

基于控制流检测的抗错误注入攻击方法

黄 威, 刘 恺, 廖 望, 戴 葵

(华中科技大学 光学与电子信息学院, 湖北 武汉 430074)

摘 要: 提出了基于控制流检测的抗错误注入攻击, 该方法对传统控制流检测技术进行了改进, 设计了基于基本块签名和基本块校验的控制流检测算法, 解决了传统控制流检测算法的盲区问题, 将该方法应用于安全芯片设计, 即使在错误注入成功的情况下, 也能检测出来。

关键词: 控制流检测; 错误注入; 安全芯片; 基本块; 签名; 校验

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2014)09-0058-05

Against Fault Injection Attacks a Method Based on Control Flow Check

HUANG Wei, LIU Kai, LIAO Wang, DAI kui

(School of Optical and Electronic Information, Huazhong University of Science & Technology, Wuhan 430074, China)

Abstract: In this paper, a method against fault injection attacks is proposed. This method has improved traditional control flow check technology through designing control flow check algorithm based on basic block signature and basic block verification, and has solved the blind spot problem of the traditional control flow check technology. So it can be used in the field of security chip design to resist fault injection attacks.

Key words: control flow check; fault injection; security chip; basic block; signature; verification

1 引言

在信息化时代, 信息的安全问题越来越重要, 作为信息安全的核心部件, 以提供加密、认证、安全存储等安全服务为主要功能的安全芯片^[1]应用越来越广. 与此同时, 针对安全芯片的攻击技术也层出不穷, 这使得研究安全芯片的防护技术成为迫切需要. 针对安全芯片的攻击手段中, 有一类叫做错误注入攻击的攻击手段, 它是一种半入侵式攻击, 其攻击原理是通过改变环境参数(芯片的电性能将随着不同的电压、温度、光、电离辐射以及周围的磁场的变化而变化), 来试图在芯片的程序逻辑中引入一些错误行为, 或为旁路攻击创造条件, 或导致信息泄露^[2]. 在目前的安全芯片设计技术中, 针对错误注入攻击的防护手段主要有两种: 一是在芯片外围增加一层

物理性防护, 防止错误注入; 二是增加芯片的检错硬件, 一旦检测到异常信息, 就错误报警, 芯片进入异常工作状态^[3]. 这两种防护手段都存在共同的缺点, 首先它们都需要大量传感器电路, 硬件成本巨大; 其次它们都仅仅是从源头上进行防御, 一旦错误注入成功, 就无能为力.

控制流检测技术从 Yau 首次提出至今已经发展了 20 多年^[4], 被广泛应用于微处理器的可靠性设计领域, 是防止由单粒子翻转事件而造成程序错误运行的有效手段之一. 传统的控制流检测技术检测粒度粗, 无法应用于安全芯片设计领域. 本文对传统的控制流检测技术提出了改进, 设计了一种基于基本块签名和基本块校验的控制流检测算法, 解决了传统控制流检测算法的盲区问题, 可以应用于安全芯片设计领域, 能有效的抵抗错误注入攻击.

收稿日期: 2013-11-28; **修回日期:** 2014-01-22

基金项目: 湖北省自然科学基金(ZRZ0051); 武汉市科技攻关重点项目(201150699190)

2 控制流检测算法

程序逻辑中的错误行为可以分为三类:非法的分支跳转、错误的分支跳转和非控制流指令错误.传统的控制流检测技术只能检测出非法的分支跳转,而且新增控制流检测指令自身也可能出错.本文对传统控制流检测算法进行了改进,采用基本块签名方法来检测非法的分支跳转,采用基本块校验方法来检测错误的分支跳转、非控制流指令错误和新增检测指令的自校验,解决了传统控制流检测算法的盲区问题.

2.1 基本块划分

基本块是控制流检测技术实施的基本单位.程序由指令组成,指令顺序执行,直到遇到一些会改变程序执行路径的指令,如分支指令.顺序执行的指令可以构成一个基本块,它是一段具有单入口和单出口的连续指令,即除了第一条指令以外,任何指令都不能成为分支指令的跳转目标,同时除了最后一条指令以外,任何指令都不能是分支指令.

在基本块的基础上就可以得到程序的控制流图.程序的控制流图是程序执行经过各基本块的所有可能的路径,可以用各基本块以及连接基本块之间的边所表示,即 $P = \{V, E\}$, $V = \{v_i \mid v_i \text{ 为基本块}, 1 \leq i \leq n, n \text{ 为基本块个数}\}$, $E = \{(v_i, v_j) \mid \text{表示程序的控制流从 } v_i \text{ (前驱基本块) 跳转到 } v_j \text{ (后继基本块)}\}$.如图 1(a)中的源程序,其程序控制流图如图 1(b)所示.

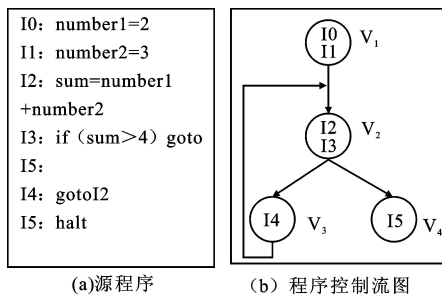


图1 程序控制流图

结合程序控制流图对程序逻辑中可能出现的三类错误描述如下:

(1)非法的分支跳转.基本块之间没有直接路径而出现的跳转,如从基本块 V_1 到基本块 V_3 的直接跳转.

(2)错误的分支跳转.基本块之间虽然有直接路径,但是在程序实际执行过程中,并不会发生的跳转.如基本块 V_2 到基本块 V_3 的跳转.

(3)非控制流指令错误.基本块内部非控制流指令出现错误.它可能并不会影响基本块之间的跳转,但是仍然可能会导致错误的执行结果.

2.2 基本块签名

基本块签名基本思想是通过程序运行时产生的签名同存储的签名信息进行比较,从而判断程序是否正确执行^[5].CFCSS 是其中一种广泛使用的基于签名的控制流检测算法,可以对 95.8% 以上的非法分支控制流检测错误进行有效检测^[6].本文采用基于 CFCSS 算法的基本块签名算法,并结合软硬件结合的实施机制对算法进行了优化,以降低其代码量和延迟开销.

基本块签名算法的原理是:在程序编译的时候由编译器按照一定的算法生成并赋予每个基本块静态签名;在程序运行的时候按照一定的算法计算每一个基本块的动态签名,并与该基本块的静态签名比较,判断是否发生了控制流错误.考虑存在多扇入基本块(存在多个前驱基本块)的情况,签名可以分成以下几类:

静态签名 S :在程序编译的时候由编译器生成,签名值是唯一的,用来唯一标识一个基本块.

调整签名 D :在程序编译的时候通过计算得到,其计算规则如式(1)所描述.

签名差 d :在程序编译的时候通过计算得到,其计算规则如式(2)所描述.

运行时签名 G :在程序运行的时候通过计算得到,并与静态签名 S 比较,如果不相等,则说明出现非法的分支跳转.其计算规则如式(3)所描述.

$$D_n = \begin{cases} 0, V_n \text{ 不与任何基本块共享后续基本块} \\ S_n \oplus S_m, V_n \text{ 与 } V_m (D_m = 0) \text{ 等共享后续基本块} \end{cases} \quad (1)$$

$$d_d = \begin{cases} S_s \oplus S_d \oplus D_s, V_d \text{ 只有 } V_s \text{ 一个前驱基本块} \\ S_m \oplus S_d, V_d \text{ 有 } V_m (D_m = 0) \text{ 等多个前驱基本块} \end{cases} \quad (2)$$

$$G_d = S_s \oplus d_d \oplus D_s \quad (3)$$

算法的证明过程如下:

(1) V_d 不是多扇入节点

V_d 不是多扇入基本块的时候,由式(1)有 $D_s = 0$,由式(2)有 $d_d = S_s \oplus S_d \oplus D_s$,通过式(3)计算 G_d 有

$$G_d = S_s \oplus S_d \oplus D_s = S_s \oplus (S_s \oplus S_d \oplus D_s) \oplus D_s = S_d \quad (4)$$

(2) V_d 是多扇入节点

V_d 是多扇入基本块的时候,由式(1)有 $D_s =$

$S_s \oplus S_m$, 由式(2)有 $d_d = S_m \oplus S_d$, 通过式(3) 计算 G_d 有

$$G_d = S_s \oplus d_d \oplus D_s = S_s \oplus (S_m \oplus S_d) \oplus (S_s \oplus D_m) = S_d \quad (5)$$

2.3 基本块校验

在通信系统中, 由于信道上各种复杂因素的影响, 所传输的信号往往会受到干扰, 严重时甚至会造成误码, 发生错误以致阻断通信. 为了提高通信的可靠性和减少误码率, 可以通过数据检验来进行差错控制, 使之能发现, 甚至能定位出错位置和实现纠错功能^[7].

基本块校验方法是将数据校验思想应用于控制流检测, 是对基本块签名算法的一种补充. 该方法以基本块为单位, 对处理器执行的指令流进行动态跟踪校验, 不仅能有效检测到错误的分支跳转和非控制流指令错误, 还能对基本块签名指令进行自校验, 解决了基本块签名算法的盲区问题.

数据校验包括检错和纠错, 这里只需要用到检错功能. 实现检错功能的数据校验方法有很多, 传统的有: 奇偶校验、校验和校验、重复码校验、恒比码校验、循环冗余码校验等. 本文结合传统数据校验方法和具体控制流检测应用, 设计了基本块校验算法, 算法设计遵循以下原则

(1) 稳定性. 如果基本块指令序列相同, 则计算出来的校验值也相同.

(2) 扩散性. 无论基本块指令序列具有何种分布特性, 校验值都具有很好的扩散性.

(3) 非线性. 基本块指令序列与校验值非线性相关.

(4) 符合硬件特性. 算法的硬件电路实现开销小、速度快.

(5) 不可逆性. 由基本块指令序列很容易计算出校验值, 反之则非常困难.

我们对于基本块校验算法进行了特性测试: 采用 MATLAB 平台, 利用工具自带的函数生成各种不同的分布样本, 对其序列计算校验值, 并对结果进行统计, 分析其扩散性及其与数据源的相关性.

校验值的扩散性通过频率分布直方图分析, 横坐标为校验值的区间, 纵坐标体现了落在该区间校验值的个数. 如果校验值分布得越均匀, 那么算法出现碰撞的概率也就越小. 具体测试结果如图 2 所示.

校验值与数据源的相关性通过 R 和 P 两个参数分析. R 是相关性系数, 其值在区间 $[-1, 1]$ 上, 越接近 0, 相关性越小. P 为假设检验中进行检验决策的一

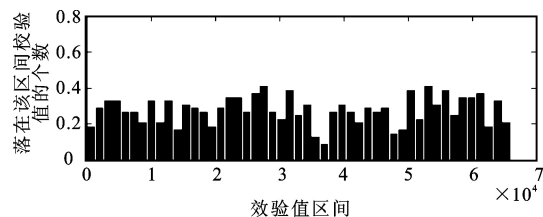


图 2 扩散性测试

个依据, P 值即概率, 它反映某一事件发生的可能性大小, 一般以 $P < 0.05$ 为显著, $P < 0.01$ 为非常显著. 本文中 P 的含义是假设扩散前后的数据不相关, 如果 $P < 0.05$, 则数据显著相关, 如果 $P < 0.01$, 则数据非常显著相关. 具体测试结果如表 1 所示.

表 1 相关性测试

分布类型	相关性系数 R	显著性水平 P
正态分布	0.000 9	0.931 4
指数分布	-0.003 6	0.716 3
泊松分布	0.003 5	0.728 6
离散分布	0.003 5	0.728 6

3 实现方法

应用于安全芯片设计的控制流检测方法采用软硬件结合如图 3 的实现方法: 通过软件处理, 对源程序代码划分基本块, 插入基本块签名与基本块校验指令; 在程序的执行过程中, 通过相应的硬件检测模块对程序的控制流进行检测. 具体过程如图 3 所示, 具体技术方案如下:

(1) 扩展硬件指令集: 对硬件指令集进行扩展, 加入控制流检测指令, 包括基本块签名指令和基本块校验指令;

(2) 基本块划分与签名: 通过签名程序对标准汇编文件进行基本块划分, 并生成基本块签名, 在基本块头部插入基本块签名指令;

(3) 基本块校验值生成: 在程序的汇编过程中, 对基本块指令序列采用基本块校验算法, 计算基本块校验值, 在基本块尾部插入基本块校验指令;

(4) 硬件检测: 在处理器内添加相应硬件检测模块, 实现基本块签名和基本块校验功能. 基本块签名模块检测非法的分支跳转; 基本块校验模块检测错误的分支跳转和非控制流指令错误, 以及对控制流检测指令进行自校验.

3.1 软件实现过程

软件实现是指对源程序(汇编程序)进行处理, 划分基本块, 插入基本块签名和基本块校验指令, 具体实现过程如下:

(1) 高级语言源文件编译生成标准汇编文件.

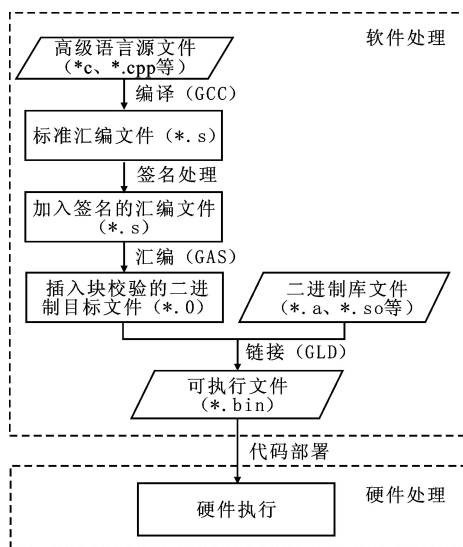


图3 软硬件结合的实现过程

高级语言源文件可以是C、C++或其他高级语言编写的源程序,只要是安全芯片所用的编译器所支持的;对于程序所调用的静态或动态二进制库文件,可以跳过签名和汇编处理部分,直接在链接部分调用。

(2)签名程序对标准汇编文件进行基本块划分,分析基本块之间的跳转关系。

签名程序对标准汇编文件进行词法分析,进行基本块划分,其划分依据为:函数标号、内部标号和跳转语句标志着一个块的开始;跳转语句和函数返回语句标志着一个块的结束;如果内部标号前一条语句不是跳转语句或返回语句,则加入一条跳转到该标号的绝对跳转语句,使得每一个基本块都必须由一条跳转语句跳入。在完成上面的工作后,遍历所有基本块,分析基本块之间的跳转关系,将具有跳转关系的基本块组织为链表,每个链表从头结点到尾结点的所有可能单向路径构成该程序的所有可能控制流路径。

(3)签名程序生成基本块的签名值,插入基本块签名指令。

通过(2)生成的链表,可以用来生成基本块的签名值,包括静态签名S、签名差d和调整签名差D。对于基本块签名值的生成方法,参考CFCSS算法,生成基本块的签名值之后,在基本块开头和结尾插入标志着块开头和结尾的伪指令,在基本块开头插入基本块签名指令。

(4)编译器将汇编文件翻译为目标文件,在翻译的过程中生成基本块校验值和插入基本块校验指令。

首先需要扩展汇编器,使其支持控制流检测扩展指令。这里使用的是GNU Binutils编译工具链中的GAS汇编器,它遵守GPL协议,可以自由地“使用、复制、修改和发布”,与其他GNU编译工具配合使用,可以完成整套开发过程。扩展GAS汇编器相对简单,主要是在指令列表中加入相应扩展指令的定义,然后编写指令处理函数,完成对指令二进制机器码的各个域的填充。其次,在汇编器将汇编文件翻译成二进制机器码的时候,通过对基本块内指令的二进制指令序列进行基本块校验算法处理,得到基本块校验值。最后在基本块尾部插入包含校验值的基本块校验指令。

(5)链接器将目标文件和静/动态库文件链接生成可执行文件,部署到支持控制流检测的硬件上执行。

通过(4)生成的目标代码,已经包含控制流检测扩展指令,代码段在硬件上执行的时候将具有控制流检测功能。而静/动态库文件则具有灵活性,可以使用已有的二进制库文件,则库文件代码段执行的时候将不具有控制流检测功能;也可以将库文件的源代码通过(1)到(4)的处理,生成具有控制流检测功能的二进制库文件。

3.2 硬件实现过程

硬件实现指扩展安全处理器指令集、增加相应硬件处理模块来支持控制流检测指令,具体实现过程如下:

(1)扩展处理器指令集

控制流检测算法中的基本块签名值和基本块校验值需要频繁访问,如果使用处理器本身的指令集,访存开销很大,会成为影响性能的瓶颈。同时对于块校验功能,每一条指令的执行都要根据当前指令,根据特定的校验算法计算块校验值,现有的硬件结构无法完成此项功能。所以有必要扩展处理器指令集,新增的指令主要包括基本块签名指令、基本块校验指令、签名寄存器压栈和出栈指令。

(2)实现基本块签名模块

基本块签名模块如图4所示。G、D和S是三个8位的寄存器,分别用来保存运行时签名G、调整签名D和静态签名S,签名差d来自签名指令中的立即数。当执行签名指令的时候,会触发硬件计算新的运行时签名 $G(G=G \oplus D \oplus d)$,然后将其存入寄存器G中。compare模块是一个比较器,它将静态签名与计算出来新的运行时签名值作比较,如果不相等,error_g信号为0,说明出现了分法的分支跳转。

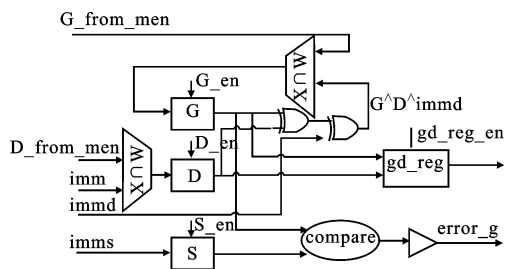


图4 基本块签名模块

(3) 实现基本块校验模块

基本块校验模块如图5所示。checkin 是签名生成模块,输入是32位的当前执行指令,输出是16位的校验值 sign,sign 不仅跟当前的输入有关还与之前的输入数据有关。当执行基本块签名指令时,checkin 模块将停止更新 sign,并将此时的 sign 与签名指令中立即数进行比较,如果不相等,error_b 信号为0,说明出现了错误的分支跳转或非控制流指令错误(包括新增控制流检测指令)。

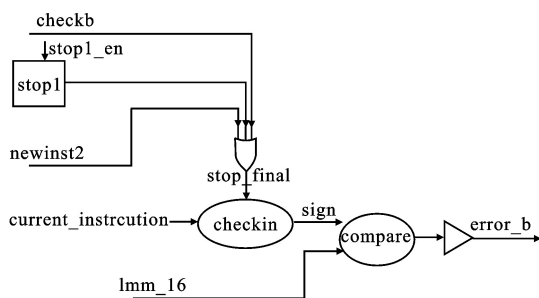


图5 基本块校验模块

4 性能分析

4.1 延迟开销

传统纯软件实现的基于基本块签名的控制流检测算法,需要对每个基本块使用通用指令完成三次签名,两次异或运算与一次比较运算,新增的大量代码带来了巨大的延迟开销。采用软硬件结合实现的控制流检测算法扩展了处理器指令集,能更充分利用硬件特性来实现基本块签名和基本块校验功能,有效减少了延迟开销。软件处理后的基本块汇编代码如表2所示,对于每个基本块,新增代码仅包括

表2 软件处理后的基本块汇编代码段

基本块	
. BLOCK_BEG	;基本块开始伪指令
SIG #0x12 #0x34	;基本块签名指令
SETD #0x01	;基本块签名指令
.....	;正常程序指令
CHKB #0x5678	;基本块校验指令
. Block_END;	基本块结束伪指令

两条基本块签名指令和一条基本块校验指令,远远小于传统纯软件实现(伪指令只是用来指导编译工作,不会生成目标代码)。

4.2 硬件开销

基于控制流检测的抗错误注入攻击的安全芯片采用了软硬件结合的实现方法,需要增加硬件用来实现基本块签名和基本块校验,这两个硬件逻辑模块结构如图4和图5所示,新增硬件开销包括寄存器、逻辑门、比较器和多路选择器,同时芯片的译码单元要做相应改动,以支持相应的扩展指令。表3显示了基于自主开发的32位处理器核 SUP320 实现的安全芯片的相应面积和等效门开销,实际新增控制流检测硬件的开销只相当于处理器核心的1%。

表3 基于 SUP320 处理器核实现的安全芯片的硬件开销

硬件	面积/cm ²	等效门
SUP320 处理器核	0.338	30 696
控制流检测模块	0.003	320

5 结束语

本文对传统控制流检测算法进行了改进,该方法采用基本块签名方法来检测非法的分支跳转,采用基本块校验方法来检测错误的分支跳转、非控制流指令错误和新增检测指令的自校验,解决了传统控制流检测算法的盲区问题。该方法采用软硬件结合的实施方式,具有小的延迟开销和硬件开销,对于安全芯片的性能影响较小。将该方法应用安全芯片设计,对于安全芯片的抗错误注入攻击提供了一种新的思路,它不再是从源头上防止错误注入攻击,而是在程序动态执行的过程中去发现错误,能更有效的抵抗错误注入攻击。

参考文献:

- [1] 曾晓洋. 信息安全芯片 SoC 平台及其应用[J]. 信息安全与通信保密, 2005(7): 358-359.
- [2] 易青松, 戴紫彬. SoC 安全芯片物理级攻击方法及安全防护探析[J]. 国外电子元器件, 2007(5): 23-30.
- [3] 叶世芳. 安全芯片物理防护研究[D]. 杭州: 浙江大学, 2005.
- [4] Yau S S, Chen F. An approach to concurrent control flow checking[J]. IEEE Trans on Software Engineering, 1980, 6(2): 126-137.
- [5] Mahmood A, Mc Cluskey E J. Concurrent error detection using watchdog process a survey[J]. IEEE Transactions on Computers, 1988, 32(2): 160-174.

(下转第 66 页)

同的通信环境,最大迭代译码次数是可以配置的。

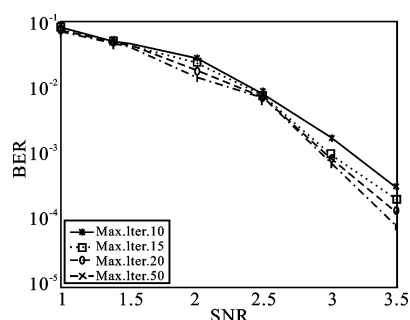


图5 不同迭代次数对译码器性能的影响

4 FPGA 实现结果

根据以上提出的 LDPC 码译码结构,成功地实现了基于 WiMAX 系统的码率为 1/2,码长为 2304 的 LDPC 码译码器.本设计采用 VHDL 语言编写,采用 Modelsim DE 10.0C 进行功能仿真,使用 ise 10.1 进行 FPGA 逻辑综合、布线,最终在 Xilinx 公司的 Virtex5 XC5VLX330T FPGA 完成验证.该译码器的硬件资源使用情况及性能如表 1 所示.

表 1 FPGA 资源消耗情况

器件	Virtex5 XC5VLX330T
ALUTs	16722
Registers	10844
Ram	54720 bit
最高时钟频率	215.3 MHz
数据吞吐率(20 次)	123 Mb/s

5 结束语

由于 BP 译码算法复杂度高,硬件实现复杂,本文采用了查找表的方法来降低译码复杂度.针对传统 BP 算法译码器系统运行频率和吞吐率低的特点,本文设计了校验节点和变量节点处理的 5 级流水结构和提前终止迭代译码策略,大大提高了系统运行的频率和数据吞吐率.在 FPGA 上的实现结果

表明基于 WiMAX 系统的码率为 1/2,码长为 2304 的 LDPC 译码器结构性能优越,其最高时钟频率达到了 215.3 MHz,迭代次数为 20 时数据吞吐率为 123 Mb/s.

参考文献:

- [1] Gallager R. Low-density parity-check codes[J]. IRE Trans. Inf. Theory, 1962(7):21-28.
- [2] MacKay D J C, Neal R M. Near shannon limit performance of low-density parity-check codes[J]. Electronics Letters, 1996(32):1645-1646.
- [3] Kou Y, Lin S, Fossorier R. Low-density parity-check codes based on finite geometries: a rediscovery and new results[J]. IEEE Transactions on Information Theory, 2001, 47(7):2711-2736.
- [4] Richardson T J, Urbanke R L. The capacity of low density parity check codes under message-passing decoding [J]. IEEE Transactions on Information Theory, 2001(47):599-680.
- [5] Cui Z, Wang Z. Area-efficient parallel decoder architecture for high rate QC_LDPC codes[C]// Proc. IEEE ISCAS. Greece, Kos, 2006:5107-5110.
- [6] 李凤飞,郝学飞,胡国荣. 一种高效的多码率 LDPC 译码器的设计[J]. 微电子学与计算机, 2011, 28(2): 23-27.
- [7] IEEE Std. 802.16e. IEEE Standard for local and metropolitan area networks Part 16: air interface for fixed and mobile broadband wireless access systems, amendment 2: physical and medium access control layer for combined fixed and mobile operation in licensed bands[S]. 2006.

作者简介:

李振辉 男, (1988-), 硕士. 研究方向为 IC 设计.

(上接第 62 页)

- [6] Oh N, Shirvani P, McCluskey E J. Control flow checking by software signatures[J]. IEEE Trans on Reliability, 2002, 51(2): 111-122.
- [7] 王永庆, 张民选. 一种改进的旋转 CRC 数据校验设计方法[J]. 国防科技大学学报, 2011, 35(6):48-54.

作者简介:

黄威 男, (1989-), 硕士研究生. 研究方向为 IC 设计.
刘恺 男, (1990-), 硕士研究生. 研究方向为 IC 设计.
廖望 男, (1990-), 博士研究生. 研究方向为 IC 设计.
戴葵 男, (1968-), 教授. 研究方向为 IC 设计.