计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(7):277−282 [doi: 10.15888/j.cnki.csa.007990]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

通用非对称多核方案设计①

陈　彬

(南京国电南自电网自动化有限公司, 南京 211153)
通讯作者: 陈　彬, E-mail: bin.c.chen@foxmail.com

摘　要: 多核处理器是目前处理器发展的主流方向, 但硬实时保证方面存在诸多挑战. 通过研究分析实时应用需求和多核处理器的应用现状, 提出一种基于通用处理器的非对称多核方案. 重点讨论了方案的软件总体设计、共享资源管理、非对称多核模式下的从核镜像加载、启动和核间通信的设计. 采用通用非对称多核方案研制的低压保护测控装置, 其现场运行情况表明方案满足电力二次设备的实时性能要求.

关键词: 非对称多核; 实时; 从核镜像加载; 核间通信; 电力二次设备

引用格式: 陈彬.通用非对称多核方案设计.计算机系统应用,2021,30(7):277–282. http://www.c-s-a.org.cn/1003-3254/7990.html

Design of General Asymmetric Multiprocessing Program

CHEN Bin

(Nanjing SAC Power Grid Automation Co. Ltd., Nanjing 211153, China)

Abstract: Multi-core processors point out the mainstream direction of processor development, but there are many challenges in hard real-time assurance. By analyzing the real-time requirements and the current application of multi-core processors, we propose an asymmetric multiprocessing program based on general-purpose processors. In this study, we focus on the overall software design, the management of shared resources, and the designs of image loading and starting of slave processors and inter-core communication in the asymmetric multiprocessing mode. The low-voltage protection device for measuring and control is developed by ourprogram, and its on-site operation shows that the program meets the real-time performance requirements of secondary power equipment.

Key words: AMP; real time; slave processor image load; inter-core communication; secondary power equipment

随着摩尔定律的终结, 多核处理器和并行计算得到大量关注, 多核并行计算技术因为能充分利用当前计算机系统的计算能力, 同时降低成本和功耗[1−3], 提升产品竞争力, 是当前的热点应用技术. 但目前多核处理器较少应用于实时系统, 这是因为实时系统要求任务的执行时间不能超过截止期, 而相对于单核处理器而言, 多核处理器在访问硬件共享资源 (如 Cache、内部互联总线、内存等) 时产生竞争干扰, 很难获得任务安全、准确的最坏情况执行时间, 这就要求处理器体系结构设计能给出确定性的实时性预估结果[4].

微机保护测控装置是对电力系统内一次设备进行监察、测量、控制、保护、调节的辅助设备[5], 其中电气量采集分析、保护控制信号响应等功能有硬实时性要求, 而屏幕显示、键盘输入、打印、网络通信等功能对实时性要求不高, 但希望软件功能丰富, 硬件覆盖全面[6,7]. 为满足这些需求, 微机保护测控装置多采用分布式多板卡多处理器硬件设计方案[8−10], 使用不同的板卡或处理器来分别执行实时任务和非实时任务. 这种分布式硬件方案可扩展性高, 实时功能和非实时功能的隔离性好, 但是系统方案复杂, 软件开发难度大, 设

① 收稿时间: 2020-10-23; 修改时间: 2020-11-23, 2020-12-12; 采用时间: 2020-12-18; csa 在线出版时间: 2021-06-30

备功耗大, 成本高. 而多核处理器的出现, 通过非对称多处理 (Asymmetric MultiProcessing, AMP), 在一个或多个处理器核心上运行通用操作系统, 如 Linux, 获得完善的硬件生态支持和丰富功能, 在另外一个或多个处理器核心上运行实时操作系统或裸程序, 满足装置的实时计算和控制要求. 相比于传统的分布式硬件设计方案, AMP 方案利用一个多核处理器实现了多个板卡或处理器的功能, 对成本敏感的低压保护等装置, 能带来较大竞争优势.

现有的 AMP 硬件设计方案中, 以异构处理器和 Xilinx Zynq-7000 处理器为主. 异构处理器多是 CPU+DSP 结构[11–13], 其中 CPU 运行非实时任务, DSP 运行实时任务. Xilinx Zynq-7000 是一款双核 CPU+FPGA 结构的 SoC 芯片, 其中 CPU0 运行非实时任务, CPU1 运行实时任务[14–16]. Xilinx Zynq-7000 针对 AMP 应用场景做了诸多芯片设计优化, 并开发了 OpenAMP 软件开发框架[17].

现有的 AMP 软件设计方案中, 以 MCAPI (Multicore Communications API) 和 OpenAMP 框架为主. MCAPI 是一个轻量级开源多核通信框架, 支持基于消息传递形式的嵌入式多核通信核同步编程[18]; OpenAMP 是 Xilinx 与 Mentor Graphics 公司共同开发的一个标准化的嵌入式多核框架, 支持 IPC 消息管理、共享内存管理、从核 CPU 的生命周期管理和资源管理等功能[19]. MCAPI 缺少从核 CPU 的生命周期管理功能; OpenAMP 有硬件依赖层, 目前只支持 Xilinx Zynq 处理器. 同时, 两者实现相对复杂, 不支持核间网络通信机制, 支持的实时操作系统种类也有限.

MCAPI 缺少从核 CPU 的生命周期管理功能, OpenAMP 缺少对通用处理器的支持. 针对上述问题, 本文提出一种基于通用多核处理器的 AMP 解决方案, 并重点讨论了 AMP 软件总体设计、共享资源管理方法和核心模块设计.

1 AMP 软件总体设计

AMP 软件总体模块框图如图 1 所示, 分为 AMP 启动和 AMP 通信两个子系统. AMP 启动子系统由用户态模块 amprun 和内核态模块 amproc 实现, 负责 AMP 从核的程序加载、启动、停止、重启等功能. AMP 通信子系统由用户态模块 ampcomm 和内核态模块 ampeth 实现, 负责 AMP 核间网络通信. 内核态模块 amproc 主要实现 AMP 从核的控制管理, 通过 sysfs 导出用户态访问接口给用户态模块 amprun 使用; 内核态模块 ampeth 在 Linux 网络子系统和 Linux 核间中断子系统之上, 基于共享内存和核间中断, 实现一个虚拟网口, 提供给用户态模块 ampcomm 实现 AMP 核间的套接字网络通信.

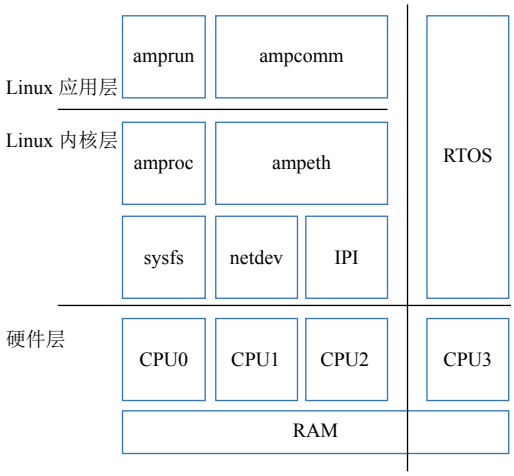

图 1 AMP 软件模块框图

与 OpenAMP 相比, 本方案对操作系统组件依赖少, 没有硬件依赖, 为快速适配到不同的通用处理器和操作系统创造了有利条件. 与 MCAPI 相比, 本方案增加了从核 CPU 的生命周期管理功能, 功能更加全面. 本方案的核间通信采用网络通信机制, 是 OpenAMP 和 MCAPI 都不支持的. 核间网络通信的缺点是实时精度不高, 但满足继电保护装置的分布式板卡通信性能要求, 而且有丰富的网络应用服务支撑和兼容当前基于网络通信的分布式系统架构等优点.

2 共享资源管理

多核处理器中, 除了 CPU 核心、L1 Cache、TLB、MMU 等少数模块是每个 CPU 的私有资源外, 其他硬件资源都是 CPU 共享资源, 如 Last Level Cache、片内互联总线、内存、外设等. 当硬实时任务在同时访问这些共享资源时, 就会发生资源访问冲突, 给硬实时任务带来不可预测的额外执行时间. 为了避免访问冲突, 需要对所有共享资源进行划分和管理.

2.1 内存

在各类共享资源中, 对系统性能影响最大的是共享内存和 Cache, 为了缓解这一问题, 除了增加可用共

享资源外, 公平高效地管理和利用共享内存资源至关重要[20]. 基于通用考虑, AMP 通信功能基于共享内存实现. 内存划分为 3 部分, 即共享内存区、Linux 内存区和 RTOS 内存区, 如图 2 所示.

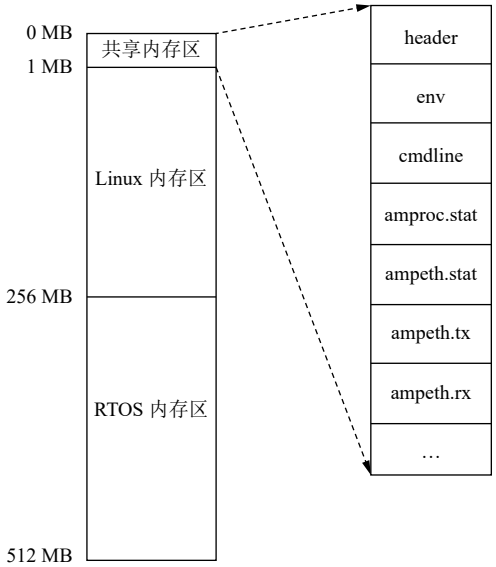

图 2 内存划分图

共享内存区位于内存头部, 方便 Linux 内存区和 RTOS 内存区的扩展. 共享内存由 Linux 初始化, Linux 和 RTOS 共同使用. 共享内存区内部划分为多个段, 在 AMP 驱动加载时按需动态申请, 段信息由共享内存区头 header 维护, 其中包括 RTOS 的环境变量 env、RTOS 的启动参数段 cmdline、AMP 从核运行状态 amproc.stat、AMP 核间通信网络状态 ampeth.stat、AMP 核间通信网络发送缓存 ampeth.tx、AMP 核间通信网络接收缓存 ampeth.rx 等.

不同的内存区设置不同的内存属性, 其中共享内存的内存属性设置为 non-cacheable, 避免单次数据占用缓存资源; Linux 内存区和 RTOS 内存区的内存属性设置为 non-shareable, 避免 cache/TLB 一致性维护在核间广播, 挤占内部互连总线带宽资源, 影响 CPU 性能.

2.2 外设

根据 AMP 主核和从核的不同功能定位, 将除内存之外的所有外设分别分配给 AMP 主核或从核使用, 尽量保证主核和从核上的外设在不同的总线上, 避免外设总线访问竞争.

2.3 处理器

AMP 从核工作模式设置为 AMP 模式, 保证 AMP 从核不会发送和接受 cache/TLB 的一致性维护广播, 有利于提升 AMP 从核的实时性.

2.4 Cache 缓存

AMP 从核关闭 Last Level Cache, 保证 AMP 从核能获得可靠的任务最坏情况执行时间, 避免因为 AMP 主核任务污染 Last Level Cache 而造成 AMP 从核任务执行时间的抖动和恶化. Last Level Cache 由 AMP 主核控制和使用. 如果 AMP 从核希望对 L2 Cache 进行刷新、无效等操作, 需要通知 AMP 主核, 由主核进行操作.

2.5 片内总线

多核处理器大都在互联结构中集成了控制多核之间优先级的总线仲裁功能, 可以在启动 AMP 从核的同时设置相应优先级.

如果处理器缺少控制多核之间优先级的总线仲裁功能, 为了保证实时任务的实时计算, AMP 主核上的软件需要精心设计, 避免长时间的总线访问风暴, 大量的总线冲突导致访问队列迅速增长, 延迟加剧, 造成 AMP 从核实时任务超时.

3 AMP 软件功能设计

3.1 AMP 从核镜像加载

AMP 从核运行实时系统需要加载的镜像文件包括 RTOS 镜像、配置文件 config、环境变量 env 和启动参数 cmdline 四部分.

RTOS 镜像、配置文件 config 采用 FIT (Flattened Image Tree) 格式打包, amprun 模块根据 FIT 镜像中的属性信息如文件列表、压缩格式、校验算法、加载地址等, 负责解析、校验、解压、加载到 RTOS 内存区.

环境变量 env 存储在 nor flash 上, 而 nor flash 由 AMP 主核控制, 所以需要 amprun 模块从 nor flash 中读出环境变量, 并写入共享内存区的环境变量 env 段, 以便 AMP 从核启动时使用.

为了灵活控制 RTOS 的启动模式和功能等, 启动参数 cmdline 需要根据用户输入动态生成, 所以 amprun 模块根据用户输入动态生成 cmdline, 并写入共享内存区的启动参数 cmdline 段.

3.2 AMP 从核启动

完成 AMP 从核镜像加载后, amprun 模块使用 amproc 模块提供的接口, 完成 AMP 从核的启动、停止、重启等生命周期的控制管理.

启动 AMP 从核操作过程如下:

1) 初始化 AMP 从核运行状态 amproc.stat;

2) 禁止 CPU 本地中断;

3) 内存屏障;

4) 刷新 L1 和 L2 Cache(PoU);

5) 通知其他所有 SMP CPU 等待 AMP 从核完成启动 (方法与步骤 (11) 相同);

6) 设置 CPU 软件复位启动地址寄存器为 AMP 从核 RTOS 镜像的入口地址;

7) 复位 AMP 从核;

8) 复位 AMP 从核的 L1 Cache;

9) 禁止 AMP 从核的外部调试访问;

10) 解复位 AMP 从核, AMP 从核启动;

11) 循环检查 AMP 从核运行状态 amproc.stat, 等待 AMP 从核完成启动. (AMP 从核完成启动后会更新 amproc.stat);

12) 使能 CPU 本地中断;

停止 AMP 从核操作过程如下:

1) 复位 AMP 从核;

2) 禁止 AMP 从核的外部调试访问;

3) 下电 AMP 从核;

4) 初始化 AMP 从核运行状态 amproc.stat;

3.3 AMP 核间通信

AMP 核间通信通过虚拟网络接口, 其基于共享内存和核间中断实现. 共享内存段中的 ampeth.stat、ampeth.tx、ampeth.rx 用于 AMP 核间通信. ampeth.tx、ampeth.rx 共享内存段结构相同, 由 64 个缓冲区描述符对象 (Buffer Descriptor, BD) 组成. 每个 BD 由有效状态 valid、帧长 len 和帧数据区 data 三部分组成. ampeth.tx 对于 AMP 主核是发送缓冲区, 对于 AMP 从核是接收缓冲区; ampeth.rx 对于 AMP 主核是接收缓冲区, 对于 AMP 从核是发送缓冲区. ampeth.stat 共享内存段由发送缓冲区描述符对象头索引 tx_idx 和发送缓冲区描述符对象头索引 rx_idx 组成, 用于 tx_idx 和 rx_idx 在主从核间的同步.

AMP 核间通信虚拟网口的发送流程如图 3 所示. 首先检查当前发送 BD 是否空闲, 如果非空闲, 说明整个发送缓存已没有可用的空闲 BD, 丢弃发送帧并退出. 如果当前 BD 空闲, 检查发送帧长度是否超过 BD 数据区长度, 如果超出, 丢弃发送帧并退出. 当所有检查都通过后, 将发送帧数据拷贝到当前发送 BD 的数据区, 并将当前 BD 设置为有效 BD, 然后更新网络统计计数, 移动发送 BD 的头索引到下一位置, 并将新的发送 BD 头索引更新到 ampeth.stat 段中, 最后给 AMP 核发送一个 IPI 收包中断, 通知 AMP 核可以从共享内存中接收数据了.

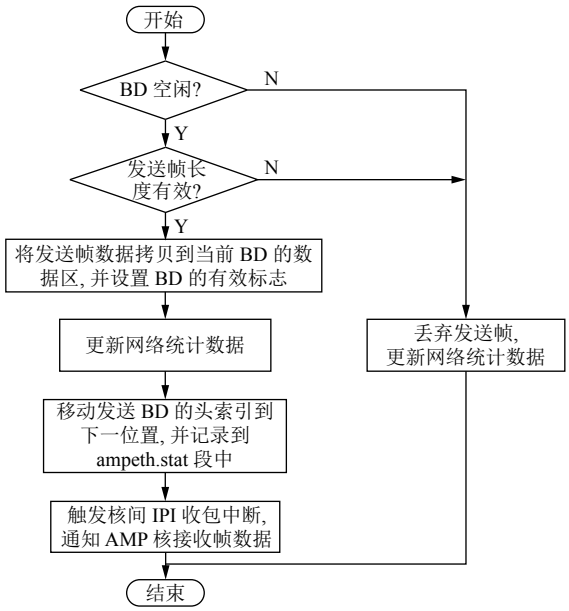

图 3 核间通信虚拟网口发送流程

AMP 核间通信虚拟网口的接收操作流程如图 4 所示. 当 AMP 核接收到 IPI 收包中断, 进入中断处理函数后, 首先关闭核间 IPI 收包中断, 然后唤醒收包线程进行收包. 为了避免大量的网络流量造成中断风暴, 从而严重影响系统性能, 我们在进行收包前先关闭 IPI 收包中断, 当有大量数据包进来后, 通过收包线程进行轮询收包, 收完所有有效包后再打开 IPI 收包中断, 同时为了避免收包线程长时间收包占用处理器资源, 影响其他任务, 对轮询收包的个数做了配额限制, 收包机制类似于 Linux 内核 NAPI 收包机制[21].

收包线程唤醒后, 首先检查轮询收包的个数是否超出了配额限制, 如果轮询收包的个数超过 64 个, 则使能核间 IPI 中断后等待下次唤醒. 如果收包个数未超过配额限制, 再检查当前接收 BD 是否空闲, 如果空闲, 说明接收缓存中的所有数据已全部接收, 使能核间 IPI 中断后等待下次唤醒. 如果 BD 非空闲, 说明接收 BD 中有数据, 再检查当前接收 BD 的 len 字段是否有效, 如果无效则丢弃接收帧, 否则将 BD 的数据拷贝到接收内存空间, 并清除 BD 的 valid 和 len 字段. 然后更

新网络统计计数, 并移动接收 BD 的头索引到下一位置, 将新的接收 BD 头索引更新到 ampeth.stat 段中. 至此, 一个 BD 的收包处理完成.

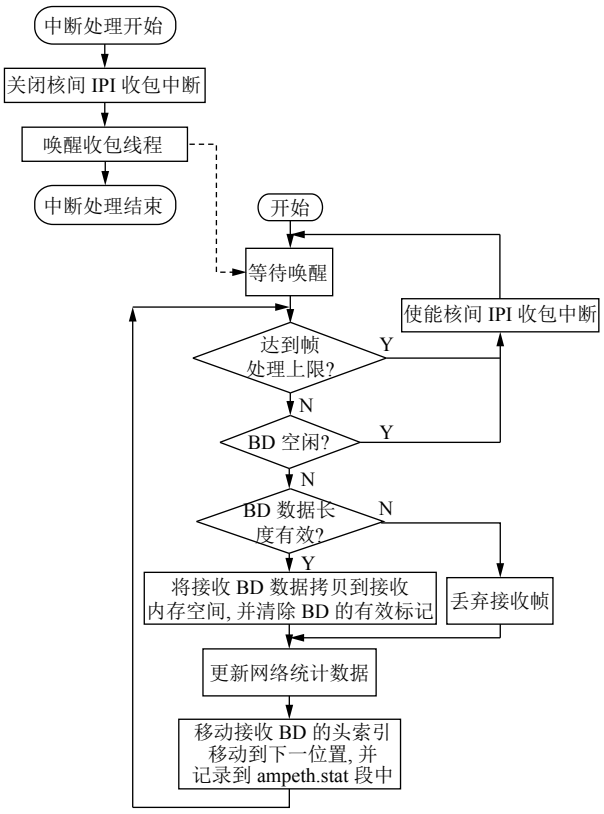

图 4 核间通信虚拟网口接收流程

通过 AMP 核间通信虚拟网口, AMP 主核可以借助完善的 TCP/IP 网络服务, 为从核提供更加丰富复杂的网络服务, 如网络文件系统、FTP、ssh 等; 另外传统微机保护测控装置采样分布式硬件架构, 其板间通信以网络通信为主, 使用虚拟网口实现 AMP 核间通信, 可以很好地兼容现有微机保护测控装置的嵌入式软件架构和通信机制.

4 工程应用

本方案应用于 35kV 及以下低压保护测控装置的研制, 装置中的 CPU 板的核心器件包括四核处理器和 FPGA. 处理器中的 3 个 CPU 核设置为 SMP 模式, 运行 Linux 操作系统; 1 个 CPU 核设置为 AMP 模式, 运行实时操作系统. FPGA 用于扩展电力专用网络 (如 MMS 制造报文规范、GOOSE 面向通用对象变电站事件网、SV 采样网等) 和总线扩展 (如开关量输入输出模块管理总线、快速报文总线、对时总线、人机接口模块通信总线、打印通信总线等). 装置保护动作响应时间小于 10 ms, 遥控传输延时小于 1 s, 遥信变化响应时间小于 1 s, 通过保护测控装置的实时性能和功能测试[5,6]. 在网络报文攻击和故障注入等异常测试中, 装置无死机或异常重启, AMP 从核上的保护、采样等实时任务可靠运行.

装置在正常负载情况下, AMP 核间虚拟网口网络通信耗时约 50 μs 左右, 其中 Linux 侧收发包平均耗时约 40 μs, RTOS 侧收发包平均耗时约 10 μs, 而分布式多板卡方案中使用的百兆以太网口网络通信耗时约 500 μs 左右. AMP 主从核通过 AMP 核间虚拟网口实现的网络文件系统、远程登录、端口转发、网络对时等网络服务功能工作稳定, 性能符合装置技术指标要求.

现有继电保护装置的 AMP 方案都是采用 Zynq SOC 芯片或异构处理器, 而不支持通用多核处理器. 本方案的应用结果表明基于通用多核处理器的 AMP 方案是行之有效和稳定可靠的.

5 结语

本文提出一种基于通用多核处理器的 AMP 设计方案, 支持 AMP 核的生命周期管理和基于虚拟网口的核间通信. 采用本方案研制的低压保护测控装置已批量发货, 现场运行情况表明基于通用多核处理器的 AMP 方案, 能够满足电力二次设备的实时性能要求. 总结来说, 本方案的优势有如下几点:

(1) 简化软硬件设计, 节约成本;

(2) 扩大处理器选型范围; 相对于异构处理器, 通用多核处理器的选型范围广, 成熟度高;

(3) 基于虚拟网口的 AMP 核间通信方案, 兼容现有的分布式软件架构和网络通信机制.

后续计划在电力二次设备中应用本方案, 并适配到更多的多核处理器和实时操作系统中.

参考文献

1 McKenney PE. Is Parallel programming hard, and, if so, what can you do about it? http://kernel.org/pub/linux/kernel/people/paulmck/perfbook/perfbook.2019.12.22a.pdf. [2019-12-22].

2 黄国睿, 张平, 魏广博. 多核处理器的关键技术及其发展趋势. 计算机工程与设计, 2009, 30(10): 2414–2418.

3 Hennessy JL, Patterson DA. Computer architecture: A quantitative approach. Waltham: Morgan Kaufmann

Publishers, 2012. 17–26.
4 张吉赞, 苑雅娟. 多核共享资源冲突延迟上限优化方法. 计算机科学与探索, 2017, 11(8): 1224–1234. [doi: 10.3778/j.issn.1673-9418.1701043]
5 国家能源局. DL/T 1075-2016 保护测控装置技术条件. 北京: 中国电力出版社, 2017.
6 中华人民共和国国家质量监督检验检疫总局, 中国国家标准化管理委员会. GB/T 30155-2013 智能变电站技术导则. 北京: 中国标准出版社, 2014.
7 周晓龙. 智能变电站保护测控装置. 电力自动化设备, 2010, 30(8): 128–133. [doi: 10.3969/j.issn.1006-6047.2010.08.028]
8 黎强, 李延新. 基于数字化变电站的系统保护装置设计. 电力系统自动化, 2009, 33(18): 77–81. [doi: 10.3321/j.issn:1000-1026.2009.18.016]
9 张建华. 变电站自动化技术的发展综述. 大科技, 2013, (22): 170–171.
10 李响, 刘国伟, 冯亚东, 等. 新一代控制保护系统通用硬件平台设计与应用. 电力系统自动化, 2012, 36(14): 52–55.
11 丁毅, 陈新之, 潘可, 等. 基于电力专用多核异构芯片架构的低压保护测控装置设计. 南方电网技术, 2020, 14(1): 58–64.
12 周华良, 夏雨, 汪世平, 等. 多核处理器在中低压保护测控一体化装置中的应用. 电力系统自动化, 2011, 35(24): 84–88.
13 王海燕, 徐云燕, 王世云, 等. 一种基于 DSP+MPC 的数字化保护测控装置. 电力系统自动化, 2010, 34(9): 112–115.
14 吴相楠, 龚行梁, 周强, 等. 双核处理器 AMP 模式在电力设备控制中的应用. 单片机与嵌入式系统应用, 2018, 18(6): 38–41.
15 邢艳芳, 朱金付, 周晓梅. 基于 Zynq 多核运行设计. 计算机技术与发展, 2018, 28(3): 60–62, 66. [doi: 10.3969/j.issn.1673-629X.2018.03.012]
16 李鑫志, 戈志华, 刘向明. 基于 ARM 平台 AMP 架构下从核重复加载设计与实现. 计算机应用与软件, 2017, 34(1): 218–221. [doi: 10.3969/j.issn.1000-386x.2017.01.040]
17 McDougall J. Simple AMP running Linux and bare-metal system on both Zynq SoC processors. https://www.xilinx.com/support/documentation/application_notes/xapp1078-amp-linux-bare-metal.pdf. (2013-02-14).
18 Multicore Association. Multicore Communication API 2.015 (MCAPI) Specification. Multicore Association, 2015.
19 Xilinx. Libmetal and OpenAMP user guide. https: //www.xilinxcom/support/documentation/sw_manuals/xilinx2019_1/ug1186-zynq-openamp-gsg. pdf. [2019-05-22].
20 高珂, 陈荔城, 范东睿, 等. 多核系统共享内存资源分配和管理研究. 计算机学报, 2015, 38(5): 1020–1034.
21 姚萌萌, 张俊, 沈亮. Linux 多核环境网卡驱动优化研究. 计算机系统应用, 2014, 23(10): 223–227. [doi: 10.3969/j.issn.1003-3254.2014.10.040]