用于加解密流程控制的协处理器①

王剑非[1], 马 德[2], 黄凯杰[3], 陈 亮[3], 黄 凯[3], 葛海通[4]

[1](公安部第一研究所, 北京 100048)
[2](杭州电子科技大学 微电子 CAD 所, 杭州 310018)
[3](浙江大学 超大规模集成电路设计研究所, 杭州 310027)
[4](杭州中天微系统有限公司, 杭州 310012)

摘 要: 本文设计与实现了一种专用于加解密流程控制的协处理器. 协处理器根据特定的应用需求, 自定义了一种精简的 8 位指令集, 同时采用与 SoC 系统一致的 32 位数据位宽设计. 协处理器采用三级流水线设计, 数据旁路的设计解决了流水线中的数据冒险. 通过与加解密算法 IP 联合测试仿真, 验证了协处理器能够灵活地完成加解密流程控制工作. 通过 SM1 加密实验, 证明了协处理器能够提供较主处理器更好的性能, 同时释放大量的主处理器资源, 显著提高了 SoC 的性能. 最后 DC 综合结果显示, 该协处理器只占用了很小面积.

关键词: 信息安全 SoC; 流程控制; 指令集; 流水线; RSIC 处理器

Coprocessor Applied for Encryption and Decryption Flow Control

WANG Jian-Fei[1], MA De[2], HUANG Kai-Jie[3], CHEN Liang[3], HUANG Kai[3], GE Hai-Tong[4]

[1](The First Research Institute of the Ministry of Public Security, Beijing 10048)
[2](Institute of Microelectronics CAD, Hangzhou Dianzi University, Hangzhou 310018)
[3](Institute of VLSI Design, Zhejiang University, Hangzhou 310027)
[4](C-Sky Microsystems CO. LTD., Hangzhou 310012)

Abstract: A coprocessor specially applied for encryption and decryption flow control was designed. According to the specific application requirement, a kind of reduced 8-bit width instruction set was designed with 32-bit data width design compliant to SoC system. The coprocessor adopted 3 stage pipeline design with data bypass design to prevent data hazards. Joint test with secure IPs showed that the coprocessor could control the encryption and decryption flow flexibly. Experiments on SM1 encryption proved that the coprocessor could provide better performance than the main processor and release main processor resources. The Design Compiler synthesis result showed the coprocessor occupied only a small area.

Key words: SoC for information security; flow control; instuction set; pipeline; RSIC CPU

1 引言

随着信息技术的快速发展, 人们对于信息化的依赖程度越来越高, 随之而来的信息安全问题成为关注的焦点. 近年来, 国防工业、政府机构和民用电子都对高性能的信息安全 SoC 芯片有着强烈的需求. 这类信息安全 SoC 芯片[1,2], 除了集成 RSA、AES、DES/3DES 这类国际上通用的加解密算法 IP 外, 还会集成由国家密码管理局制定的国家商用密码算法 IP, 例如 SM1、SM2、SM3. 在大批量数据处理中, 这类加解密算法 IP 需要频繁地搬运明文密文、配置相关寄存器、判断加解密完成标志位. 在传统的信息安全 SoC 芯片中, 由于 DMA(Direct Memory Access, 直接内存存取)无法完成流程控制类工作, 所以这类工作往往都交由主处理器来完成. 文章[1]描述的平台架构中, 将 AES、DES 等加解密算法 IP 直接挂在总线上, 在大批量的数据处理中, 这类加解密算法 IP 流程控制操作会占用大量的主

① 收稿时间:2013-04-23;收到修改稿时间:2013-07-01

处理器资源, 进而影响 SoC 芯片的整体性能. 文献[2]虽然引入了加解密子系统的概念, 使用一个通用 CPU CK803 来专门控制加解密算法 IP 的工作流程, 但是对于流程控制这类简单操作, 通用 CPU 的性能过剩, 而且面积与功耗也过大.

基于加解密过程中简单却又频繁的操作, 本文设计与实现了一种应用于加解密流程控制的协处理器. 该协处理器使用 8 位精简指令集, 32 位数据位宽的设计, 硬件上采用 3 级流水线, 使用独立的指令总线与数据总线, 实现了数据存取、程序控制和逻辑运算三类指令, 能够灵活地完成加解密算法 IP 流程控制的所有工作.

2 整体设计与实现

协处理器主要应用于加解密算法 IP 流程控制, 通过 AMBA[3]总线与加解密算法 IP 互连, 可以组成一个加解密子系统. 其典型应用如下图 1 所示. 协处理器拥有一个 AHB Slave 接口, 主处理器通过该 Slave 接口可以访问协处理器内部的所有寄存器, 以实现对于协处理器的控制. 表 1 列出了协处理器内部的所有寄存器. 除了 AHB Salve 接口, 协处理器还有两个独立的 AHB master 接口, 分别为指令总线接口和数据总线接口. 指令总线接口通过 AHB 总线连接到片上存储, 负责协处理器指令的读取. 数据总线接口通过 AHB 总线, 除了与各类加解密算法 IP 互连外, 还与片上存储相连, 负责数据的存取.

表 1 协处理器内部寄存器

寄存器名	寄存器描述
Control	控制寄存器, 可控制协处理器的启动与停止
GPRA0~3	A 组通用寄存器
GPRB0~3	B 组通用寄存器
AR0~3	地址寄存器
DR	数据寄存器
PC	程序计数器
PSR	程序状态寄存器
INS_ADDR	指令地址寄存器

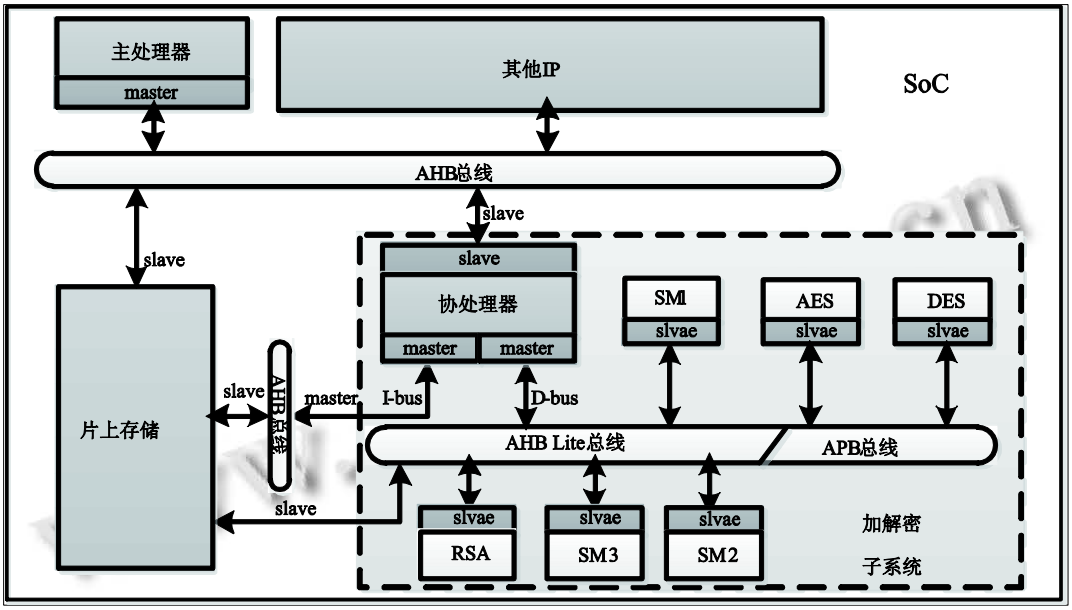

图 1 协处理器的典型应用

2.1 指令集设计

指令集表征着处理器的基本功能和使用属性, 是处理器系统设计中的核心向题[4]. 处理器的复杂度会随着指令集的不同而发生变化. 协处理器只负责加解密算法 IP 的流程控制工作, 其他复杂的运算操作都可以由主处理器来负责完成, 因此考虑到面积的影响, 同时为了简化设计, 协处理器采用 8 位的 RSIC 指令集.

指令集的定义与应用的对象密切相关, 因此在定义指令集之前, 需要先了解加解密算法 IP 工作的流程. 加解密算法 IP 的工作流程基本可以分为以下三类: 搬运明文与密文、配置相关寄存器、判断加解密完成标志位. 针对以上三种操作, 我们精心设计了以下三类指令:

运算指令(ADDC、SUBC、AND、OR、CMPE): 这类指令使用两个通用寄存器, 对它们进行相应运算, 然后将结果存入其中一个寄存器. ADDC 和 SUBC 是带进位的加减法指令, 其结果会影响条件位. CMPE 是比较指令, 相等则置条件位. AND 和 OR 是逻辑操作指令, 其结果不影响条件位.

分支和跳转指令(BR、BT): 这类指令使用一个 GPRB 寄存器, GPRB 寄存器中存放跳转的目的地址. BR 是无条件跳转指令, BT 是条件跳转指令, 根据条件位来判断是否跳转到目的地址.

载入和存储指令(LD、ST): 这些指令使用一个 AR 寄存器和一个 DR 寄存器, AR 寄存器中存放地址基址, 在指令码中还有一个立即数偏移量, 它们共同构成了指令的操作码. 指令的操作结果全部放入统一的 DR 寄存器.

除了以上 3 类, 还设计了 NOP 和 MOV 指令, NOP 指令是空操作, MOV 指令用于实现寄存器之间数据转移. 表 2 给出了协处理器中实现的所有指令和对应的指令码.

表 2　指令码表

指令码								指令
0	0	0	0	0	0	0	0	nop
0	0	0	0	0	0	0	1	reserved
0	0	0	0	0	0	1	x	reserved
0	0	0	0	0	1	gb_sel		cmpe dr, gprb
0	0	0	0	1	0	gb_sel		br gprb
0	0	0	0	1	1	gb_sel		bt gprb
0	0	0	1	x	x	x	x	reserved
0	0	1	x	x	x	x	x	reserved
0	1	0	0	ga_sel		gb_sel		addc gpra,gprb
0	1	0	1					subc gpra, gprb
0	1	1	0					and gpra, gprb
0	1	1	1					or gpra, gprb
1	0	0	ar_sel		offset			st dr, ar
1	0	1						ld dr, ar
1	1	0	0	ga_sel		gb_sel		mov gpra, gprb
1	1	0	1	ar_sel		ga_sel		mov ar, gpra
1	1	1	0	0	0	ga_sel		mov gpra, dr
1	1	1	0	0	1	gb_sel		mov gprb, dr
1	1	1	0	1	0	ga_sel		mov dr, gpra
1	1	1	0	1	1	gb_sel		mov dr, gprb
1	1	1	1	gb_sel		ar_sel		mov gprb, ar

2.2 流水线设计

本文设计的协处理器采用了 3 级流水线[4], 3 级流水线分别为:

IF: 指令取值. 根据 PC(Program Counter, 程序计数器)指示的地址通过 I-BUS 总线从存储器中读取指令, 同时 PC 加 1(因为每条指令是 1 个字节)以获取下一条指令.

EX: 指令执行. 在指令执行阶段, 会完成指令译码、ALU 运算和存储器访问三项工作. 对于算数逻辑运算, 其结果会在 EX 阶段直接写回寄存器堆.

WB: 写回周期. 由于数据存储器访问存在延时, 无法在单个 Cycle 读回数据, 所以对于 LD(load, 从存储器载入)指令需要在 WB 阶段写回寄存器堆.

图 2 给出了协处理器内部流水线完整的微架构设计. 协处理器内部主要由 PC 生成器、指令总线控制器、指令译码器、寄存器堆、ALU(算术逻辑单元)和数据总线控制器 6 部分组成.

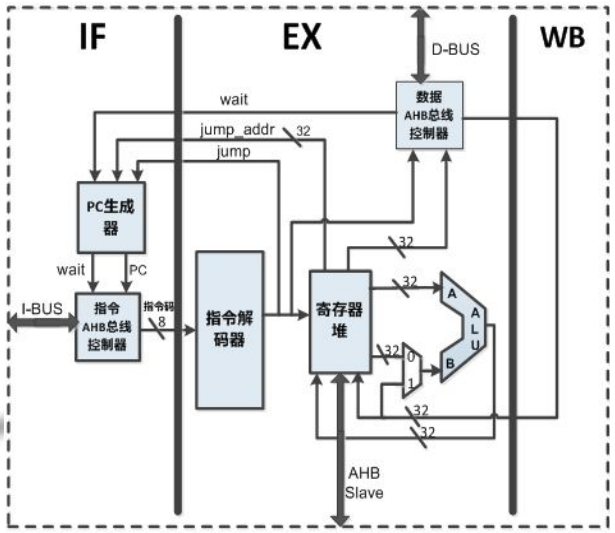

图 2　流水线架构

图 3 给出了简单的 3 级 RISC 流水线执行过程. 每一个时钟周期都有一条新的指令取进来并开始长达 3 个时钟周期的执行过程. 在同一个时钟周期, 最多会有 3 条指令在同时执行, 这样就有可能产生冒险问题. 例如在时钟周期 3, 如果指令 i+1 在 EX 阶段需要用到指令 i WB 阶段的结果, 那么由于指令 i WB 的结果在时钟周期结束时才写回到寄存器堆, 而指令 i+1 的 EX 需要在时钟周期开始就需要从寄存器堆取数, 这样就出现了数据冒险. 解决这个数据冒险的方法是增加数

据旁路, 即指令 i+1 EX 阶段需要的数据直接从 WB 阶段取, 而不是从寄存器堆取数. 图 2 ALU B 端的 32 位选择器即为此数据旁路.

指令	时钟					
	1	2	3	4	5	6
指令 i	IF	EX	WB			
指令 i+1		IF	EX	WB		
指令 i+2			IF	EX	WB	
指令 i+3				IF	EX	WB

图 3 简单流水线指令执行

在协处理器内部还有可能出现指令总线(I-BUS)或者数据总线(D-BUS)阻塞的情况,即由于总线或者存储被抢占导致指令或数据无法在下一个 Cycle 取回. 在这种情况下就需要停顿流水线, 指令 AHB 总线控制器检测到这种阻塞情况就会把指令码清零, 即用 NOP 指令代替. 图 4 给出了数据总线和指令总线被阻塞情况下流水线指令执行.

指令	时钟					
	1	2	3	4	5	6
指令 i	IF	EX	Block	WB		
指令 i+1		IF	nop	nop		
指令 i+1			IF	EX	WB	
指令 i+2				Block	nop	nop
指令 i+2					IF	EX

图 4 总线阻塞流水线指令执行

3 RTL仿真验证与性能评估

3.1 RTL 仿真与验证

在完成整体架构设计和 RTL(Register Transter Level, 寄存器传输级)级代码之后, 需要对协处理器的功能进行完整的验证. 根据该协处理器的典型应用, 我们集成了图 5 所示的 SoC 系统[5], 同时构建了一个测试平台. 在 SoC 系统中, 主处理器采用了国产 32 位嵌入式处理器 CK803[6](CK803 是由杭州中天微系统有限公司自主研发的拥有低成本、低功耗嵌入式 CPU, 性能为 1.5 DMIPS/MHz, 与 ARM Cortex M3[7](1.25 DMIPS/MHz)性能相当), 片上存储采用拥有多个 Bank, 支持多个端口同时访问不同 Bank 的 SRAM 系统. 图 5 给出了完整的验证环境.

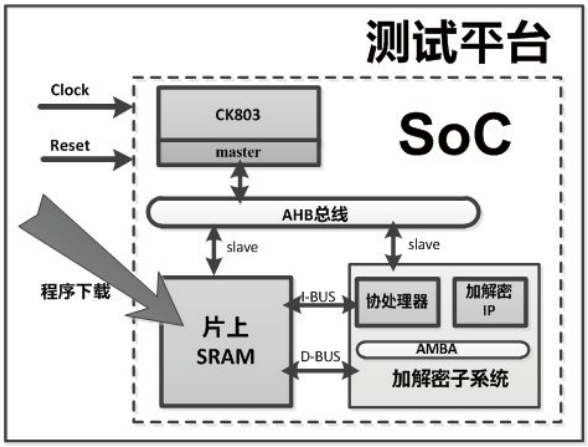

图 5 仿真环境

在上述的验证环境与流程下, 首先对协处理器单条指令的执行进行了仿真测试, 这是协处理器最基本的功能. 接下来对协处理器各种指令的组合进行仿真测试, 用于验证流水线中可能存在的冒险问题. 最后, 通过与各加解密算法 IP 联合测试仿真, 完成验证工作. 表 3 给出了协处理器验证过程中使用的部分测试激励.

表 3 试向量列表

	测试向量名
单条指令激励	ins_ld.s
	ins_bt.s
	ins_mov.s
	…
多条指令组合激励	ins_ld_addc.s
	ins_ld_mov.s
	ins_addc_bt.s
	…
算法 IP 联调激励	coprocessor_sm1.s
	coprocessor_aes.s
	coprocessor_rsa.s
	…

3.2 能评估实验

在传统的信息安全 SoC 芯片中, 加解密算法 IP 会直接挂在 AMBA 总线上, 由主处理器来实现加解密流程控制. 本实验就通过对比传统信息安全 SoC 流程控制实现方法, 从加解密速率和主处理器需要参与的操作两方面来说明使用协处理器的加解密子系统拥有更高的加解密速率, 同时能够释放大量的主处理器资源, 能够显著提升 SoC 的整体性能.

主处理器对于加解密算法 IP 的流程控制可以分为

轮询法和中断法. 轮询法由主处理器循环查询加解密算法 IP 的加解密完成标志位, 该方法会一直占用主处理器的资源, 但能够迅速的完成加解密流程控制. 中断法即在加解密算法 IP 完成加解密操作后产生中断信号, 主处理器在中断服务函数里完成对加解密算法 IP 的流程控制, 该方法能够释放一定的主处理器资源, 但在进出中断服务函数时需要进行现场保护, 所以会降低加解密的速率.

相较于传统的 SoC 加解密算法 IP 流程控制方法, 使用协处理器的加解密子系统, 主处理器只需完成少量的协处理器配置工作, 即可以将大量的加解密流程控制操作交给协处理器来完成.

本实验集成了图 1 所示的典型应用 SoC 系统, 使用图 5 所示的仿真环境, 分别开发了以上三种 SM1 加解密流程控制程序(图 1 所示 SoC 系统中, 协处理器内部的数据通路设计可以保证主处理器可以经过协处理器 Slave 接口和 D-BUS 接口直接访问加解密子系统内的加解密算法 IP, 所以能够实现传统 SoC 的控制方法). 下表 4 和表 5 分别给出了以上三种流程控制方法控制 SM1 加密 64Byte 和 2KByte 数据主处理器和协处理器的工作时间(Cycle 数)的统计结果.

表 4 SM1 加密 64Byte 性能统计

方法名		主处理器	协处理器	总时间
主处理器	轮询	458	0	458
	中断	329	0	596
协处理器		130	267	397

表 5 SM1 加密 2KByte 性能统计

方法名		主处理器	协处理器	总时间
主处理器	轮询	14163	0	14163
	中断	10341	0	17695
协处理器		130	7571	7701

比较协处理器和主处理器的工作时间可以发现, 协处理器控制要比主处理器控制来得更快. 分析发现, 由于主处理器连在标准 AHB 总线上, 每次执行 Load 和 Store 指令都需要申请总线并等待总线仲裁器的授权后才能发起操作, 而协处理器连在 AHB Lite 总线上, 每次执行 Load 和 Store 指令无需申请总线, 因此在数据搬运时更为迅速.

分析上面两个表格数据可以发现, 加密 64Byte 和 2K Byte 数据, 使用协处理器要比使用主处理器轮询法分别快了 13.31%和 45.62%. 而且使用主处理器轮询法, 主处理器 100%的资源都用在加解密流程控制上; 而协处理器控制中, 主处理器只需配置好协处理器, 即可将所有的操作交给协处理器, 主处理器工作分别只占了全部工作的 32.75%和 1.7%. 使用主处理器中断法, 虽然一定程度上释放了主处理器的资源(主处理器的工作分别下降了 28.16%和 26.99%), 但是由于在进出中断服务函数时需要进行保护现场, 总共加解密时间分别会有 30.13%和 24.94%的增加.

综合以上分析, 使用协处理器可以有效地加快加解密流程控制速度, 并且释放大量的主处理器资源, 这种优点在大批量数据加解密控制中显得尤其突出.

3.3 逻辑综合结果

在逻辑综合过程中, EDA 工具使用业界最常用的 Synopsy(新思科技)的 Design Compiler 进行综合实现, 标准单元库则使用了 SMIC(中芯国际)0.18um 工艺的 Stand Cell. 综合结果显示, 其最大工作频率可以达到 200MHz. 表 6 给出了在 100MHz 时钟约束下, 加解密子系统的综合结果. 结果显示, 协处理器只占了整个加解密子系统 1.82%的面积.

表 6 综合结果

IP	面积
协处理器	0.081mm^2
AHB 总线	0.032mm^2
加解密算法 IP 面积总和	4.491mm^2

4 结束语

本文设计与实现了一种应用于加解密流程控制的协处理器, 结合这个设计论述了 RISC 处理器指令集的定义与流水线的基本设计方法. 通过与加解密算法 IP 联合仿真测试, 证明了协处理器能够完成加解密算法 IP 的流程控制工作. SM1 的加密实验结果, 说明使用协处理器可以有效地加快加解密流程控制速度, 并且释放了大量的主处理器资源, 能够显著提高整个 SoC 的性能. Design Compiler 的综合结果表明该协处理器具有很小的面积.

参考文献

1 曾晓洋,吴敏,韩军,吴永一,林一帆,陈俊,闵昊,章倩苓.信息安全芯片 SoC 平台及其应用.信息安全与通信保密,2005,7: 28–30.

2 陈亮.基于嵌入式 CPU 的数据加解密子系统的设计研究

(下转第 217 页)

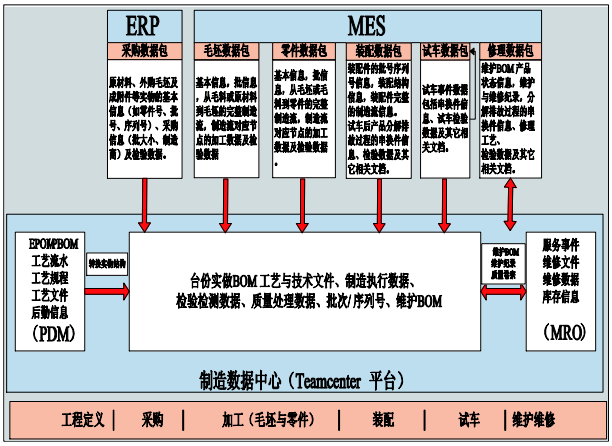

图 5 制造数据中心与其它系统集成的数据关联示例

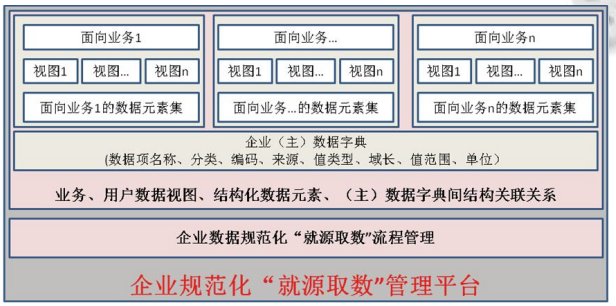

图 6 企业规范化"就源取数"管理平台示意图

4 结语

两化融合是信息化能力和企业能力的融合，是信息技术和业务全面融合的过程. 对复杂产品(如航空发动机)制造企业来说，各业务系统信息融合是企业信息化进程的重要阶段、难点和关键，其中数据集成是企业各业务系统信息融合的核心和基础.

企业需要基于系统工程方法和两化融合的顶层信息集成规划来引导数据集成实施工作. 企业顶层信息集成规划包括企业信息规划和集成实施规划. 不同企业在信息集成工作中面对的内容和挑战是不尽相同的. 本文基于某企业两化融合过程实践经验，阐述了企业信息集成的挑战内容、工作思路和关键实施内容等.

企业信息集成规划与实施的成功性取决于企业顶层关注度、文化变更接受力、计算机基础条件、业务流程优化、数据标准化、IT 技术能力和实施团队执行力等多种因素. 企业信息集成规划与实施还有许多挑战和探索，在此愿与同仁交流，共同推进我国复杂产品制造业的两化深度融合.

参考文献

1 王学颖.企业信息资源规划:ILEA 的研究与设计[博士学位论文].武汉:武汉大学,2010.
2 毕新华,于宝君,齐晓云.中国企业信息系统宏观成长过程及阶段分析.情报科学,2008,26(2):161–166.
3 胡海清.多元化集团型企业的信息化实践研究[博士学位论文].山东:山东大学,2009.
4 张晓卿.基于 CMM 的组织信息化能力成熟度模型研究[博士学位论文].哈尔滨:哈尔滨工业大学,2010.
5 高复先.系统工程方法在信息化建设中的指导作用.战略与政策中国信息界,2010(12):14–17.
6 宋俊典,李名敏,金涛,杨根兴.基于 TOGAF 的轨道交通企业信息化架构规划研究.计算机应用与软件,2010,27(5):165–168.
7 赵国志.装备制造企业的信息化整体规划.中国制造业信息化,2008(18):40–42.

(上接第 208 页)

[硕士学位论文].杭州:浙江大学,2013.2.
3 ARM Inc. AMBA Specification Rev2.0. http://www.arm.com.
4 Hennessy JL, Patterson DA. Computer architecture:A quantitative approach. 4th ed. San Franciscok:Morgan Kaufmann. 2007. 66–78.
5 潘赟.CK-CPU 嵌入式系统开发教程.北京:科学出版社,2011.9:23–52.
6 杭州中天微系统有限公司.C-SKY®嵌入式 CPU 之 CK803. www.c-sky.com.
7 ARM Inc.ARM Cortex M3 处理器.www.arm.com.