

基于虚拟化的软件定义 DSP 计算技术研究

李志刚, 陶磊, 周玄文, 李玉成

(中国电子科技集团公司第五十二研究所, 浙江 杭州 310000)

摘要:随着天基计算密集型任务快速增长, 星上任务对计算的需求也随之迅猛上涨。星载计算机系统通常采用异构电子系统, 其“算力”被封闭在独立的星载计算机内, 各计算资源不能彼此共享, 容易产生资源浪费, 且各个嵌入式软件和可编程逻辑采用非标准的编程, 不支持统一升级和综合调用。针对此问题, 以异构 CPU 和 DSP 计算资源作为研究重点, 设计一种基于虚拟化技术的软件定义 DSP 计算架构, 通过分析当前所面临的抽象化和多租户的技术挑战, 并借鉴异构 GPU 和 FPGA 相关虚拟化技术, 综合运用硬件抽象层、时分复用虚拟化、接口调用重定向、组件注册和数据路由等技术, 实现 DSP 异构计算资源多用户共享和高效统一调用。试验验证表明, 软件定义 DSP 计算架构可实现星载异构计算系统软硬件灵活解耦, 并支持多个用户和应用同时访问和调用单个 DSP 计算资源。

关键词:硬件抽象层; 虚拟化; 时分复用; 多租户; 软件定义 DSP

中图分类号: TP391

文献标识码: A

文章编号: 1673-629X(2024)05-0010-06

doi: 10.20165/j.cnki.ISSN1673-629X.2024.0034

Research on Software Defined DSP Computing Technology Based on Virtualization

LI Zhi-gang, TAO Lei, ZHOU Xuan-wen, LI Yu-cheng

(The 52nd Research Institute of China Electronics Technology Group Corporation, Hangzhou 310000, China)

Abstract: With the rapid growth of space-based computationally intensive tasks, the demand for computing in onboard tasks has also skyrocketed. Spaceborne computer systems typically use heterogeneous electronic systems, with their "computing power" enclosed in independent onboard computers, where computing resources cannot be shared with each other, leading to resource waste. Additionally, various embedded software and programmable logic use non-standard programming, which does not support unified upgrades and comprehensive calls. To address this issue, a software defined DSP computing architecture based on virtualization technology is designed with heterogeneous CPU and DSP computing resources as the research focus. By analyzing the current challenges of abstraction and multi tenant technology, and drawing on virtualization technologies related to heterogeneous GPUs and FPGAs, technologies such as hardware abstraction layer, time division multiplexing virtualization, interface call redirection, component registration, and data routing are comprehensively utilized, the multi user sharing and efficient unified invocation of DSP heterogeneous computing resources is realized. Experimental verification shows that software defined DSP computing architecture can achieve flexible decoupling of software and hardware in spaceborne heterogeneous computing systems, and support multiple users and applications accessing and calling a single DSP computing resource simultaneously.

Key words: hardware abstraction layer; virtualization; time division multiplexing; multi tenant; software defined DSP

0 引言

随着数字化和智能化趋势不断加深, 数据计算量和信息处理规模出现爆发式增长, 对信息系统的计算和控制能力均提出更高的要求。以新一代天基信息系统智能计算需求为例^[1], 随着天基计算密集型任务快速增长, 星上任务对计算的需求也随之迅猛上涨, 且远远超过了传统星载计算机的能力所及。传统航天器的

星载计算机系统是一个典型异构电子系统^[2], 根据任务需求不同通常采用 CPU、GPU、FPGA、DSP 等各种计算资源^[3], 其“算力”被封闭在独立的星载计算机内, 各计算机的资源不能彼此共享, 产生资源浪费, 且各个嵌入式软件和可编程逻辑采用非标准的编程, 不支持统一升级和综合调用。现有的功能独立的封闭式航天电子系统已经不能满足航天器智能计算

收稿日期: 2023-08-15

修回日期: 2023-12-15

基金项目: 国家重点研发计划(2020YFB1806002)

作者简介: 李志刚(1983-), 男, 高级工程师, 研究方向为计算机应用、云计算等。

需求^[4]。

因此,有必要在设法提高单点计算能力的同时,借鉴软件定义无线电^[5]相关思想,采用虚化或云化的软件定义计算架构,由若干异构器件互相配合完成目标计算任务,共享稀有计算资源,提升计算效率,满足智能化计算的需求。

该文从软件定义异构计算面临的挑战出发,以异构 CPU、DSP 计算资源作为研究重点,提出一种基于虚拟化的软件定义 DSP 计算架构,以期为空间异构计算资源的高效灵活使用提供有价值的建议和参考。

1 软件定义 DSP 计算面临的挑战

参照软件定义的核心本质^[6],软件定义 DSP 计算是在 DSP 硬件资源数字化、标准化的基础上,通过软件编程去实现虚拟化、灵活、多样和定制化的功能,对外提供客户化的专用智能化、定制化的服务,实现应用软件与 DSP 硬件的深度融合。

虚拟化技术是软件定义计算最主要的解决途径,文中主要基于 DSP 虚拟化技术来实现软件定义 DSP 计算。DSP 虚拟化的主要目标及面临的挑战如下:

(1) 抽象化。

DSP 虚拟化的第一个目标是通过创建物理硬件的抽象,并为应用程序提供简单易用的接口来访问虚拟 DSP 资源,从而提高 DSP 的可用性,使开发和和使用 DSP 变得容易。这就需要构建灵活通用的硬件抽象

层向应用开发人员隐藏硬件和底层交互细节。

(2) 多租户^[7]。

DSP 虚拟化的第二个目标是在多个用户和应用程序之间共享相同的 DSP 资源。在 DSP 虚拟化基础设施中,有必要支持多个应用程序,因此 DSP 虚拟化系统需支持跨多个用户共享资源。

传统上,DSP 主要用于嵌入式应用程序,其中在 DSP 上运行的专用应用程序只能由专用用户访问,受限于 DSP 架构和设计流程,无法在多个用户和应用之间共享 DSP 硬件资源,使 DSP 虚拟化的多租户计算目标成为一项具有挑战性的任务。

2 软件定义 DSP 计算的设计

2.1 系统架构

对于 DSP、GPU、FPGA 等异构计算器件,目前工业界与学术界实现 GPU 虚拟化的方式主要有设备模拟(QEMU)^[8]、API 重定向(vCUDA^[9]、rCUDA^[10])及 GPU 全虚拟化(gVirt)等方式,实现 FPGA 虚拟化的方式主要有 FPGA Overlay^[11]、微软的 Catapult^[12]、IBM cloudFPGA^[13]等,但尚无实现 DSP 虚拟化的技术方式。

文中主要借鉴 GPU 和 FPGA 相关虚拟化技术,并根据 DSP 动态重构的特点^[14],基于时分复用虚拟化的方式实现软件定义 DSP 计算,其技术架构堆栈如图 1 所示。

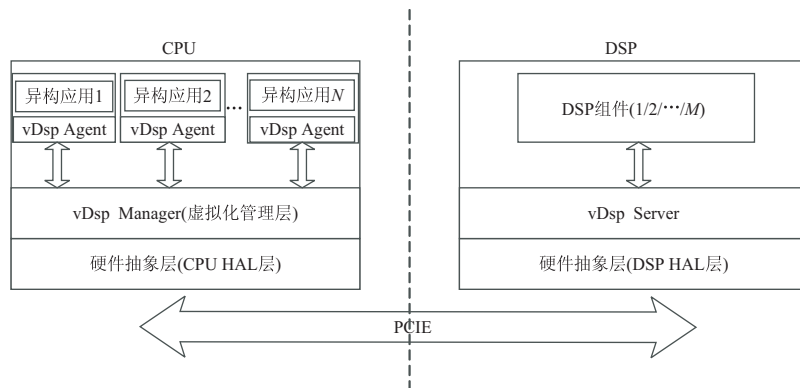


图 1 软件定义 DSP 计算的技术架构堆栈

由图 1 可知,该软件定义 DSP 计算系统架构采用分层架构,CPU 和 DSP 之间通过 PCIE 总线互联,其架构实现涉及三类软件堆栈:硬件抽象层(CPU HAL 层和 DSP HAL 层)、虚拟化管理层(vDsp Manager)和远程调用框架层(vDsp Agent 和 vDsp Server)。

(1) 硬件抽象层。

硬件抽象层通过对底层硬件模块接口进行高度抽象定义,为上层组件或应用提供一致的通信访问接口,使得开发人员不必关心如何与底层硬件实现交互,从而在很大程度上实现上层设计与具体硬件平台解

耦合^[15]。

硬件抽象层包括 CPU 侧 CPU HAL 层和 DSP 端 DSP HAL 层,主要职责是作为软总线实现 CPU 和 DSP 端对象之间的通信,使 CPU 侧请求和 DSP 端对象间连接独立于软硬件平台。其主要功能包括数据读写、报文封装和解析、屏蔽底层交互协议以及提供接口库等,从而实现透明地收发 CPU 和 DSP 间的请求和应答。

(2) 虚拟化管理层。

虚拟化管理层的主要目的是实现多租户支持,即

在多个用户和应用程序之间高效共享 DSP 硬件资源, 以提升 DSP 资源的利用率。对于多租户支持, 需要具有空间或时间多路复用能力。考虑到 DSP 器件独占使用的特性^[16], 文中采用时分复用的方式来实现 DSP 虚拟化。

该层主要实现将多个 CPU-DSP 异构应用的 DSP 请求进行队列管理, 并将请求重定向到实际的物理 DSP 进行请求处理。

(3) 远程调用框架层^[17]。

远程调用框架层主要实现在 CPU 侧可以透明地调用 DSP 端对象, 其包括运行在 CPU 侧的 vDsp Agent 模块和运行在 DSP 端的 vDsp Server 模块。

vDsp Agent 功能体现在两个方面: 一方面接收 CPU 侧异构应用的请求并对其进行编码, 然后通过硬件抽象层发送到 DSP 端; 另一方面接收 DSP 端通过硬件抽象层返回的结果并对其进行编码, 然后返回给 CPU 侧异构应用。

vDsp Server 功能与 vDsp Agent 类似, 其实现两方面的功能: 一是接收 CPU 侧异构应用通过硬件抽象层发送的请求, 然后路由发送给 DSP 组件; 另一方面接受 DSP 端的应答并对其进行编码, 然后通过硬件抽象层发送至 CPU 侧。

2.2 硬件抽象层

硬件抽象层是在 CPU 和 DSP 中运行的基础支撑软件, 是一个底层的通信模型, 其为上层应用和硬件之间的通信提供标准协议和接口^[18], 功能模块示意图如图 2 所示。

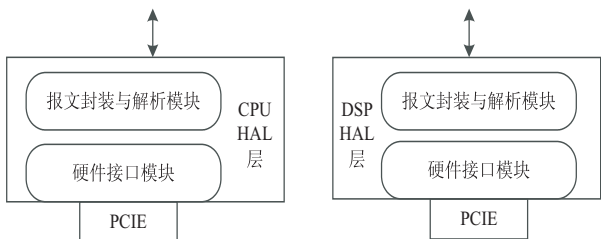


图 2 硬件抽象层功能模块示意图

CPU 和 DSP 数据交互中采用 HAL 报文方式^[19], DSP 硬件抽象层接口模块收到的数据是标准的 HAL 报文, 报文的解析功能实现对报文的解析, 然后将解析后的数据送到 vDsp Server 进行处理。而对于 DSP 组件通过 vDsp Server 发来的数据, 必须经过报文的封装模块封装成标准的 HAL 报文, 之后再报文交给 DSP 硬件接口模块进行发送。

HAL 报文格式如图 3 所示。

HAL 报文中各字段的意义如下所述:

(1) MS 字段: MS 字段用来指示 DSP 组件是否已完成消息处理, DSP 组件在消息处理结束之前需设置

该字段, 以声明消息还在处理。



图 3 HAL 报文格式

(2) LD 字段: 表示消息应被送达的目的端口的逻辑地址, 每个端口都有一个 LD。LD 是在 DSP 内定义的常量, 指向一个 DSP 组件中具体的接口, 例如 DSP 组件的数据端口、DSP 组件的处理端口等。

(3) LEN 字段: 硬件抽象层消息报文长度。

(4) Payload 字段: 实际传输的报文内容。

2.3 虚拟化管理层

虚拟化管理层通过拦截各个异构应用中的 API 并将其重定向到真实的 DSP 设备来工作。其功能模块示意图如图 4 所示。

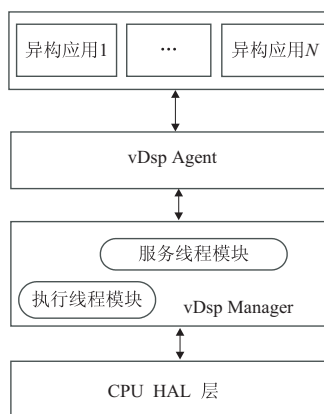


图 4 虚拟化管理层功能模块示意图

在该虚拟化管理层中设计一个一对多的模型在多个用户应用中时分复用 DSP 设备, 图 5 显示了一对多模型。

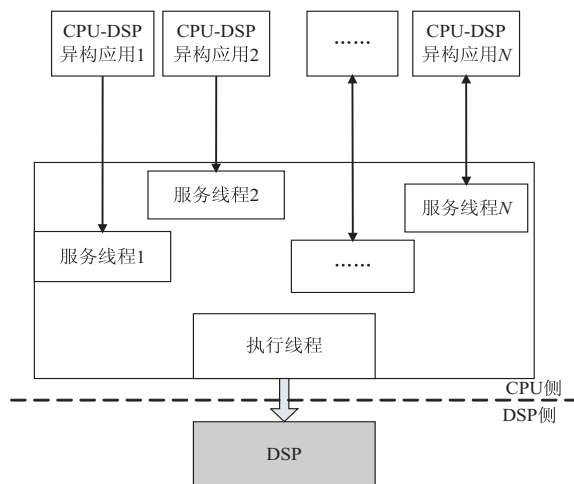


图 5 虚拟化管理层 DSP 时分复用模型

在此模型中, vDsp Manager 中有两种不同的线程: 执行线程和服务线程。

每个服务线程对应一个等待 DSP 服务的 CPU-DSP 异构应用, 该线程接收 vDsp Agent 的 DSP 调用请求命令, 将命令转换为服务器端表示, 然后将命令和相关信息转发给执行线程。

每个执行线程对应一个物理 DSP 器件, 它从服务线程中获取 DSP 函数参数, 并操纵 DSP 器件完成 DSP 调用请求, 当执行线程完成后, 服务线程收集结果并将其传输到相应的 vDsp Agent 作为返回结果。

在 vDsp Manager 的协调下, 两个或多个不同的服务线程可以通过连接到一个执行线程来协同操作一个 DSP 硬件资源, 执行线程负责将服务线程中的所有请求序列化并通过真实 DSP 器件中组件的调用一一执行。

2.4 远程调用框架层

远程调用框架层主要实现在 CPU 侧可以透明地调用 DSP 端对象, 其包括运行在 CPU 侧的 vDsp Agent 模块和运行在 DSP 端的 vDsp Server 模块。其功能模块示意图如图 6 所示。

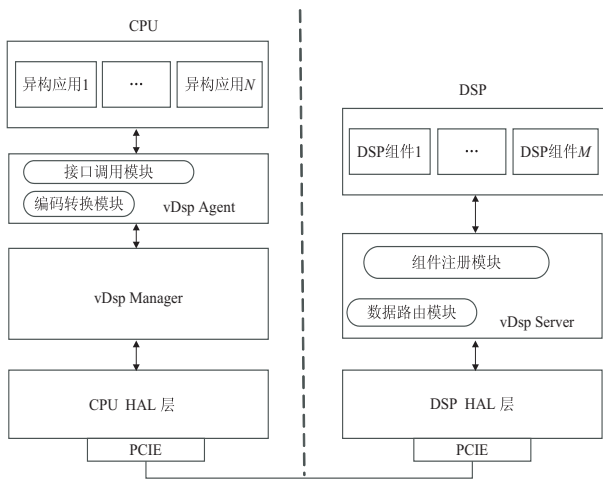


图 6 远程调用框架层功能模块示意图

(1) vDsp Agent。

vDsp Agent 主要包括接口调用模块和编码转换模块。其提供接口 API 供 CPU-DSP 异构应用调用, 并将调用接口的参数进行编码转换, 然后通过 vDsp Manager 和硬件抽象层发送到 DSP 端。当接收到 vDsp Manager 返回的执行结果时, 将执行结果进行编码, 然后通过接口返回给 CPU-DSP 异构应用。

(2) vDsp Server。

vDsp Agent 主要包括组件注册模块和数据路由模块。文中设计是基于信号量触发的多任务模式, 各个 DSP 组件都可看作一个任务, 并设计相应信号量来触发。但当 DSP 组件读任务被触发时, 自身并不知道读数据的地址, 仅知道有数据给它, 因此还需一个用来传

递地址的变量。对于各个 DSP 组件而言, 需注册能触发自身任务的信号量以及相应读数据地址。

对于数据路由模块的上行路由来说, 路由模块收到报文解析模块发来的数据, 根据 LD 信息读取注册信息, 然后读出注册表中该 LD 对应的信号量及相应的读数据地址, 并释放该信号量, 进行后续操作。

对于数据路由模块的下行路由来说, 路由模块接收到 DSP 组件发来的数据, 在注册表中根据 LD 获取该 LD 对应的任务写地址, 然后将该地址中的数据进行报文封装, 并调用相应的硬件接口模块将数据正确地发送出去。

2.5 工作流程

文中通过虚拟化方法实现多用户 CPU-DSP 异构应用同时访问 DSP 硬件资源的工作流程主要分为以下几步:

(1) CPU-DSP 异构应用在需使用 DSP 组件服务时通过 vDsp Agent 发出 DSP 调用请求, 在该架构下, 支持多个 CPU-DSP 异构应用同时发出调用请求;

(2) vDsp Manager 将调用请求进行排队处理, 并启动 CPU 侧服务线程接收 DSP 调用请求命令, 将命令转换为服务端表示, 然后将命令和相关信息转发给执行线程, 如图 5 所示;

(3) 执行线程对应实际物理 DSP 设备, 它从服务线程中获取 DSP 函数参数, 将服务线程中的请求序列化并重定向为对 DSP 的调用;

(4) DSP 侧根据服务线程中的具体请求情况路由选择并触发运行相应 DSP 组件;

(5) 通过 CPU 和 DSP 的硬件抽象层将序列化的请求传递至 vDsp Server, vDsp Server 将该请求路由发送给相应的 DSP 组件;

(6) DSP 组件处理该请求, 将处理结果通过 vDsp Server、硬件抽象层及 vDsp Agent 传输并编码返回给 CPU-DSP 异构应用。

3 试验验证

3.1 试验平台

本技术研究基于 CPU-DSP 处理板平台进行, 该处理板集成了 CPU 和 DSP 芯片, CPU 和 DSP 之间通过 PCIE 高速总线互联, 其中 CPU 采用国产 FT2000/4 处理器, 用于 DSP 控制管理并提供通用计算资源, DSP 采用 TI TMS320C6678, 用于图像或信号处理。

CPU-DSP 处理板采用 VPX 行业标准设计, 单元机械尺寸为 3U 标准结构设计, 实物图如图 7 所示。

3.2 多租户使用验证

传统上, DSP 主要用于嵌入式应用程序, 在 CPU 与 DSP 协同工作时, 一般先将编写好的 DSP 嵌入式应

用程序加载部署到 DSP 中,然后启动 DSP 运行,在 DSP 程序运行后,运行于 CPU 上的程序通过 PCIE 接口与 DSP 进行交互通信和数据传递,从而完成信号和数据处理工作。当需要 DSP 运行其他处理任务时,需重新编写 DSP 嵌入式应用程序并加载到 DSP 中运行,在 DSP 上运行的专用应用程序只能由专用用户访问,不支持由多个用户同时共享访问单个 DSP 资源。



图 7 CPU-DSP 处理板实物图

在采用基于虚拟化的软件定义 DSP 计算架构后,通过在 CPU-DSP 处理板上部署资源虚拟化演示软件对 DSP 多租户虚拟化功能进行验证测试。该测试通过虚拟化演示软件创建图形处理相关的多个 CPU-DSP 用户任务,并同时运行进行图像处理,通过输入输出结果查看对 DSP 虚拟化功能进行测试。

如图 8 所示,可创建多个 CPU-DSP 任务(如下所示为 8 个),其中每个 CPU-DSP 任务包括 CPU 上的异构应用以及 DSP 中的图像处理任务组件。



图 8 CPU-DSP 多任务创建及启动示意图

在演示软件中同时启动运行多个 CPU-DSP 异构应用,包括图像分割和直方图均衡化等,输入输出结果如图 9、10 所示。

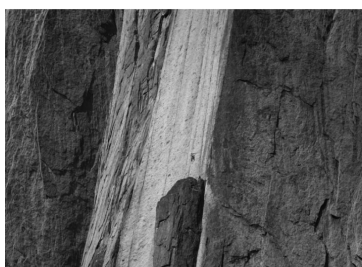


(a)

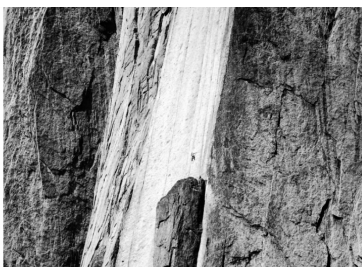


(b)

图 9 图像分割任务输入输出示意图



(a)



(b)

图 10 直方图均衡化任务输入输出示意图

图 9(a)为图像分割前的图片,图 9(b)是图像分割后的图片。

图 10(a)为原始图像,图 10(b)为直方图均衡化

后的图像。

由上述多个 CPU-DSP 任务同时运行验证可知,文中的软件定义 DSP 计算技术可支持多个用户同时

访问使用 DSP 硬件资源。

4 结束语

通过硬件抽象层设计及时分复用虚拟化技术等实现的软件定义 DSP 计算架构,可有效实现系统软硬件灵活解耦及异构 DSP 资源的多租户同时访问使用,提高异构 DSP 资源使用的灵活性和利用效率。

但当前软件定义 DSP 计算架构也存在一定的不足:一是硬件抽象层技术虽然一定程度上解决了异构器件间的互联互通问题,但实际应用中各机构实现的硬件抽象层在通用性、稳定性以及效率方面得不到保证;二是虚拟化管理层采用的是基于粗粒度的任务组件切换的时分复用技术,存在一定的上下文切换和虚拟化开销。在后续的研究中可以将这些不足之处作为方向进行优化设计和深入研究。

参考文献:

- [1] 李双博,张艳松,李德仁,等.我国天基信息系统发展现状及未来展望[J].中国工程科学,2020,22(1):127-132.
- [2] 张明栋.一种基于 DSP 的高可靠星载浮点计算平台的设计[J].现代工业经济和信息化,2019,46(4):32-34.
- [3] 王元乐,杨玉辰,方火能,等.一种可重构的星载高性能智能异构计算系统[J].空间控制技术与应用,2022,48(5):125-132.
- [4] 朱 维.航天器综合电子系统技术研究[D].上海:上海交通大学,2013.
- [5] 杨小牛,楼才艺,徐建良.软件无线电技术与应用[M].北京:北京理工大学出版社,2010:123-128.
- [6] 梅 宏,郭 耀.面向网构软件的操作系统的操作系统:发展及现状[J].科技导报,2016,34(14):33-41.
- [7] 杨 兴.云计算环境下多租户模式的研究与应用[D].成都:电子科技大学,2015.
- [8] 李 强.QEMU/KVM 源码解析与应用[M].北京:机械工业出版社,2020:24-27.
- [9] SHI L, CHEN H, SUN J, et al. vCUDA: GPU-accelerated high performance computing in virtual machines[J]. IEEE Transactions on Computers, 2012, 61(6): 804-816.
- [10] DUATO J, PENA A J, SILLA F, et al. rCUDA: reducing the number of GPU-based accelerators in high performance clusters[C]//2010 international conference on high performance computing and simulation (HPCS). Caen: IEEE, 2010: 224-231.
- [11] SOH K H, LIU C. FPGA overlays[C]//FPGAs for software programmers. [s. l.]: Springer, 2016: 285-305.
- [12] PUTNAM A. A reconfigurable fabric for accelerating large-scale datacenter services[C]//Proceedings of the IEEE forty-first annual international symposium on computer architecture (ISCA'14). Minneapolis: IEEE, 2014: 13-24.
- [13] FAHMY S A, VIPIN K, SHREEJITH S. Virtualized FPGA accelerators for efficient cloud computing[C]//Proceedings of the IEEE international conference on cloud computing technology and science. [s. l.]: IEEE, 2015: 430-435.
- [14] 白生炜.多核 DSP 局部重构系统的设计与实现[D].天津:天津工业大学,2017.
- [15] 陈 伟.基于多核 DSP 的通用软件无线电平台设计与实现[D].南京:南京理工大学,2014.
- [16] 牛金海.TMS320C66x Keystone 架构多核 DSP 入门与实例精解[M].上海:上海交通大学出版社,2014:275-280.
- [17] 齐永波.远程调用技术的研究及其应用[D].大连:大连理工大学,2015.
- [18] 王金波.基于 SCA 的硬件抽象层连接设计及其在 DSP/FPGA 上的实现[D].长沙:国防科学技术大学,2005.
- [19] 刘鹏飞,洪锡军,张 激.SCA 硬件抽象层接口设计[J].计算机工程,2008,34(z1):31-33.