嵌入式微处理器多协议 ISP 设计及安全性分析

刘根贤[1,2],汪东升[3],王海霞[3]
(1. 清华大学 计算机科学与技术系,北京 100084;
2. 第二炮兵工程大学 信息管理中心,陕西 西安 710025;
3. 清华大学 信息科学技术国家实验室(筹),北京 100084)

摘 要:ISP(In-System Programming,在系统可编程)指电路板上的微处理器可以通过预先设定的接口对内部 FLASH 存储器进行编程,而不需要从电路板取下器件。ISP 技术已经得到广泛的应用,越来越多的微处理器支持用户自行设计 ISP 程序。但 ISP 存在升级方式单一的问题,还隐含一些安全风险,例如误触发擦写操作造成程序丢失,或者恶意 ISP 程序窃取敏感信息。针对这些问题,文中提出了多协议 ISP 设计,分析了 ISP 安全的硬件支持条件,并进行了相应评估,验证了多协议 ISP 以及安全设计的可行性。

关键词:在系统可编程;微处理器;安全;多协议
中图分类号:TP39 **文献标识码**:A **文章编号**:1673-629X(2015)08-0020-04
doi:10.3969/j.issn.1673-629X.2015.08.004

Design of Multi-protocol ISP and Its Safety Analysis for Embedded Microprocessor

LIU Gen-xian[1,2], WANG Dong-sheng[3], WANG Hai-xia[3]
(1. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China;
2. Information Management Center, The Second Artillery Engineering College, Xi'an 710025, China;
3. Tsinghua Laboratory for Information Science and Technology, Beijing 100084, China)

Abstract: In-System Programming programs the inner FLASH of the microprocessor through a specific interface when the device is mounted on the circuit board. ISP technology has been used widely in microprocessors, some of which support users defined ISP firmware. However, ISP faces upgrading problems as well as security problems, such as the program loss caused by false trigger or the sensitive information missing caused by malicious ISP program attack. To solve those problems, a multi-protocol ISP technology is proposed. In addition, analyze and assess the hardware support requirements for ISP security and prove the feasibility of multi-protocol ISP and security design.

Key words: ISP; microprocessor; security; multi-protocol

0 引 言

微处理器在系统可编程指电路板上的微处理器可以通过预先设定的通讯接口对程序 FLASH 存储器进行在线编程,ISP 技术的优势是不需要专用编程器就可以通过上位机对器件进行在线编程,因此芯片可以直接焊接到电路板上,避免调试时频繁地插入取出芯片对器件和电路板带来的损坏。

一般具有 ISP 功能的微处理器可以由上位机软件通过串口来进行升级,也可以通过 USB、SPI 或其他接口来接收上位机下载的数据并写入程序存储器中,所以只要留出和上位机通讯的这个接口,就可以对器件擦写编程[1]。

ISP 技术已经在微处理器开发中得到广泛应用[2]。但目前 ISP 还存在升级方式单一问题,使用中

收稿日期:2014-09-01 **修回日期**:2014-12-10 **网络出版时间**:2015-07-21
基金项目:国家自然科学基金资助项目(61303002,61373025);国家"863"高技术发展计划项目(2012AA0100905);清华信息科学与技术国家实验室(筹)学科交叉基金
作者简介:刘根贤(1974-),男,博士研究生,研究方向为计算机体系结构、嵌入式系统;汪东升,教授,博导,研究方向为计算机体系结构。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20150721.1433.012.html

也隐含一些安全风险,包括误触发擦写造成程序丢失;存在恶意 ISP 程序窃取敏感信息。针对这些问题,设计了多协议 ISP,此外分析了 ISP 程序安全的硬件支持必要条件,并进行了相应评估,证明多协议 ISP 程序以及硬件安全设计的可行性。

1　ISP 概述

微处理器出厂时预先烧写 ISP 程序,这样在系统加电或复位时可以通过预先设定的通讯接口和通讯协议对片内集成的或板载程序 FLASH 存储器进行在线编程,而不需要从电路板上取下器件[3]。

1.1　ISP 功能及类似技术

ISP 是指利用上位机程序通过预定接口和通讯协议对电路板上微处理器直接进行编程,需要微处理器中预先固化的 ISP 程序的配合,可以对整个应用程序、配置空间或数据空间进行擦写编程。

与 ISP 在系统编程技术相似的还有在应用编程 IAP(In Applicating Programming)和在电路编程 ICP(In Circuit Programming)[4-5]。IAP 指微处理器可以在运行中对自己重新编程,修改同一地址空间的内容,即可用程序来改变程序。IAP 虽然也是在板上进行编程,却是自己对自己进行。一般说来 IAP 的对象在操作时是作为数据进行读写的。ICP 指通过完全硬件逻辑实现对器件的在板编程,器件可以是全新器件,不需要预先编程。例如通过 JTAG 口下载程序就是 ICP 的一种特例,ARM 公司设计的 SWD 也具有 ICP 功能。ICP 需要专用硬件支持,通常需要硬件厂商提供上位机软件和专用下载工具[6]。

文中 AP 指应用程序,即真正实现设备功能的程序部分,IAP 是指 AP 在运行中可被读写的数据,与 AP 代码分离存储,ISP 则是加电复位时先执行的,具有在线升级功能的引导程序。

1.2　ISP 代码安全

ISP 程序可以擦写用户应用程序代码,因此一旦在非设定条件下触发 ISP 操作往往造成应用程序执行代码损坏。系统重新上电后会出现程序崩溃,不能正常运行,即程序丢失现象。对于所有包含 FLASH 擦除或写操作的程序,都有可能出现程序执行代码或数据错误的现象,程序设计的逻辑缺陷是数据损坏或程序丢失的主要原因。

另外也有人为恶意 ISP 代码,通过 ISP 程序篡改应用程序,窃取用户敏感数据或应用程序代码本身,因此 ISP 程序的安全性不容忽视,但目前大多数处理器并没有针对 ISP 程序安全提供特别的支持。一般说来 ISP 程序对系统具有完全控制权限,可以访问整个 FLASH 程序空间和内存数据空间,系统隐含较大的安全风险,此外 ISP 程序的擦除或写操作触发条件简单,误触发 ISP 操作(对 FLASH 进行擦写)的可能性较大。

2　ISP 安全的处理器硬件支持

处理器支持 ISP 功能,首先要求 ISP 程序必须可以先于应用程序启动,ISP 程序一般隐含在程序空间顶端,用户不可访问,因此需要硬件支持复位时从 ISP 起始地址执行或者支持硬件设置复位运行地址[7]。

ISP 程序执行在线编程过程中需要对程序空间进行擦写,为了避免误触发操作,要求用于执行 FLASH 擦写操作的 ISP 命令采用序列触发方式,必须按次序在设定的时间间隔内依次写入触发命令字。如图 1 所示,需要依次连续写入 0x11,0x22,0x33,0x44 到命令寄存器才能触发 ISP 操作。

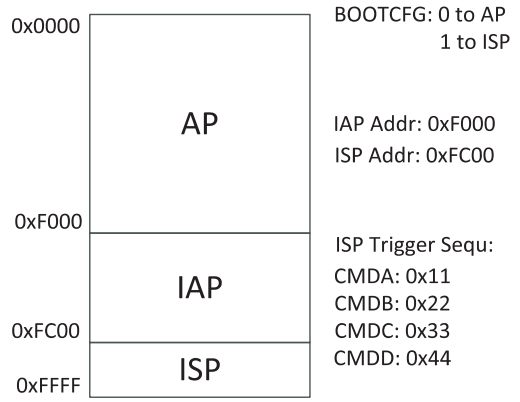

图 1　程序空间及 ISP 相关设置

为了确保应用程序代码自身安全,需要建立程序代码区域分级保护机制,按照程序重要性的差别,一般认为安全性 ISP 代码>AP 代码>IAP 数据,任何区域的程序都不能修改自身程序代码以及高于自身安全级别区域的程序代码,而只能修改低级安全级别的区域代码数据。例如 ISP 程序可以擦写 AP 或 IAP 区域代码或数据,而 AP 程序只能擦写 IAP 区域数据。这些擦写限制的安全机制在程序指令上并没有区别,而是通过设计处理器硬件配置寄存器,设置不同安全级别程序空间的边界地址来实现。执行该指令时检查 PC 是否满足地址边界条件,这个检查与指令执行是流水操作,并不会延迟指令的执行。

此外,为了进一步保护代码,还可以设计程序代码区域的密码保护机制,包括单一密码机制或者独立的擦写和读取密码机制。仅当密码校验正确时,才可以执行程序,或者对程序代码区域进行相应操作,以充分保护程序代码安全。

3　多协议 ISP 结构设计与实现

考虑到设备开发阶段和生产维护阶段程序升级方式的不同需求,设计在线升级程序时实现多接口多协

议 ISP 方案。除常规的串口升级外,多协议 ISP 程序还可以支持通过 FLASH 更新程序代码进行升级,设备运行中通过 3G 或 4G 运营商网络进行远程升级。这种升级方式先缓存更新版本的程序代码到片内或者片外 FLASH 中,校验数据完整性后,软件复位执行 ISP 程序进行升级。此外由于设备为物联网终端设备,带有非接触智能卡读卡功能,因此特别设计了刷卡升级,通过多张大容量非接触智能卡直接进行升级,方便快捷。

多协议 ISP 框架如图 2 所示。

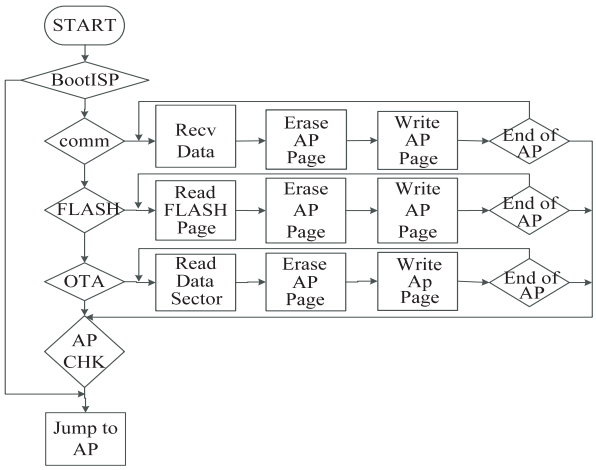

图 2 多协议 ISP 框架

多协议 ISP 程序设计时需要根据不同接口方式,以及处理器特性设计各预定接口相应的通讯协议。设计时选择处理器具有 1 k 字节的 RAM,FLASH 页面大小为 512 字节,因此设计串行通讯协议按数据包传输,除最后一个数据包外,有效数据载荷为 512 字节,另加 8 位校验码。通讯报文如表 1 所示。

表 1 串口通讯协议

包头	序号	类型	包长	数据	校验	包尾
8 位	8 位	4 位	12 位	512 位	8 位	8 位

类型与包长共两字节,第一字节高 4 位为包类型字段,第一字节低 4 位与第二字节为包长字段,这样数据包长为 12 位。4 位包类型字段决定有效载荷是数据还是命令和参数,命令包括擦除处理器内部存储页面,写页面,读页面,擦除外部 FLASH 页面,写 FLASH 页面,读出 FLASH 页面等命令。

FLASH 升级模式相对简单,擦除处理器程序代码空间页面后,直接读取外部 FLASH 页面新版本代码数据,写入内部 AP 程序空间页面即可。

非接触智能卡升级除需要考虑智能卡内部数据组织方式和大小外,与其他方式相似。而远程升级则需要 AP 应用程序配合,由 AP 在运行中通过运营商网络接收升级数据包写入 FLASH 缓存,校验数据完整性后采用 FLASH 升级方式升级。

4 仿真及验证

4.1 ISP 代码保护仿真

为了验证 ISP 代码安全机制设计,基于 DW8051 的 Verilog 模型,扩展两个 16 位地址配置寄存器和一个复位选项控制寄存器[8]。两个地址寄存器用于将地址空间划分为 AP 区域、IAP 区域和 ISP 区域,如图 1 所示。复位控制寄存器用于指定复位后选择执行 ISP 还是 AP。此外通过外部 SFR 总线扩展 ISP 部件,提供 ISP 寄存器以及硬件代码保护功能[9]。

ISP 部件寄存器包括模式寄存器 ISP_MODE,用于选择读字节、写字节或擦除页面。16 位 ISP 地址寄存器,由于 DW8051 寄存器为 8 位,因此分为高低字节地址寄存器 ISP_ADDRH 和 ISP_ADDRL,此外数据寄存器 ISP_DATA,用于存放写入或读出的字节数据,命令寄存器 ISP_SCMD 用于按序列依次写入命令字,以触发 ISP 操作[10]。

ISP 安全设计包括两组 16 位地址比较器和一个命令序列比较器和定时器,比较 PC 与边界地址,比较 ISP 地址与边界地址,仅当满足权限才可以执行 ISP 操作,如表 2 所示。

表 2 ISP 程序操作权限

		ISP_ADDR		
		AP	IAP	ISP
PC	AP	×	√	×
	IAP	×	×	×
	ISP	√	√	×

此外必须在定时器设定的计数器时间内,依次写入 4 字节命令字到命令寄存器才能触发 ISP 操作。这样既确保代码不被自身修改,也不易误触发 ISP 操作造成程序丢失。

4.2 多协议 ISP 验证

为了验证多协议 ISP 设计的可行性,选择具有 ISP 硬件支持的 51 指令集单片机 MPC82G516 作为验证平台,外围设计基于 FM1702SL 非接触智能卡控制器的读卡电路,板上集成 AT45DB161 数据存储器,为 SPI 接口的大容量 FLASH,串口一电平转换电路 MAX3232,通过串口二集成 SIM900A GPRA 模块,以支持远程升级功能[11]。

硬件平台框图如图 3 所示。

ISP 程序流程如图 2 所示,整个 ISP 程序代码空间为 4 kB,程序设计支持串口升级、板载 FLASH 升级、非接触智能卡升级、GPRS 网络远程升级。可以通过液晶屏直观显示升级进度和状态。

加电或复位启动后如果 MPC82G516 硬件设置为 ISP 启动,执行 ISP 程序,轮询串口一,如果串口一在设

定时间内没有收到 ISP 握手信号,就进入下一步 FLASH 升级流程。如果有握手信号,则进入串口 ISP 流程,根据接收的指令,缓存数据包,擦除 AP 页面,写入 AP 页面,完成后检测 AP 的程序完整性,检测通过则跳转到 AP 执行。串口 ISP 流程也可以根据指令将程序数据包存入 FLASH,作为备份程序[12]。

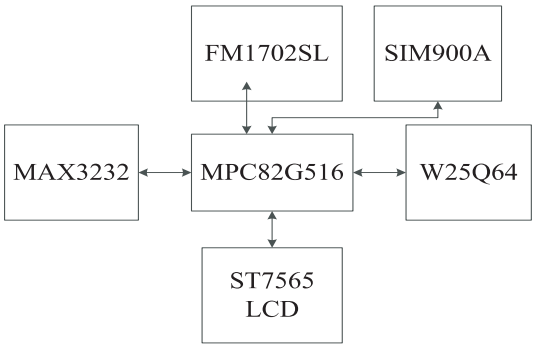

图 3　单片机验证平台框架

FLASH 升级流程,检查 FLASH 的备份程序空间,如果存在更新版本,先校验整个备份程序的数据完整性,校验通过后依次按 512 字节页面读取备份程序片断,擦除 AP 页面,按字节依次写入整个 AP 页面,循环执行直到完成所有备份程序的读入升级[9]。完成升级后进行 AP 完整性检测,通过后跳转执行 AP,这里 FLASH 中的备份程序可以来自串口 ISP 流程的备份程序,或者 SIM900A 的远程升级流程[13]。

如果 FLASH 中没有更新版本,则 ISP 程序通过非接触智能卡模块检测有无升级卡存在,通过密码检测和数据标志检测判定升级卡,存在升级卡则依次读入扇区数据块数据,擦除 AP 页面,写入 AP 页面,完成升级后进行完整性检测,通过后跳转执行 AP。与其他升级方式有一个需要特别注意的地方,就是 FLASH 的按块擦除特性。由于非接触智能卡的数据容量并不与页面相匹配。AP 页面为 512 字节,而非接触智能卡为扇区、块结构,如图 4 所示。

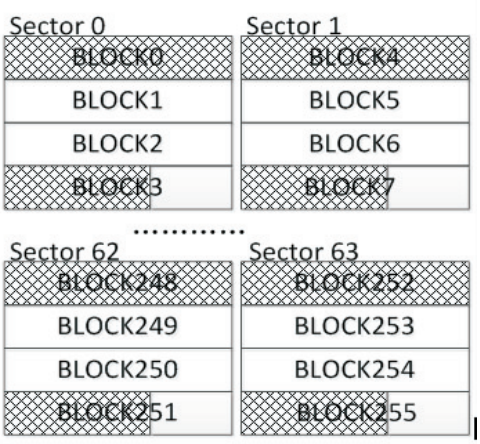

图 4　升级卡结构

采用 FM11RF32 非接触型智能卡作为升级卡,该卡共 64 扇区,每扇区包含 4 块,每块大小 16 字节。扇区间密码独立。其中扇区 0 的块 0 为只读块,每扇区的第三块为密码及控制块,包括 6 字节密码 A、4 字节控制码和 6 字节密码 B,当只使用一个密码时,6 字节密码 B 空间可以用于存放数据。因此单卡最多可以 3 440字节数据,其中占用一块数据用于卡识别校验码、数据起始地址、数据长度,因此单卡可以存放 3 424 字节程序。为了进一步提高代码容量,可以进行数据压缩[14]。

升级卡容量与 AP 页面不匹配,因此除第一张卡外,其他卡存在数据地址与 AP 页面不对齐问题,擦除时将丢失该页面之前写入的升级数据。因此需要根据当前升级卡的起始地址,计算所在的 AP 页面和偏移地址,读出该页面到缓存,再擦除页面,并从偏移地址开始继续缓存数据,满 512 字节后写入 AP 页面,如图 5 所示。

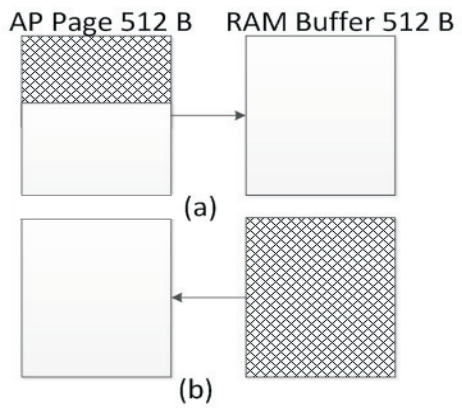

图 5　非接触智能卡升级页面处理

AP 程序也需要参与到 ISP 流程中,以便在系统运行时也可以进入 ISP 流程。一旦 AP 在运行中检测到类型为升级卡的非接触智能卡,则通过软件复位指令,跳转到执行 ISP 程序。

设备运行中,AP 程序接收到 SIM900A 发送的远程升级指令时,则进入远程升级流程,接收升级程序代码数据,写入 FLASH 缓存,校验通过后写入升级标志,同样通过软件复位,跳转到 ISP 执行 FLASH 升级[15]。

5　结束语

实际测试证明多功能 ISP 设计是完全可行的,该 ISP 程序不仅便于开发测试阶段对微处理器在线下载编程,也易于在产品应用阶段进行刷卡升级维护或者通过通信网络进行远程升级,降低了维护成本。而且由于协议设计充分考虑了升级过程存在的不确定因素,对升级数据进行完整性校验,并始终保存一份可用代码备份,确保产品的可用性。该设计已经成功应用

(下转第 28 页)

信方式,在 VMware 虚拟化平台下使用 VMCI 的通讯方式。在单节点多 GPU 内部通信方面,设计了两种方式(G_G 和 G_H)完成 GPU 间的数据传输,以矩阵复合运算为应用场景设计了实验,结果验证了预测结果的正确性。将来的工作是在集群环境下对 GPU 虚拟化的数据传输问题的分析。

参考文献:

[1] 崔泽永,赵会群. 基于 KVM 的虚拟化研究及应用[J]. 计算机技术与发展,2011,21(6):108-111.

[2] Overby E. Process virtualization theory and the impact of information technology[J]. Organization Science, 2008, 19(2): 277-291.

[3] Lambert D, Domingue J. Photorealistic semantic web service groundings: unifying RESTful and XML-RPC groundings using rules, with an application to Flickr[C]//Proc of the 4th international web rule symposium. [s. l.]: [s. n.], 2010.

[4] Vinoski S. CORBA: integrating diverse applications within distributed heterogeneous environments[J]. IEEE Communications Maganize, 1997, 35(2): 46-55.

[5] Henning M. A new approach to object-oriented middleware [J]. IEEE Internet Computing, 2004, 8(1): 66-75.

[6] 张 舒,褚艳利. GPU 高性能运算之 CUDA[M]. 北京:中国水利水电出版社,2009.

[7] Zhang X, McIntosh S, Rohatgi P, et al. Xensocket: a high-throughput interdomain transport for virtual machines[C]// Proc of international middleware conference. Newport Beach, CA: ACM, 2007: 184-203.

[8] Kim K, Kim C, Jung S, et al. Inter-domain socket communications supporting high performance and full binary compatibility on Xen[C]//Proc of international conference on virtual execution environments. Seattle: ACM, 2008: 11-20.

[9] Wang J, Wright K, Gopalan K. XenLoop: a transparent high performance inter-VM network loopback[C]//Proc of international symposium of high performance distributed computing. Boston: ACM, 2008: 109-118.

[10] 陈 浩,彭萃芬,孙建华,等. XenRPC:安全的虚拟机远程过程调用设计与实现[J]. 计算机研究与发展,2012,49(5):996-1004.

[11] 顾晓峰,王 健. 基于 Intel VT-x 的 XEN 全虚拟化实现[J]. 计算机技术与发展,2009,19(9):242-245.

[12] 孟江涛,卢显良,董贵山. Xen 的虚拟机网络优化研究[J]. 电子科技大学学报,2010,39(1):106-109.

[13] VMCIoverview[EB/OL]. 2014-09-08. http://pubs.vmware.com/vmci-sdk/.

[14] 包敬海,周小珠,樊东红. 基于 VMWare 构建虚拟网络实验室的研究[J]. 计算机技术与发展,2010,20(6):242-245.

[15] Direct 2.0[EB/OL]. 2014-09-16. http://www.valtra.com/news/6568.asp.

[16] 董小社,刘 超,王恩东,等. 面向 GPU 异构并行系统的多任务流编程模型[J]. 计算机学报,2014,37(7):1638-1646.

(上接第 23 页)

于多个产品,充分验证了该设计的可行性。

参考文献:

[1] 郑小军,胡道徐. 一种通用的嵌入式系统 ISP 方法[J]. 电子技术应用,2005,31(7):22-23.

[2] NXP. In-circuit and in-application programming of the 89C51Rx+/Rx2/66x microcontrollers[S/OL]. 2002. http://www.nxp.com/documents/application_note/AN461.pdf.

[3] 阮军杰,方 岑,李时昌. C8051F12x 单片机 128KB Flash 的 BootLoader ISP/IAP 的实现[J]. 工业控制计算机,2013(10):128-129.

[4] Atmel. In-system programming[S/OL]. 2008. http://www.atmel.com/Images/doc0943.pdf.

[5] 吴 军,华更新,刘鸿瑾. SoC 验证方法学研究与应用[J]. 空间控制技术与应用,2012,38(5):27-33.

[6] 姚 露,朱念好. 基于 DW8051 平台的 MPU 设计与验证[J]. 信息技术,2012(1):118-119.

[7] 夏宇闻. Verilog 数字系统设计教程[M]. 北京:北京航空航天大学出版社,2003.

[8] Baer Jean-Loup. Microprocessor architecture: from simple pipelines to chip multiprocessors[M]. Cambridge: Cambridge University Press, 2009.

[9] 尹 恒,严 华. 一种针对嵌入式远程升级安全的存储解决方案[J]. 计算机应用,2011,31(4):942-944.

[10] Harris D, Harris S. Digital design and computer architecture [M]. 2nd ed. Burlington: Morgan Kaufmann, 2010.

[11] 蒋美娟,郑 羽,陈瑞林,等. 基于 LPC1768 汽车故障远程诊断控制器的设计[J]. 计算机技术与发展,2013,23(8):238-241.

[12] 李国俊,董晶晶,周 瑾. 智能卡 COS 安全性测试研究[J]. 计算机技术与发展,2014,24(2):164-167.

[13] 褚东升,刘 滨,綦声波,等. ISP 技术在智能仪器远程升级中的应用[J]. 单片机与嵌入式系统应用,2002(4):46-48.

[14] 付 超,余本功. 嵌入式无线移动设备的开放式远程现场升级[J]. 计算机工程与应用,2007,43(1):3-5.

[15] 刘根贤,龚雪容,生拥宏,等. 基于高频 RFID 的微处理器 IAP 技术[J]. 电子技术应用,2013,39(4):29-31.

嵌入式微处理器多协议ISP设计及安全性分析

作者： 刘根贤， 汪东升， 王海霞， LIU Gen-xian， WANG Dong-sheng， WANG Hai-xia

作者单位： 刘根贤,LIU Gen-xian(清华大学 计算机科学与技术系，北京 100084；第二炮兵工程大学信息管理中心，陕西 西安 710025)， 汪东升,王海霞,WANG Dong-sheng,WANG Hai-xia(清华大学 信息科学技术国家实验室筹,北京,100084)

刊名： 计算机技术与发展 ISTIC

英文刊名： Computer Technology and Development

年，卷(期)： 2015(8)