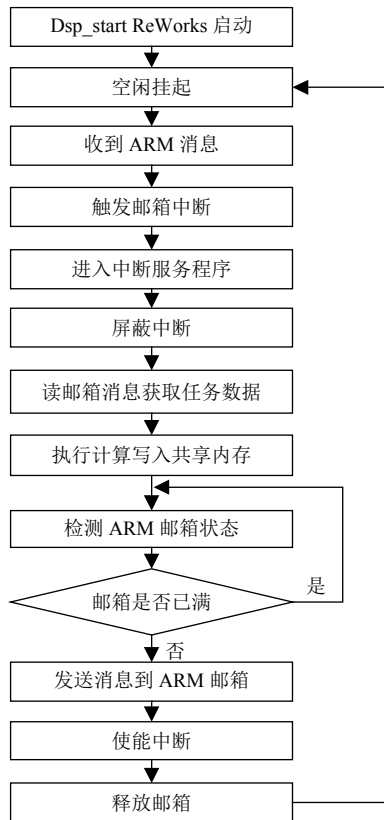


图 7 ReWorks 侧流程图

2.4 Linux 与 ReWorks 数据交互

Linux 创建计算任务后需要将本地的任务数据写入共享内存地址. 由于共享内存区域在异构多核处理器的不同核上会映射到不同的地址空间, 即同一块共享内存区域在不同处理器的本地地址空间的逻辑位置不同, 因此当 ARM Linux 创建计算任务后通过将查找共享内存地址表将本地内存地址映射到共享内存地址, 根据本地指针及区域 ID 获取当前共享区域指针, 将任务数据写入共享内存后, 通过消息将共享内存地址发送给 DSP. ReWorks DSP 收到任务消息时, 邮箱中断触发, 进入中断程序后读取消息中的共享内存地址. ReWorks 从共享内存读取任务数据需要先完成相应的地址转换, 获取本地地址后从共享内存中读取任务数据并完成相应的计算; 将结果写入转换后的共享内存地址, 以消息形式发送给 ARM 核, ARM 核得到计算结果后继续向下执行, 至此一次数据交互完成, 交互流程

见 图 8.

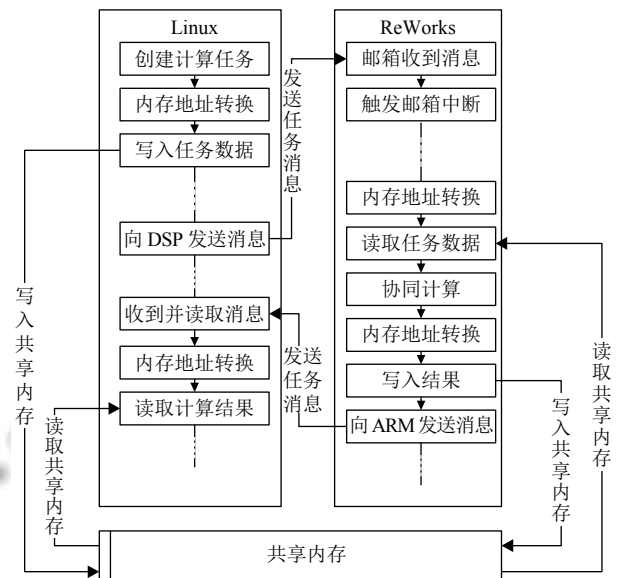


图 8 Linux 与 ReWorks 交互流程图

3 测试与分析

在完成设计之后, 我们在 AM5718 上进行了测试, ARM 和 DSP 核分别使用 RT-Linux 和 ReWorks-2.1 实时 DSP 操作系统, 依次进行下列操作:

(1) MailboxDriver 模块安装, 测试结果显示邮箱配置成功, DSP 子系统初始化完成.

(2) 创建设备号, 使用 ImageLoader 工具加载镜像, DSP 程序的入口地址读取正确, 将各个段写入相应地址成功.

(3) 核间通信测试, 由 ARM 向 DSP 发送消息, DSP 收到消息后从共享内存读取运算数据, 完成后向 ARM 发送消息.

(4) ReWorks 任务切换测试, 在 ReWorks 创建两个任务进行任务切换. 经测试, ReWorks 任务切换符合实时性要求.

(5) 协同计算: ReWorks 接受 Linux 上发来的任务消息, 成功调用 FFT 函数完成, 并将结果以消息形式返回给 Linux.

ImageLoader 对 ELF 镜像程序进行分析, 解析符号表并分配相应的内存空间后写入镜像. 测试加载一个 2 MB 的镜像文件, 使用 ImageLoader 的整个加载过程实际耗时 6.8 s. 而使用常用的 J-TAG 加载相同的镜像耗时 8.7 s. 因此, ImageLoader 在加载效率上具有优

势。ImageLoader 直接在 ARM 上运行,对 DSP 进行动态加载不需要断电重启,相比 J-TAG 等工具加载的复位等过程加载完成后,在 ReWorks 操作系统可以实现 DSP 核上的任务切换,任务切换时间为 0.25 us,符合实时性要求。继续使用 ImageLoader 向指定地址继续动态加载应用程序,这种逐层加载的方式具有灵活且简易的优势。

4 总结与展望

本文主要论述了国产 DSP 嵌入式实时操作系统 ReWorks 面向异构多核平台的核间通信研究和设计,基于 RISC+DSP 共享总线的邮箱机制,着重对 ARM Linux 与 DSP ReWorks 核间通信进行了研究分析,设计并实现了 Mailbox 邮箱驱动模块、DSP 镜像动态加载程序和 ReWorks 邮箱中断模块一系列核间通信组件。通过硬件开发板和软件程序对本文的设计进行了验证和测试分析。

本设计能够实现 Linux 端对 DSP 程序的动态加载并可以实现与 ReWorks 的核间通信,增强了 DSP 的协同处理能力,降低了 DSP 应用程序的开发难度,提高了开发效率。多核处理器的负载均衡等问题仍然是协同处理技术的难点,我们会在后续的研究工作中在 ReWorks 操作系统上继续改进或提出新的方案。

参考文献

1 Bryan Schauer. Multi-core processor: A necessity. <http://www.csa.com/discoveryguides/discoveryguides-main.php>, 2008.

- 2 Koppanalil J, Yeung G, O'Driscoll D, et al. A 1.6 GHz dual-core ARM cortex A9 implementation on a low power high-K metal gate 32nm process. Proceedings of 2011 IEEE International Symposium on VLSI Design, Automation and Test. Hsinchu, China. 2011. 1-4.
- 3 Zhou R, Chen H M, Liu Q, et al. A server model for reliable communication on cell/B.E. Proceedings of the 42nd IEEE International Conference on Parallel Processing. Lyon, France. 2013. 197-201.
- 4 Greenhalgh P. Big.LITTLE processing with ARM cortexTM-A15 & cortex-A7. ARM White Paper. ARM, 2011.
- 5 Texas Instruments. AM574, AM572x and AM571x compatibility guide. Texas Instruments, 2018.
- 6 周楠, 胡娟, 胡海明. 多核处理器发展趋势及关键技术. 计算机工程与设计, 2018, 39(2): 393-399, 467.
- 7 陈芳园, 张冬松, 王志英. 异构多核处理器体系结构设计研究. 计算机工程与科学, 2011, 33(12): 27-36. [doi: 10.3969/j.issn.1007-130X.2011.12.006]
- 8 陈国兵. 嵌入式异构多核体系的片上通信[硕士学位论文]. 杭州: 浙江大学, 2007.
- 9 Texas Instruments. AM571x Sitara™ processors silicon revision 2.0. Texas Instruments, 2017.
- 10 蒋建春, 汪同庆. 一种异构多核处理器嵌入式实时操作系统构架设计. 计算机科学, 2011, 38(6): 298-302. [doi: 10.3969/j.issn.1002-137X.2011.06.069]
- 11 李健. 高可靠嵌入式实时操作系统的研究与实现[硕士学位论文]. 上海: 上海交通大学, 2008.
- 12 Texas Instruments. AM571x (SR2.0, SR1.0) AM570x (SR2.1, SR2.0) Sitara™ processor technical reference manual. Texas Instruments, 2018.