E-KEY电子密钥系统仿真器的设计与实现

吴 瀛,贺建立
(安徽建筑工业学院 计算机与信息技术系,安徽 合肥 230022)

摘 要:介绍一种电子密钥系统仿真器的设计。它实现了目标系统中的CPU模块仿真存储器仿真、以及系统外部设备的仿真。针对该仿真器设计中难点,依据其仿真目标,提出了有效的解决方案。仿真测试实例证明该仿真器的工作正确性。实践证明,作为目标系统软件同步开发工具,该仿真器大大地加快了工程开发的进度。

关键词:仿真器;CPU仿真;外部设备仿真
中图分类号:TP391.9 **文献标识码**:A **文章编号**:1673-629X(2006)06-0145-03

Design and Implementation for an E-KEY Simulator

WU Ying, HE Jian-li
(Dept. of Computer Science, Anhui Institute of Architecture and Industry, Hefei 230022, China)

Abstract: An electric key system simulator is introduced. It achieves simulation for CPU, memory, and peripherals modules in the target system. According to the simulation aim, there provides feasible solutions to the difficulties in the simulator designing. The testing confirms the simulator's validity, and in practice, as a parallel developing tool for the target system software, this simulator accelerates the whole project's development considerably.

Key words: simulator; CPU simulation; peripherals simulation

仿真器是在宿主机上运行并能模拟目标体系结构机器行为的一种软件系统。它可以解释并执行目标体系结构机器上可执行程序,同时,记录软件系统状态及事件,为目标机系统或软件的重新设计提供历史信息。从仿真层次上,仿真器可分为门级仿真器以及系统级仿真器。本仿真器是为了确保整个USB-EKEY电子密钥系统[1]按期完成,使该项目的CPU及专用微处理机研制开发可以和建立在该电子密钥系统上的片上操作系统及应用进行同步开发而设计的。文中首先给出仿真器的设计目标、仿真器组成,分析仿真设计中难点及解决方法,进而给出具体设计方案,包括实现过程中的类设计方法及算法描述,最后基于仿真测试及实际应用对本仿真器进行了评价。

1 仿真器设计目标

1.1 仿真目标

本仿真器作为电子密钥软件设计的同步开发工具,为确保其实际有效性,它必须满足以下目标:

(1)仿真级别目标:E-KEY电子密钥仿真器是系统级仿真器,只须反应CPU中各个寄存器、内存的变化,该仿真器执行片上操作系统软件及其应用程序的开发与调试,故需要较快仿真速度。

(2)仿真时间目标:必须在有效时间内完成片上操作系统及建立于其上的应用程序的仿真。

(3)仿真准确性目标:要求仿真器上的执行结果基本上在实际目标机上可以重现。但不要求E-KEY仿真器整机状态任何时刻都与实际目标机器相一致。

(4)可扩展性目标:目标机体系结构较为稳定,本仿真器可以不考虑外部设备的可扩展性。

(5)可调试性目标:为了调试运行于目标机上的片上操作系统及建于其上的应用程序,该仿真器必须为用户提供良好的图形用户界面(GUI)及以下调试功能:a.支持功能强大的文件输入格式;b.提供观察CPU、内存、系统中寄存器以及外设控制寄存器的窗口;c.设置或修改断点,单步执行、多步执行功能;d.运行步进式仿真测试方式;e.支持日志,记录运行中发生的各种事件。

1.2 仿真器的组成

本仿真器必须包括以下内容:

(1)GDC2000 CPU仿真模块;

(2)存储器仿真模块;

(3)设备仿真模块:在E-KEY系统中,外围设备主要有DES加解密模块、RSA加解密模块、RDG随机数生成模块、USB通信模块。仿真器中设备模块的仿真从而可细分为:DES仿真模块。RSA仿真模块、RDG仿真模

收稿日期:2005-09-30
基金项目:国家"八六三"项目(2001AA141030);安徽省教育厅高等学校青年教师科研项目(2005jq1145);硕博科研启动基金资助项目(2004042)
作者简介:吴 瀛(1970-),男,安徽合肥人,讲师,研究方向为嵌入式系统仿真、Linux操作系统。

块、USB 仿真模块。

1.3 仿真器设计中的难点及解决方法

目标机中,所有的外设均是采用全硬件模块实现,完全按其数据流程仿真,将十分复杂。基于此,对于设备的仿真,仅进行功能上模拟,但保留与设备进行通信的接口及通信过程,保证仿真器上进行调试的程序与外设的通信过程与该程序在目标机上运行时对外设访问的一致,至于如加解密可靠性等问题,本仿真器不保证提供与目标机相同的能力,而是由具有知识产权的硬件提供商来保证。这样处理并不影响本仿真器作为软件开发与调试工具这一目标的实现。

2 EKeySim 仿真器的设计与实现

EKeySim[2]在结构与功能上模拟 USB E-EKey 专用微处理机系统,采用面向对象的技术,并利用了 Windows 内核提供的多线程编程技术以及消息驱动机制。

2.1 EKeySim 线程类及其继承关系

EKeySim 中的线程分为工作线程与监控线程[3]。工作线程对目标机中各个模块的行为进行模拟,监控线程管理和协调系统中所有工作线程并充当工作线程与用户界面之间的接口。

系统中主要的工作线程:

cpuThread: 中央处理器 GDC2000 工作线程;

desThread: DES 加密模块工作线程;

rsaThread: RSA 加密模块工作线程;

rdgThread: RDG 随机数产生模块工作线程;

usbThread: USB 通信模块工作线程;

系统中监控线程:monitor。

EKeySim 中工作线程类均是从 CWinThread 继承而来,其继承关系如图 1 所示。

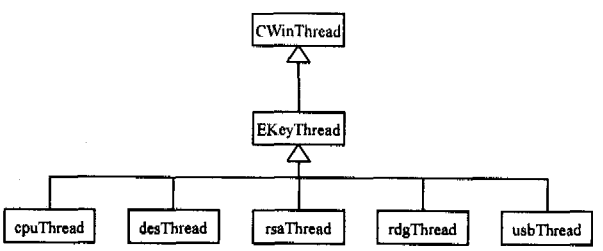


图 1 采用 UML 描述的仿真器工作线程类继承图

EKeyThread 线程类中增加了一个静态全局线程注册表,该线程类及其子类(各个工作线程)共用该注册表,以对系统中所有模块的仿真线程进行登记和管理工作。EKeyThread 线程类对其基类 CWinThread 中的 3 个虚函数 InitInstance(),ExitInstance(),Run()实行了重载,实现系统中线程的注册与注销,以及模块仿真操作流程的工作。EKeyThread 线程类增加了相应成员,实现对模块一些共同功能的模拟,并提供各工作线程可重载的虚函数 acting(),以实现对被仿真的目标模块的动作流程的模拟。

2.2 EKeySim 仿真器中的线程启停关系

仿真器中各个模块仿真线程均要经历创建、注册、运行、注销等过程,具体如下:

(1)主线程创建各子线程;

(2)各子线程注册自己;

(3)主线程启动各子线程同时进入仿真过程;

(4)子线程运行,以实现各个目标模块的仿真,仿真过程结果时,各子线程注销自己;

(5)主线程退出。

为实现仿真器中各工作线程启停关系的要求以及协调线程对仿真器中大量临界资源的访问行为,按不同资源属性,采用了 MFC 封装的同步对象,主要有信号量类(Semaphore)、互斥类(Mutex Semaphore)、临界区(Critical Section)。

2.3 EKeySim 仿真器中的主要工作线程仿真算法

仿真线程典型的消息处理流程是在该仿真线程类中的 acting()方法中进行的,流程见图 2。针对目标机中不同模块的功能,定义相应仿真线程处理的消息及其行为,从而形成不同仿真线程的算法。

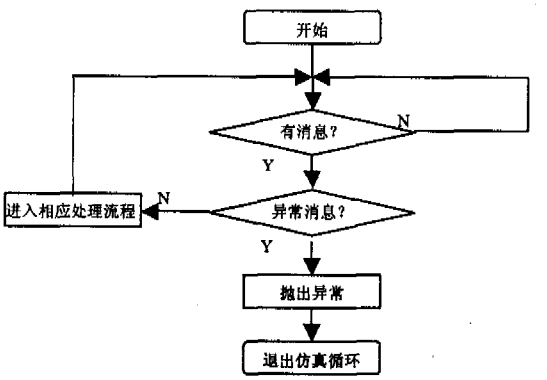


图 2 仿真线程典型消息处理流程

2.3.1 EKeySim CPU 模块仿真线程及算法

GDC2000 CPU 中算术逻辑运算模块、寄存器模块、参数栈模块、返回栈模块的行为均采用相应类来模拟,逻辑与时序控制功能由 cpuThread 类中相应过程来实现,不作为一个单独类来实现。

CPU 模块仿真线程类 cpuThread 用来创建 CPU 的工作线程,主要功能如下:

1) 模拟 GDC2000 CPU 中各个功能单元的动作;

2) 初始化 CPU 启动时刻状态,根据用户输入文件的要求设置运行模式、启动地址等。

3) 实现 DEBUG 功能。

CPU 模块仿真线程类采用多重继承技术,其父类有 EKeyThread, cpuRegister, ALU, Memory, portRegisters 等。CPU 主算法只是严格模拟目标 CPU 加工信息的流程,在此不作进一步解释。

2.3.2 加、解密模块 DES,RSA 仿真线程及其主要算法

加解密模块在目标 EKEY 系统中居于非常重要的地位,并构成电子密钥系统安全性能的基础。目标系统中,DES 以 64 位数据为分组实现数据的加解密处理;RSA 是安全性能依赖于因数分解困难程度的公钥密码系统。

DES处理相对简单,不作介绍,在此主要介绍对RSA模块的仿真设计。

目标机中,RSA模块采用中断方式进行数据的输入与输出,其特点为:

(1)采用全硬件格式进行RSA计算;

(2)采用超标量流水线方式进行数据的处理。以1024bit为一个加工组,按64bit为最小加工单元对数据进行分割;

(3)采用自适应的控制逻辑;

(4)信息加/解密的处理流程,按智能化、自动化的方式有序推进;

(5)采用了高位宽单拍乘法器与除法器(均为64bit)来加速处理过程。

笔者分析了该模块的仿真目标,认为:a.该模块采用具有知识产权的第三方设计,在本仿真器中,没必要对该硬件模块的正确性进行验证,目标机中该模块加解密正确性由硬件模块提供商保证;b.基于a.,在仿真器中,没有必要严格按目标模块加工数据的实际流程进行数据处理,但是,由于本仿真器以调试目标机片上OS以及应用程序为目标,故必须保留目标机应用程序与该模块的接口。

仿真器中,RSA仿真线程类由EkeyThread等类通过public属性多承继承方式设计,其父类有:EkeyThread,memory,portRegister。其主要算法如下:

1) RSA仿真线程非阻塞地等待来自CPU的消息,若没有消息,转1),否则转2);

2) 有消息到达,按消息进行处理;

2.1) 若消息为系统消息,则忽略;

2.2) 若消息为RSA模块初始化或停止消息,则调用相应函数处理;

2.3) 若消息为数据准备好、请求加解密运算,则:

2.3.1) 检查状态寄存器,若命令状态字中D/C位为0,则表示本次操作为加密操作,进入加密操作流程:

2.3.1.1) 设置命令状态字为加密结果数据不确定状态;

2.3.1.2) 从02号端口寄存器取第一组子密钥二进制系列,修改密钥的段长度为原长度减一,向CPU仿真线程发送中断信号,请求输入下一组子密钥系列;得到密钥下一组二进制系列后,与前面得到的子密钥二进制系列进行拼接;转2.3.1.2),直至段长度为0;

2.3.1.3) 执行明文数据的加密运算;

2.3.1.4) 设置状态寄存器为密文数据有效;

2.3.1.5) 设置CPU中断寄存器中的RSA中断请求位,请求输出密文数据;

2.3.1.6) 转1);

2.3.2) 若命令状态字中D/C位为1,则表示本次操作为解密操作,则:

2.3.2.1) 设置命令状态字为解密结果数据不确定状态;

2.3.2.2) 从02号端口寄存器取第一组子密钥二进制系列,修改密钥的段长度为原长度减一,向CPU仿真线程发送中断信号,请求输入下一组子密钥系列;得到密钥下一组二进制系列后,与前面得到的子密钥二进制系列进行拼接;转2.3.1.2),直至段长度为0;

2.3.2.3) 执行密文数据的解密操作;

2.3.2.4) 设置状态寄存器为明文数据有效;

2.3.2.5) 设置CPU中断寄存器中的RSA中断请求位,请求输出明文数据;

2.3.2.6) 转1)。

目标模块是分段进行加解密运算,即首先采用第一组子密钥进行加解密运算,第一组子密钥二进制系列移空后,再发出中断要求第二组子密钥系列。仿真器对该过程的处理是首先组建自己的完整密钥二进制系列,在组建过程中,仍通过中断服务获得子密钥,只有完整的密钥组建好以后,才执行加密或解密运算。这使得RSA仿真线程更易于设计、调试,通过中断服务获取子密钥系列使其与应用程序的接口仍然得到完整的保留。

2.3.3 随机数产生模块RDG仿真线程

该模块的仿真线程同样采用多重继承技术,其父类为EkeyThread, cpuRegister, portRegister。16位随机数由C++中伪随机数产生函数来模拟,随机数的种子采用本机系统的当前时间转换而来,其算法简单,在此不作描述。

2.3.4 USB模块的仿真

目标机中CPU对USB模块的读写操作进行了特殊的处理。目标机系统通过USB模块将数据处理结果显示给用户或者通过该模块从用户得到输入数据。仿真器中,设计了一个文本窗口,数据的输入输出均通过该窗口来实现,且该仿真线程与CPU仿真线程的通信过程严格遵循目标机的要求。

该模块的仿真线程同样采用多重继承技术,其父类为EkeyThread,cpuRegister,portRegister。在此不作描述。

3 仿真器的测试

为验证EKeySim仿真器,对仿真器分别采用目标机代码程序段编译的目标代码程序进行了测试,主要是测试DES加解密仿真模块、RSA加解密仿真模块,其输入输出数据如表1所示。

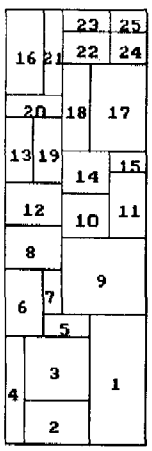
表1　仿真器加解密测试数据

		RSA*	DES**
加密	加密密钥	e=65537 n=29476058360569827577652156935129661681247772943	0123456789abcdef
	明文	M=34871571057140357415157147581431374434l5	0123456789abcde7
	密文	C=22410846520901923626549836010073840896508 29	C95744256a5ed31d
解密	解密密钥	d=14001543751207704080440701502854928318009950953 n=29476058360569827577652156935129661681247772943	0123456789abcdef
	密文	C=22410846520901923626549836010073840896508 29	C95744256a5ed31d
	明文	M=34871571057140357415157147581431374434l5	0123456789abcde7

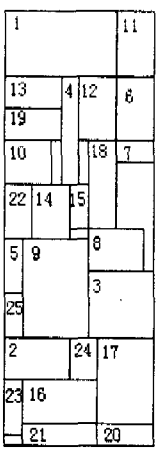
*) RSA加解密对应的两个素数为:p = 149137095665811291858500 3;q = 19764404173875180295981

**) DES加解密的密钥、输入输出数为十六进制数

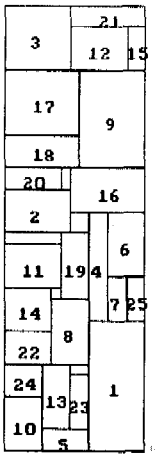
(下转第221页)



(a) 文献[1]中实例1原图(宽40高15)



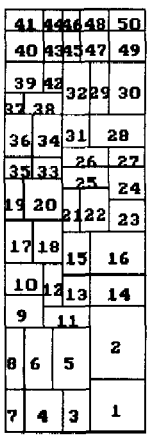
(b)文献[2]所得排样图(宽16高40)利用率93.75%



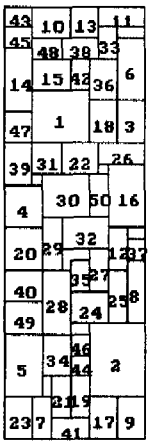
(c)文中所得排样图(宽15高41)利用率97.56%

图 3　实例 1 原图、文献[2]所得排样图和文中所得排样图

例 2:50 块矩形件进行排样(图 4(a)、(b)分别为实例 2 原图和文中所得排样图)。



(a)文献[1]中的实例 2 原图(宽 40 高 15)



(b)文中所得排样图(宽 15 高 41)利用率 97.56%

图 4　实例 2 原图和文中所得排样图

由上述两个例子可以看出在文献[1,2]中都对这两个算例用不同的方法进行了求解,但效果都不是很好,文献[1]所得最好结果为宽 40 高 17,利用率为 88.24%,文献[2]所得最好结果为宽 16 高 40,利用率为 93.75%。而文中通过将遗传算法与剩余矩形排样算法进行结合,在设置种群大小为 300、终止代数为 50 代的条件下,得到了比以往更好的解,板材利用率达 97.56%。

为进一步验证方法的有效性,又随机模拟了 10 个 25 块矩形排样的例子,运用上述方法求解,均得到了近似最优解,成功率较高。充分说明该方法用于求解矩形件正交排样问题具有较高的有效性和可行性。

5　结　论

探讨了矩形件正交排样优化问题的遗传算法求解,将其转化为排列问题,并提出了将排列转化为相应排样图的剩余矩形排样算法,讨论了此混合遗传算法的具体实现。并通过实例验证表明这种方法与传统方法相比具有更高的准确性、有效性以及可行性。尽管如此,仍然未能求得矩形件排样问题的最优解,这还有待于今后作进一步的研究改进。

参考文献:

[1] Jakobs S. Theory and Methodology on Genetic Algorithms for the Packing of Polygons[J]. European Journal of Operational Research, 1996, 88: 165 - 181.

[2] 刘德全,滕弘飞.矩形件排样问题的遗传算法求解[J].小型微型计算机系统,1998,19(12):20 - 25.

[3] 贾志欣,殷国富,罗　阳,等.矩形件排样的模拟退火算法求解[J].四川大学学报(工程科学版),2001,33(5):35 - 38.

[4] Healy P. An Optimal Algorithm for Rectangle Placement[J]. Operations Research Letters, 1999, 24: 73 - 80.

[5] 李满江,孟祥旭,王志强.矩形件和任意多边形排样问题的算法及应用[J].贵州工业大学学报(自然科学版),2002,31(4):126 - 141.

[6] 刑文训,谢金星.现代优化计算方法[M].北京:清华大学出版社,1999.129 - 136.

(上接第 147 页)

对于以上结果,采用相应的成熟加解密算法[4,5]验证了其结果正确性。以上测试了仿真器 DES, RSA 加解密功能,实际上也基本测试了整个仿真器的工作正确性。

4　结　论

文中所介绍的仿真器是为拥有自主版权的 GX0108 智能令牌系统的开发而设计的,在工程应用中,该仿真器使得 GX0108 智能令牌系统的系统硬件与软件的设计同步进行,大大加快了系统设计速度,节省了项目开发周期。由于时间与能力有限,该仿真器未能实现电子密钥体系结构及 GDC2000 CPU 的评测功能,这也是本仿真器未来进一步努力的方向。

参考文献:

[1] 国芯安集成电路设计有限公司. E - KEY 体系结构概述[R].[出版地不详]:国芯安集成电路设计有限公司,2002.

[2] Magnusson P S. SimICS/sun4m[Z]. Sweden: Swedish Institute of Computer Science, 1998.

[3] Kruglinkski D J. VC + + 技术内幕[M]. 王国印译. 北京:清华大学出版社,1998.

[4] 卢开澄. 计算机密码学(第 2 版)[M]. 北京:清华大学出版社,2000.

[5] Hottice. 如何产生素数[EB/OL]. www. vccode. com, 2003.