

动态可重构高速串行总线中断请求方法的设计与实现

万玛宁¹, 王莹¹, 周继芹^{1,2}, 张伟功¹

(1 首都师范大学 信息工程学院, 北京 100048;

2 首都师范大学 数学科学学院, 北京 100048)

摘 要: 动态可重构高速串行总线(UM-BUS)具有多通道并发冗余的特征以及远程存储访问能力,然而传统的固定网络结构已不能适应 CPS、物联网对异构资源动态接入、多源感知融合的需求. 为解决 UM-BUS 总线网络中功能模块的即插即用和实时响应,本文提出一种低开销、高效率的中断请求与仲裁方法,采用基于编码仲裁的方式实现总线节点设备的中断处理,满足 CPS、物联网应用环境下,UM-BUS 总线节点设备之间事件中断通知与快速实时响应的需求.

关键词: 动态可重构总线;中断请求;CPS;物联网

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2018)10-0023-06

Design and Implementation of Interrupt Request Method for Dynamic Reconfigurable High-speed Serial Bus

WAN Ma-ning¹, WANG Ying¹, Zhou Ji-qin^{1,2}, ZHANG Wei-gong¹

(1 College of Information Engineering, Capital Normal University, Beijing 100048, China;

2 School of Mathematical Sciences, Capital Normal University, Beijing 100048, China)

Abstract: Dynamic reconfigurable high-speed serial bus (UM-BUS) has the characteristics of concurrent lanes redundancy and remote memory accessing capabilities. However, traditional fixed network structure does not meet the requirements of dynamic access and multi-source information fusion for the heterogeneous resources of CPS and Internet of things. To solve the problem of plug and play and real-time response of UM-BUS modules, this paper presents a method of interrupt request and arbitration with low cost and high efficiency. It adopts encoding arbitration to fulfill the interrupt handling of UM-BUS devices. This method can satisfy the requirements of event interrupt notification and real-time response between UM-BUS devices in the CPS or Internet of things application environment.

Key words: dynamic reconfiguring bus; interrupt request; CPS; IoT

1 引言

信息物理系统(Cyber-Physical Systems, 简称为 CPS)是通过计算、通信与控制技术的有机结合,将信息处理与物理感知执行深度融合,实现计算资源与物理资源协调工作的新一代智能系统^[1-2].

CPS 强调计算和物理的紧密结合,同时也强调网络化,核心仍然是信息处理. 数据感知则是 CPS

实现实时分析、科学决策的基础,是 CPS 数据闭环流动的起点^[3].

目前 CPS、物联网感知数据的一个主要途径是依靠无线传感器网络进行数据采集. 然而,在大量的工业生产现场、嵌入式控制等领域,由于受到噪声、信号衰减、报文冲突等因素的制约,无线传感器网络在实时性、精准性、可靠性等方面难以满足应用要求. 同时,各种无线网络及高速总线,大多缺乏设备

间的中断支持,难以给 CPS、物联网应用中设备间事件快速实时响应提供良好支持^[4-6].

动态可重构高速串行总线(UM-BUS)^[7]是针对嵌入式系统小型化与一体化设计提出的一种能够将冗余容错与高速通信有机统一,具备远程扩展能力的高速串行总线.

UM-BUS 总线可为 CPS、物联网中传感器与执行单元的即插即用、数据方法属性封装、高速可靠连接、异构实时接入等提供技术支撑.针对 UM-BUS 总线的特点及 CPS、物联网的应用需求,本文提出一种适于 UM-BUS 总线结构的低开销、高效率的中断请求与仲裁方法.

2 动态可重构高速串行总线协议

UM-BUS 总线采用基于 M-LVDS(TIA/EIA-899)的多通道智能动态冗余的总线型拓扑结构,支持最多 30 个节点直接连接到总线上,而不需要通过中继器或者路由器进行扩展,其网络拓扑如图 1 所示.UM-BUS 通过多个通道(2~32 通道)并发的方式进行数据传输,单通道速率达到 200 Mbps,在最高 32 通道配置下,总线通信速率最高可达 6.4 Gbps;同时各个通道又互为冗余结构,一旦某个通道出现故障,可通过通道动态冗余及故障重构技术自动屏蔽故障通道,在剩余健康通道上继续通信.这种传输方式使得 UM-BUS 既能达到很高的数据传输速率,又能获得良好的容错能力.

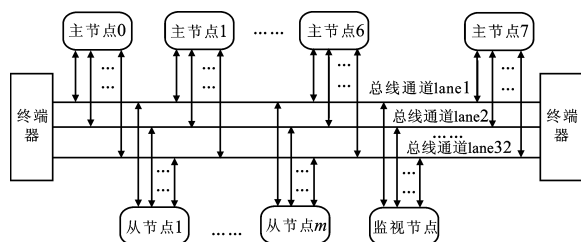


图 1 UM-BUS 总线型拓扑结构

UM-BUS 总线采用主从命令应答的通信模式,可为系统提供远程存储访问及非智能扩展能力.连接在总线上的通信节点按功能不同可分为主节点、从节点及监视节点三种类型,节点间通过数据包的形式交互信息,一次通信过程只能由主节点发起,并且由从节点或其它主节点响应,监视节点用于监视总线上的通信过程.UM-BUS 总线主控节点可以对其它节点内部功能单元按地址读写访问,可支持 IO 空间、存储器空间和配置空间三种地址空间.UM-BUS 总线具有时间同步功能,可保证总线在工作过

程中所有节点都处于精确时间同步状态^[8].

UM-BUS 总线通信协议分为处理层、数据链路层和物理层三个层次,如图 2 所示.其中处理层负责对整个总线的管理、协议封装和对上层应用接口的转换,通过地址变换管理协议,可支持主机端通过远程存储访问方式对远端从节点设备进行内部存储单元的直接访问.数据链路层又分为传输子层和链路管理子层两部分,传输子层根据现存的有效线路对数据进行分组和动态重构;链路管理子层负责通信线路检测,向传输子层提供通道健康状况,完成对通道传输信息进行二次打包和解包,实现总线节点的时间同步.物理层是协议的最底层,它为数据通信提供传输媒体及互连设备,实现了网络的物理连接、完成了串并转换、8b/10b 编解码、时钟同步等功能,为总线提供可靠的通信基础.

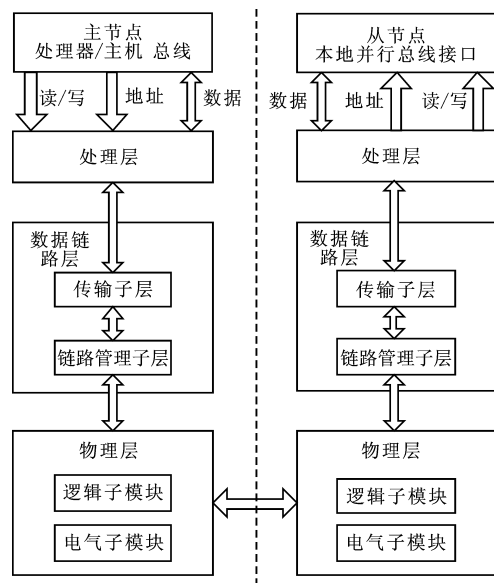


图 2 UM-BUS 总线协议层次结构

3 UM-BUS 总线中断请求方法

为满足 UM-BUS 总线在 CPS、物联网等场景中的应用需求,提高总线节点事件通知的实时性和可靠性,总线各节点采用专门的共享中断信号线进行中断申请,在设计过程中,需要解决以下问题:

(1)UM-BUS 总线最多可连接 30 个节点,在同一时刻,可能存在多个节点发出中断请求,需要解决总线上多个节点通过共享中断信号线向主节点提起中断申请的仲裁问题.

(2)总线节点通过中断请求逻辑电路发送中断请求信号到共享信号线上,为提高总线中断请求效率与中断处理的实时性,需要对中断请求逻辑电路

进行合理设计,从而提供一种高效的中断信息传送方法.

(3)UM-BUS 总线通过共享中断信号线和中断请求逻辑电路的设计解决总线上多个节点同时向主节点进行中断申请的问题,在此基础上,还需对中断请求发送和处理流程进行优化设计,提高中断请求的效率和可靠性.

3.1 中断请求仲裁方法设计

UM-BUS 总线在所有节点之间,设置专门的共享中断信号线.如图 3 所示,总线所有节点均可以通过 OC 型输出驱动^[9],采用串行编码方式发送中断请求信号到共享中断信号线上,向总线主节点提起中断请求;同时,所有节点也可以从共享中断线上接收信号,获取中断线上传输的信息.这样,通过中断请求向量的编码发送方式,总线各节点就能通过信号线与方式进行中断优先级仲裁,解决总线上多个节点在共享中断信号线向主节点进行中断申请的问题.

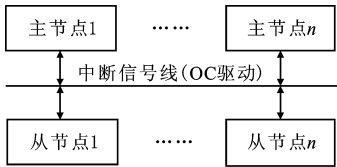


图 3 UM-BUS 总线中断信号线连接示意图

中断请求向量设计如图 4 所示,由起始位、优先级选择、中断事件、源节点、目标节点和校验位组成.其中起始位固定为 0,表示一个中断请求向量的开始;优先级选择为 1 位,为 0 表示高优先级,为 1 表示低优先级;中断事件是以编码方式表示的中断请求原因,共 4 位,可以表示 0~15 共 16 种事件;源节点号为发出中断请求的节点编号,共 5 位,按 UM-BUS 总线规定取值为 1~30;目标节点号为接收中断请求进行中断处理的主节点编号,共 5 位,取值 1~7,对应主节点 1~7,取值为 0 时,表示所有主节点;校验位占 1 位,由优先级选择、中断事件、源节点号和目标节点号的所有位按位异或生成.当目标主节点收到一个中断请求向量时,将优先级选择、中断事件、源节点、目标节点和校验位按位进行异或,如果为 0 表示中断向量校验正确,否则校验错误.

起始位	优先级选择	中断事件	源节点	目标节点	校验
-----	-------	------	-----	------	----

图 4 中断请求向量组成

中断请求向量在发送时,按照起始位、优先级选

择、中断事件、源节点号、目标节点号、校验位的顺序,逐一发送,最先发送起始位,最后发送校验位,中断事件、源节点号、目标节点号发送时高位在前,低位在后.

在中断请求向量发送过程中,由于各节点均采用 OC 型输出驱动,如果多个节点同时产生中断请求,进行中断请求向量发送,各节点输出的中断请求向量不断进行逻辑线与操作.对于每一个输出数据位,如果各节点输出值相同,所有节点都继续输出;如果不同,则只有输出值为 0 的节点才能继续输出,输出值为 1 的节点则不再输出.这个过程其实就是一个优先级仲裁的过程,即编码小的事件具有更高的优先级,事件相同时,编码小的节点具有更高的优先级.

3.2 中断请求逻辑电路设计

基于上述中断请求仲裁方法设计,在 UM-BUS 总线节点控制器内部设置一个中断请求逻辑电路,用来生成中断请求向量,并控制中断请求向量的发送.如图 5 所示,该中断请求逻辑电路包括向量构造器、串行发送器、驱动器和向量检测器.

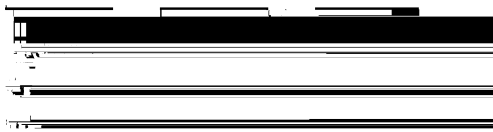


图 5 总线节点内部中断请求逻辑组成结构

图 5 中,向量构造器在本节点有中断请求时,根据中断请求的原因、优先级要求、目标节点等构造中断请求向量,送串行发送器发送;串行发送器根据向量检测器送来的中断信号线空闲指示,将中断请求向量,按 1 Mbps 的速率,转为串行数据流,送驱动器输出;驱动器为两个 OC 输出型双向驱动电路,将串行发送器送来的串行数据流输出到两条冗余的中断信号线上,并将两条中断信号线的电平值相与后,送向量检测器.向量检测器主要完成以下两个功能:

(1)中断信号线空闲检测,生成中断信号线占用标志给串行发送器.在空闲态下,当中断信号线变为低电平,表示有节点占用中断信号线发送中断请求向量,生成中断信号线占用标志.当一个中断请求向量发送完成后,撤消中断信号线占用标志.当中断信号线长期为高电平时,认为中断信号线处于空闲状态,将中断信号线占用标志置为无效;

(2)信号回读检测,串行发送器每发送 1 位数据后,在 500 ns 后,将发送的数据与驱动器送来的中

断信号线电平进行比较,如果不一致,说明有其它节点也同时在发送具有更高优先级的中断请求,生成指示信号送串行发送器,终止本节点的中断请求向量发送过程。

3.3 中断请求处理流程设计

UM-BUS 总线上的节点通过共享中断信号线,使用图 5 所示的中断请求逻辑电路发送中断请求,可向一个或多个总线主节点提起中断申请,中断请求发送流程设计如图 6 所示,具体方法与步骤如下:

(1) 当一个节点需要向主节点申请中断时,首先根据中断事件,由向量构造器构造一个中断请求向量,然后转步骤(2)。

(2) 利用向量检测器检测中断信号线是否被其它节点占用,如果中断信号线未被其它节点占用,直接转步骤(3)进行中断向量发送,否则等待中断信号线被其它节点释放后,再转步骤(3)进行中断向量发送。

(3) 由串行发送器,按 1 Mbps 的速率,以串行方式向共享中断信号线发送中断请求向量。在发送过程中,向量检测器持续监测中断信号线上的状态,对中断信号线上状态进行回读检测。如果中断信号线上的状态值与本节点发送值不相同,说明有更高优先级的中断请求也在由其它节点同时发送,本节点串行发送器立即停止中断请求向量的发送,转步骤(2),等待其它节点释放中断信号线后重新开始发送;如果中断信号线上的状态值与本节点发送值相同,本节点持续发送中断请求向量。中断请求向量正确发送完成后,转步骤(4);

(4) 中断请求向量发送完成后,延迟约定的时间,如果中断请求事件消失,表示主节点进行了中断响应,则结束中断请求过程;否则,可能是主节点未响应中断请求,也可能是中断请求向量传送错误,也可能是产生了新的中断事件,转步骤(1)重新开始中断请求过程。

总线上的一个主节点在收到中断请求向量后,首先对中断请求向量进行校验,校验正确后,如果中断请求向量的目标节点号与该主节点的节点号相符,说明该中断请求是给本节点的,则向处理器发送中断请求,并将中断源节点号、中断事件、优先级选择及可能的数据放入相应的寄存器,供处理器读取处理。如果中断请求向量的目标节点号是 0,则认为这是一个全局中断请求,所有的主节点都需要响应,同样向处理器发送中断请求,并将中断源节点号、中断事件、优先级选择及可能的数据放入相应的寄存

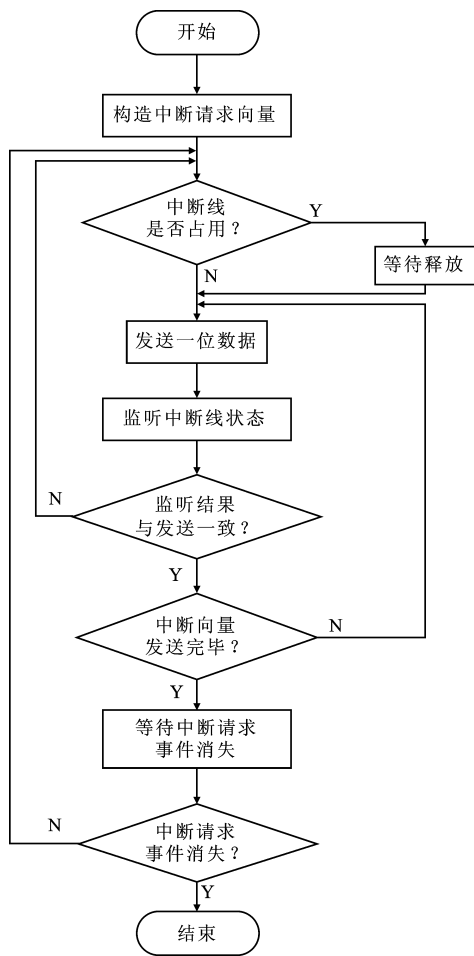


图 6 中断请求发送流程

器,供处理器读取处理。如果中断请求向量的目标节点号不是 0,也与该主节点的节点号不相符,则忽略该中断请求。

处理器对中断请求的处理,可根据中断请求源节点号、中断事件、中断优先级以及附加数据进行。基于具体的设计约定,先读取发起中断请求节点的属性空间、IO 空间进一步查询中断原因,再通过读写其属性空间、IO 空间、存储空间进行相应的中断处理。

4 验证及结果

4.1 验证系统

为了验证 UM-BUS 总线中断请求方法的正确性和有效性,搭建包含 1 个主节点和 6 个从节点的测试系统,节点之间通过包含 8 通道 UM-BUS 总线和中断信号线的总线线缆进行连接。测试系统的组成结构如图 7 所示。

在图 7 的测试系统中,主节点的节点号定义为 1,各从节点的节点号分别定义为 11~16。总线中断

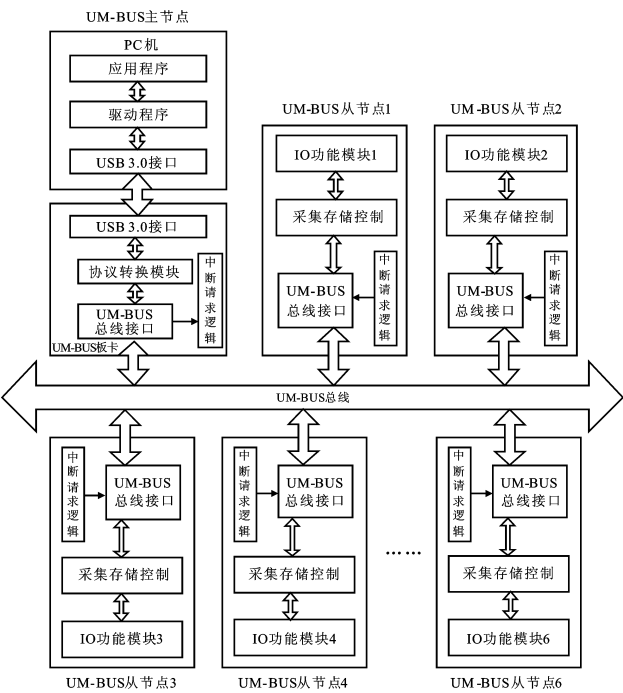


图 7 测试系统组成结构

信号线数据传输速率为 1 Mbps,即各节点中断请求信号串行转换时钟为 1 MHz,传送 1 个数据位的时间为 1 000 ns;总线长度为 20 m,距离最远的两个节点间信号传送时间小于 100 ns. 同时,UM-BUS 总线上所有节点处于时间同步状态,各节点中断请求信号串行转换时钟同步误差不大于 100 ns.

4.2 功能测试

基于上述测试系统,令各从节点发送中断请求给主节点,不同从节点设置不同的中断优先级或中断事件. 根据前文中断请求方法的设计,多个节点同时申请中断时,从节点的中断请求逻辑通过比较中断请求向量中各数据位的大小完成优先级仲裁,按照优先级的高低依次完成不同节点中断请求向量的发送. 在测试系统中,通过观察主节点上层软件响应中断的先后次序,可判断中断请求和优先级仲裁功能的正确性. 测试用例设计和测试结果如下:

(1)中断请求功能验证:分别测试 11~16 号节点发送中断请求时主节点的响应情况. 经过测试,各从节点在发送中断请求后,主节点都能接收到中断,提交给上层软件进行处理.

(2)优先级仲裁功能验证:测试 11、14 号节点同时产生中断时主节点的响应情况,分别设置不同的优先级进行两组测试.

①第一组测试:11 号节点发送中断请求向量为 00010101011000010 的中断,14 号节点发送中断请

求向量为 01000101110000010 的中断,观察主节点软件响应中断的次序. 经过测试,主节点先接收到 11 号节点的中断请求,之后接收到 14 号节点的中断请求;

②第二组测试:11 号节点发送中断请求向量为 00010001011000011 的中断,14 号节点发送中断请求向量为 00001001110000011 的中断,观察主节点软件响应中断的次序. 经过测试,主节点先接收到 14 号节点的中断请求,之后接收到 11 号节点的中断请求.

4.3 中断延迟测试

然后分别测试单个从节点、两个从节点、以及六个从节点同时发送接入中断请求的情况下,中断请求发送与接收的延迟时间. 测试时各节点的中断请求向量中的起始位、中断事件均相同,从节点 11 和 16 采用高优先级中断,12~15 号从节点采用低优先级中断. 因此,对中断的仲裁依靠优先级选择和源节点号进行,优先级相同时编号小的节点优先发送中断请求. 经过实物测试,总线测试系统的中断请求测试结果如表 1 所示.

表 1 测试结果

发送中断 节点号	接收中断 节点号	中断发送 延时/ μs	中断接收 延时/ μs	总延时 / μs
11	11	17	18	18
12~13	12	17	18	18
11~16	15	107	108	108
11~16	16(高优先级)	35	36	36

测试结果表明中断请求方法设计正确,且能取得较高的运行效率. 最坏情况下,当 6 个从节点同时发送请求时,优先级最低的 15 号从节点的中断延迟时间只有 108 μs .

5 结束语

本文提出一种适用于动态可重构高速串行总线的中断请求方法,通过共享中断信号线和中断请求逻辑电路的设计,实现基于编码仲裁的方式进行总线节点设备的中断处理,能够解决动态可重构总线上中断请求的快速传输、识别及优先权仲裁问题,减少总线节点对中断原因进行判别的查询时间,从而提高总线中断传输与处理的效率与实时性.

参考文献:

[1] 李仁发,谢勇,李蕊,等. 信息-物理融合系统若干关

键问题综述[J]. 计算机研究与发展, 2012, 49(6): 1149-1161.

[2] Huang S, Zhou C J, Yang S H, et al. Cyber-physical system security for networked industrial processes[J]. International Journal of Automation & Computing, 2015, 12(6):567-578.

[3] Edward A Lee. CPS foundations[C]//Design Automation Conference. San Diego, California, USA, ACM, 2010:737-742.

[4] 钱志鸿, 王义君. 面向物联网的无线传感器网络综述[J]. 电子与信息学报, 2013, 35(1):215-227.

[5] Liu Y, Dong M, Ota K, et al. Active trust: secure and trustable routing in wireless sensor networks[J]. IEEE Transactions on Information Forensics & Security, 2017, 11(9):2013-2027.

[6] 郑国雄. 基于 SERCOS-III 网络系统高精度同步机制的研究[J]. 物联网技术, 2015, 4(2):40-42.

[7] 张伟功, 周继芹, 李杰, 等. UM-BUS 总线及接入式体

系结构[J]. 电子学报, 2015, 43(9):1776-1785.

[8] 李超, 邱柯妮, 张伟功, 等. 一种新型的动态可重构总线时间同步方法[J]. 微电子学与计算机, 2016, 33(7):60-63.

[9] 成立, 朱漪云, 王振宇, 等. 高速低耗 BiCMOS OC 门及其线与逻辑系统[J]. 江苏大学学报(自然科学版), 2007, 28(2):156-159.

作者简介:

万玛宁 女, (1981-), 硕士, 讲师. 研究方向为通讯总线、高可靠嵌入式系统. E-mail: wanmaning@cnu.edu.cn.

王莹 女, (1984-), 硕士. 研究方向为高可靠嵌入式系统与冗余容错技术.

周继芹 女, (1978-), 博士研究生. 研究方向为计算机系统体系结构与性能评价.

张伟功 男, (1967-), 博士, 研究员. 研究方向为高可靠嵌入式系统.

(上接第 22 页)

对比图 6 和图 7 的数据, 可以知道在实现双核 cache 一致性的情况下, 可以从 core0 核正确获取 core1 核的状态信息, 以反映系统的运行状态.

5 结束语

本文考虑到多核系统的任务不仅在本核上运行, 还会涉及到核间信息的交互, 采用软件实现的核间通信机制, 在某个内核异常的时候, 并不能有效的传递出该核的任务信息, 导致任务监测的失效. 本文提出基于 cache 一致性的多核任务监测机制以解决该问题. 并通过测试用例证明了该方法的可行性, 该方法可以避免多核间复杂的通信机制, 基于硬件提供的便利, 既降低了系统的复杂性, 也为异常情况下, 仍然能有效正确的获取核上的任务信息提供了便利, 为任务管理和异常处理提供了监测手段.

参考文献:

[1] 王欣, 邱昕, 慕福奇, 等. 基于 MCAP1 的嵌入式多核通信机制的研究[J]. 微电子学与计算机, 2017, 34(11): 85-88.

[2] 蔡玉鑫. 嵌入式多核处理器核间通信方法的设计与实现[D]. 西安: 西安电子科技大学, 2015.

[3] 郭海林, 陈香兰. 一种支持多级存储架构的嵌入式多核

操作系统模型[J]. 计算机系统应用, 2015, 24(12): 110-117.

[4] 孙荪. 提高多核处理器片上 Cache 利用率的关键技术研究[D]. 合肥: 中国科学技术大学, 2015.

[5] 刘文佳. 多核处理器 Cache 一致性研究与设计[D]. 哈尔滨: 哈尔滨工程大学, 2010.

[6] 黄安文, 张民选. 多核处理器 Cache 一致性协议关键技术研究[J]. 计算机工程与科学, 2009, 31(S1): 104-108.

[7] Dalui M, Som T, Bansal S, et al. Handling cache write-back and cache eviction for cache coherence [P]. WO2014022397, 2014-02-06. United States Patent Application 20140040561.

[8] Amer Diwan, David Tarditi, Eliot Moss. Memory system performance of programs with intensive heap allocation [J]. ACM Transactions on Computer Systems (TOCS), 1995, 13(3):244-273.

作者简介:

陈慧 女, (1993-), 硕士研究生. 研究方向为嵌入式系统. E-mail: chenhui@ciotc.org.

慕福奇 男, (1957-), 研究员, 博士生导师. 研究方向为无线通信.

张然 男, (1988-), 硕士, 工程师. 研究方向为无线通信.

姜彬 男, (1989-), 硕士, 工程师. 研究方向为嵌入式.