Journal of Computer Applications
计算机应用,2015,35(5):1412 - 1416
ISSN 1001-9081
CODEN JYIIDU
2015-05-10
http://www.joca.cn

文章编号:1001-9081(2015)05-1412-05
doi:10.11772/j.issn.1001-9081.2015.05.1412

嵌入式系统芯片中 SM2 算法软硬件协同设计与实现

钟 丽[1],刘 彦[2],余思洋[2],谢 中[1*]

(1. 湖南大学 物理与微电子科学学院,长沙 410082; 2. 湖南大学 信息科学与工程学院,长沙 410082)

(* 通信作者电子邮箱 xiezhong@hnu.edu.cn)

摘 要: 针对现有的椭圆曲线算法系统级设计中开发周期长,以及不同模块的性能开销指标不明确等问题,提出一种基于电子系统级(ESL)设计的软硬件(HW/SW)协同设计方法。该方法通过分析 SM2(ShangMi2)算法原理与实现方式,研究了不同的软硬件划分方案,并采用统一建模语言 SystemC 对硬件模块进行周期精确级建模。通过模块级与系统级两层验证比较软硬件模块执行周期数,得出最佳性能划分方式。最后结合算法控制流程图(CFG)与数据流程图(DFG)将 ESL 模型转化为寄存器传输级(RTL)模型进行逻辑综合与比较,得出在 180 nm CMOS 工艺,50 MHz 频率下,当算法性能最佳时,点乘模块执行时间为 20 ms,门数 83 000,功耗约 2.23 mW。实验结果表明所提系统级架构分析对基于椭圆曲线类加密芯片在性能、面积与功耗的评估优势明显且适用性强,基于此算法的嵌入式系统芯片(SoC)可根据性能与资源限制选择合适的结构并加以应用。

关键词: SM2 算法; SystemC; 软硬件划分; 电子系统级; 周期精确

中图分类号: TP302.1 **文献标志码:** A

Hardware/Software co-design of SM2 encryption algorithm based on the embedded SoC

ZHONG Li[1], LIU Yan[2], YU Siyang[2], XIE Zhong[1*]

(1. *College of Physics and Microelectronics, Hunan University, Changsha Hunan* 410082, *China*;
2. *College of Information Science and Engineering, Hunan University, Changsha Hunan* 410082, *China*)

Abstract: Concerning the problem that the development cycle of existing elliptic curve algorithm system level design is long and the performance-overhead indicators are not clear, a method of Hardware/Software (HW/SW) co-design based on Electronic System Level (ESL) was proposed. This method presented several HW/SW partitions by analyzing the theories and implementations of SM2 algorithm, and generated cycle-accurate models for HW modules with SystemC. Module and system verification were proposed to compare the executing cycle counts of HW/SW modules to obtain the best partition. Finally, the ESL models were converted to Rigister Transfer Level (RTL) models according to the CFG (Control Flow Graph) and DFG (Data Flow Graph) to perform logic synthesis and comparison. In the condition of 50 MHz, 180 nm CMOS technology, when getting best performance, the execute time of point-multiply was 20 ms, with 83 000 gates and the power consuption was 2.23 mW. The experimental result shows that the system analysis is conducive to performance and resources evaluation, and has high applicability in encryption chip based on elliptic curve algorithm. The embedded SoC (System on Chip) based on this algorithm can choose appropriate architecture based on performance and resource constraints.

Key words: SM2 algorithm; SystemC; hardware/software partition; Electronic System Level (ESL); cycle-accurate

0 引言

随着移动支付终端、智能卡、无线通信等嵌入式设备的普及,数据的安全性也面临越来越大的挑战。作为密码类嵌入式系统的核心部件,算法模块的性能、功耗、开发周期等要求在系统设计中越来越重要。SM2 算法是国家密码管理局于 2010 年 12 月发布的椭圆曲线公钥密码算法,并要求对现有基于 RSA 算法的电子认证系统、密钥管理系统等进行升级改造,因安全性高而被广泛应用于加密芯片中。由于 SM2 算法复杂度高、执行周期长,以及密码类嵌入式系统对性能开销要求严格,使得基于该算法的系统芯片(System on a Chip, SoC)需要在设计初期对算法模块进行软硬件划分,根据不同的需求选择合适的划分方式以平衡性能和开销。目前基于 SM2 算法的 SoC 芯片中,主要采用基于指令集的协处理器以及基于总线协议的外设 IP(Intellectual Property)作为硬件加速模块,前者更高效,后者简单、易实现,但两种实现方式依然涉及对算法模块本身的软硬件划分。

基于软件的实现方式灵活性高,成本低,但对于嵌入式 RISC(Reduced Instruction Set Computer)微处理器,当终端设备对运算速度有较高的要求时,其运行效率难以达到所需运算能力。采用专用集成电路的实现方式能显著提高算法整体运行效率,缺点是增加了芯片的面积和成本。文献[1]分析了二元扩域椭圆曲线实现过程及软硬件协同设计思想,但是没有提出具体划分方式,文献[2-3]提出基于椭圆曲线的协

收稿日期:2014-12-10;修回日期:2015-01-20。 基金项目:国家自然科学基金资助项目(61300037)。

作者简介:钟丽(1990-),女,安徽合肥人,硕士研究生,主要研究方向:数字芯片设计与验证; 刘彦(1979-),男,湖南长沙人,讲师,博士,主要研究方向:计算机体系结构、可重构计算、多处理器任务调度; 余思洋(1986-),男,湖南岳阳人,博士研究生,主要研究方向:加密芯片; 谢中(1957-),男,湖南攸县人,教授,博士,主要研究方向:无线电物理。

处理器设计,能快速实现点乘运算,但是缺少对算法实现的软硬件划分。文献[4]主要涉及对椭圆曲线算法指令集设计以及基于指令集的相关模块的性能、时间和存储开销比较,对算法模块的架构分析与性能表征不足。文献[5]提出椭圆曲线算法 SoC 架构,但没有对不同架构下性能与资源的具体分析和比较。

本文主要工作包括: 首先提出所设计的 SoC 系统整体结构,分析 SM2 算法加解密实现原理,提出不同的软硬件划分方式; 然后基于上述划分方式,使用 SystemC 语言对 SM2 算法硬件实现部分进行周期精确级建模; 接着通过模块级与系统级仿真,将软件实现的执行周期与硬件模块执行周期数相比较,得到最佳性能的划分方式; 同时将寄存器传输级(Rigister Transfer Level, RTL)模型在 180 nm 互补金属氧化物半导体(Complementary Metal Oxide Semiconductor, CMOS)工艺下进行实现,通过比较面积和功耗得到性能和资源的平衡。实验结果表明对基于椭圆曲线算法的嵌入式 SoC 芯片或专用集成电路设计,用户可根据实际芯片对性能、功耗与面积的要求,结合文中所提出的几种软硬件划分方式与分析,选择合适的结构加以应用。

1　系统结构分析

1.1　基于 CortexM0 的 SoC 设计

文献[5]提出的 SoC 架构中,采用 32 位 RISC 处理器 C310S 处理器作为控制核心,所有外设包括杂凑部件均通过系统总线互联。文献[6]研究了基于 8051 微控制器的 160 位椭圆曲线算法模运算层软硬件协同设计。在本文系统中,采用 ARM CortexM0 作为主控制器,SM2 算法硬件部分作为系统外设连接到总线,并通过总线控制器与主处理器及其他模块进行通信,其他模块主要包括外部存储器设备、顶层控制逻辑、随机数发生器等。对于杂凑部件与密钥扩展部分已经在 SM2 算法软硬件划分中完成,系统如图 1 所示。

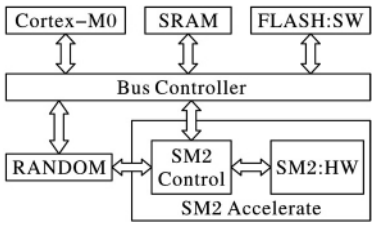

图 1　系统 SoC 框图

软硬件划分是系统级设计的关键,通过协同设计与验证,对复杂系统架构进行探索和优化,同时结合现有成本,对速度、面积、功耗进行平衡。图中 SW(Software)包括系统固件、应用程序及算法软件实现部分,存储于非易失性只读存储器 FLASH 中。SM2 Accelerate 作为算法的硬件加速器,包括算法硬件实现模块(Hardware,HW),以及与总线接口进行通信的控制模块 SM2 Control,控制模块不仅实现加速模块与主控制器之间的数据的存储与读取,同时从随机数发生器 RANDOM 读取用于生成点乘运算所需的随机数 k。随机数发生器根据 SM2 算法协议规定,需要采用国家密码管理局批准的随机数发生器,由于占用资源多,将其作为一个单独的硬件模块实现,这样不仅能有效增加运算性能,同时提高了系统的安全性。Cortex-M0 处理器是整个系统的控制核心。SRAM 用于保存程序运行的中间结果。

1.2　SM2 加解密协议与软硬件划分

根据 SM2 公钥加密算法标准[7],消息发送者 A 和消息接收者 B,对于加密,用户 B 产生公钥 Pb,用户 A 利用 Pb 根据加密流程对要发送的消息 M 进行加密,输出密文 C 并发送给用户 B。对于解密,用户 B 需要利用自己的私钥结合解密流程从密文 C 中获取原始消息 M。算法采用的椭圆曲线方程为: $y^2 = x^3 + ax + b$,通过指定 a、b 系数,确定了唯一的标准曲线。同时,为了将曲线映射为加解密算法,SM2 标准还确定了其他参数供算法程序使用。

椭圆曲线密码学算法的密钥长度是可变的,SM2 标准规定使用 256 位,这对算法的性能优化要求更高。SM2 属于计算密集型算法,且具有很好的层次性,其实现结构从底层到顶层可分为:

1) 模运算层: 模加、模减、模乘、模逆;

2) 点运算层: 点加、倍点;

3) 多倍点运算层: 点乘 $Q = [k]P$;

4) 应用层: 加解密、密钥交换、数字签名等。

其中点运算通过调用模运算实现,多倍点运算则通过多次调用点加和倍点运算实现。每一级运算根据其复杂度的不同来选择用软件或者硬件实现,不适当的划分方式会使得计算周期长且资源开销大。

本文主要对素域 256 位 SM2 加解密应用进行系统级软硬件划分,划分的主要目的在于平衡算法的整体性能与开销。

这里将性能转化为对算法模块执行周期数的比较,包括两点内容: 第一点是算法模块本身的执行时间,第二点是模块在算法实现过程中的被调用的次数。

$$T(total) = \sum_{i=0}^{n_1} u_1 + \sum_{i=0}^{n_2} u_2 + \sum_{i=0}^{n_3} u_3 + \cdots + u$$

其中: $T(total)$ 为整个算法运行周期数; $u_1, u_2, u_3, \cdots$ 为硬件子模块的运行时间; $n_1, n_2, n_3, \cdots$ 为各个模块在整个算法应用中与软件控制流程的通信次数; u 为顶层软件流程执行时间。开销主要包括算法在实现过程中所占芯片面积以及功耗。

根据上述描述,需要分别对 SM2 加解密算法四个层次进行建模仿真。除了全部使用软件实现以及全部使用硬件实现,可以得出的划分方式如表 1。

表 1　软硬件划分架构

结构	模运算	点运算	多倍点	应用层
结构 1	HW	SW	SW	SW
结构 2	HW	HW	SW	SW
结构 3	HW	HW	HW/SW	SW
结构 4	HW	HW	HW	H/SW

表 1 所示的划分方式中,对硬件部分采用周期精确级建模,软件运行于 ARM 处理器,综合比较软硬件执行周期数即可得出最佳性能的划分方式。对细化后的 RTL 模型进行实现,则可比较各模块的面积功耗开销。

2　系统建模与分析

2.1　基于 ESL 的周期精确级建模

系统建模时不同的模型对系统性能比较以及后期实现有很大影响。基于电子系统级(Electronic System Level, ESL)的

设计与验证方法能从系统级到时序级对芯片整体架构进行软硬件划分和仿真，包括高层次建模、系统集成、架构探索、高级综合等[8]，设计者可以通过增强不同抽象层次之间的移植性，实现各阶段测试平台的复用，达到快速收敛的目的，克服了传统基于C/C++系统设计方法在架构探索、硬件建模及系统验证方面的时间长、模型不准确等局限性[9]。把高层次建模语言细化为寄存器传输级模型可以采用高层次综合技术，或者手动转化。由电子设计自动化(Electronic Design Automation，EDA)综合出的RTL代码虽然包含控制通路和数据通路，但代码可读性依然不高，出现错误难以调试。相比手动转化C模型与ESL模型，后者只需定义合适的状态机并结合原有的数据/控制流程图，即可得到对应RTL模型，过程简单快速。因此基于周期精确的ESL模型更利于硬件实现。

文献[10-11]均证明了基于ESL的协同验证方式快速高效。文献[11]提出一种基于SystemC周期精确级模型的椭圆曲线SoC软硬件协同设计，但其主要集中在椭圆曲线底层模运算，缺少对算法不同层次架构的划分与建模，同时没有对底层实现的分析，因而无法全面评估算法性能与开销。

本文的设计采用SystemC语言，通过添加对算法时序信息的描述，结合上述软硬件划分架构，对SM2算法底层到顶层各个模块进行周期精确级建模，首先描绘出SM2算法不同层次各个模块的控制流程图(Control Flow Graph，CFG)和数据流程图(Data Flow Graph，DFG)，并将其转化为统一建模语言(Unified Modeling Language，UML)状态图，最后用SystemC进行描述。进行模型细化时，通过定义状态机，结合CFG与DFG，将ESL模型转化为RTL，通过仿真比较两者的执行周期可以得出所建立模型的精确性，并对RTL模型进行逻辑综合，进而评估算法的资源开销。

2.2 硬件模型

对于素数域256位点乘运算，运算数据为应用程序写的点$P(x,y)$和随机数发生器生成的随机数k，完成运算$Q=[k]P$。点乘运算是通过多次调用点加和倍点运算完成，次数为密钥长度。而最底层的模加、减等运算是最基本的运算部件，上层运算可以通过对底层模块的调用完成，如图2所示。

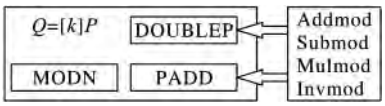

图2 SM2算法顶层模型框图

Addmod、Submod、Mulmod、Invmod分别为底层模加、减、乘、除运算。DOUBLEP用于相同点倍点运算，而非互逆的不同两点相加则调用模块PADD。本设计中模乘算法采用的是基为2的改进Montgomery模乘算法，可用于消除求模操作中的大数除法运算。该算法的运算结果会多生成一个$2^{(-k-2)}$，所以为了得到正确的计算结果，需要同时以$AB2^{(-k-2)} \bmod n$与$2^{(2k+4)} \bmod n$作为输入，则得到输出$AB \bmod n$。因此设计了一个预处理模块MODN用于输出$2^{(2k+4)} \bmod n$，与以AB作为输入进行的第一轮模乘的结果作为输入进行第二轮模乘，即可得到正确的模乘结果。$Q=[k]P$总的实现过程如下：

1) 设置$Q=0$；

2) 计算MODN；

3) k从255到0执行：

3.1) $Q=2Q$；

3.2) 当$k=1$，$Q=Q+P$；

4) 输出Q。

根据以上描述，得到点乘运算的CTF及DFG，采用SystemC语言以及相关行为级控制语句，在必要的循环之间插入wait()等待，实现周期精确。

对算法的顶层建模时，除了需要考虑底层模运算的实现，还需要对中间过程进行详细的设计。椭圆曲线上的点有多种坐标表示方式，我们的设计在仿射坐标下，同时考虑到非互逆不相同点相加($P \neq \pm Q$)与倍点($P=Q$)规则，兼顾实现过程的简单易用，设计了表2所示的几个模块。

表2 中间运算模块

模块名	运算内容	适用规则
Getx	$\lambda^2-x_1-x_2$	点加、倍点
Gety	$\lambda(x_1-x_3)-y_1$	点加、倍点
Multil	$(3x_1^2+a)/2y_1$	倍点

通过对这3个中间模块与底层模运算模块的调用及传参，即可实现不同规则下的点运算。图3所示为点乘运算控制流程。

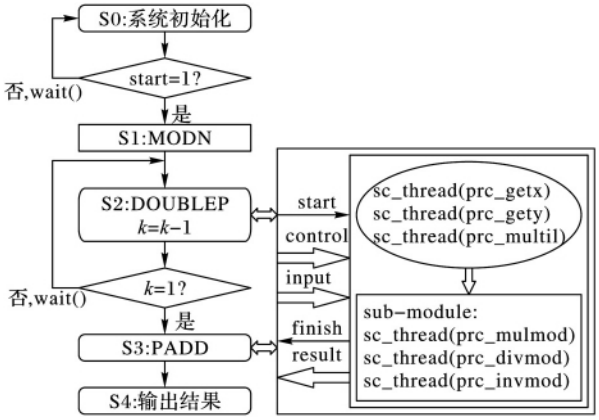

图3 基于SystemC的顶层建模

SM2算法除了核心点乘运算模块，对于应用层加解密流程，主要用到三类辅助函数，分别是杂凑函数、密钥派生函数和随机数发生器函数。杂凑函数用于将一个比特串映射为一个固定长度比特串的函数，与其他椭圆曲线加解密应用不同的是，协议规定杂凑函数使用国密SM3算法，其安全性与SHA-256相当，但高于SD-5、SHA-224等[12]；辅助函数虽不是算法执行的核心部件，但其运算强弱也会影响加解密算法的安全性。本文主要对SM3算法采用相同的建模方式，比较其软硬件架构对算法性能的影响。

2.3 基于TomMath库的软件设计

软件设计包括了算法模块的设计以及流程控制，它的作用一方面作为参考模型，用来验证硬件模型的正确性；另一方面通过比较软硬件执行周期数，得出合理的划分方式。实验使用TomMath库对算法模块进行实现，有别于SUN公司提供的基于OpenSSL实现方式的是，TomMath库编译后可以直接运行在ARM处理器。该库包含了一系列模运算，通过其自带的数据类型以及函数调用可以很方便地实现算法底层到顶层各个模块。

对于算法应用层的流程控制，则主要包括：1) 初始化输入数据，传递参数；2) 算法辅助函数的实现，包括密钥扩展函

数、杂凑函数等; 3) 输出结果,对结果进行分析。本实验采用 ARM 内核 IP,所以通过添加存储器等外设,即可让软件编译后直接运行在内核上,通过在线仿真与调试得到与相应模块的运行效率,过程简单,结果准确。

3　实验结果分析

在验证过程中,ESL 与 RTL 模型的仿真均采用 Linux 环境下 VCS(Verilog Compile Simulator) 仿真器,主要包括系统级仿真与模块级仿真。模块级仿真主要针对 256 位素域模运算与点运算,将结果与软件运算的结果进行比较可以得知模型的正确性,同时,重点通过比较两者的执行周期数判断模块性能。

3.1　性能比较

不同的参数使得算法的运算时间不同,特别是模 n 的选取,会使得软件运算结果有很大不同。实验采用 SM2 标准加密算法推荐使用素数域 256 位参数的椭圆曲线,在计算软硬件执行周期过程中,通过改变最高 4 位的值,循环计算并取平均值,得到更加精确的结果。以模乘运算为例,分别取 n 的最高 4 位为 0(252) , 1(253) , 3(254) , 7(255) , f(256) ,对应横坐标输入也是同样的变化,其结果如图 4、图 5 所示。

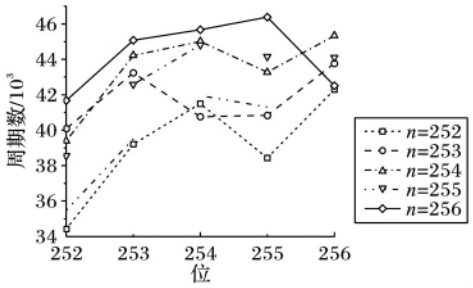

图 4　模乘(软件) 运算周期数

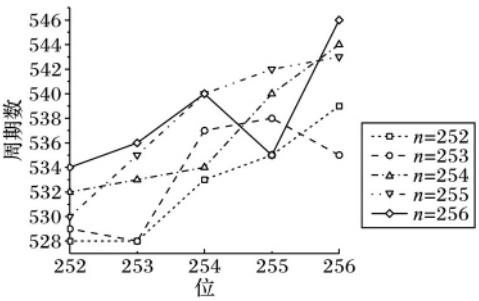

图 5　模乘(硬件) 运算周期数

由此可以得出用软件与硬件实现 256 位模乘平均需要周期数分别为 42 293 与 536。对于算法其他模块的执行周期数如表 3 所示。

表 3　软硬件周期数比较

实现方式	算法模块			
	模加减	模逆	点加	倍点
SW	10 020	489 494	11 861	11 055
ESL	99.5	958	1 467	1 038

根据以上数据,可以得出对于 256 位素域模运算,采用硬件实现模加减要相对软件的加速比为 100.7,模乘加速比为 80,而模逆由于运算复杂度高,硬件模块加速比能达到 500。对于点加和倍点运算,硬件加速比为 10。

根据第 2 章的分析,在进行软硬件划分时除了要考虑模块本身的执行周期,还需要考虑软硬件之间的通信次数。由于顶层点乘模块会多次调用点加与倍点运算,次数为密钥长度 256,所以综合模块运算时间与调用次数,硬件实现点乘相较于软件实现所需周期数,其倍数能提高 500 倍,性能得到大幅度提高。对于应用层杂凑部件 SM3,本文实验中软件实现该模块需要周期数 28 900,由于模块较小,在整个流程中只调用一次,为了减小面积及功耗,增加应用的灵活性,选择用软件实现。

在时钟频率为 50 MHz 下对加解密进行整体评估,用软件完成一次加密算法超过 10 s。若采用点乘部分作为硬件加速模块,点乘时间为 20 ms,完成一次加密需 40 ms,对应解密运算由于比加密少计算一次点乘,所需时间为加密时间的一半。因此针对以上不同的划分方式以及相应的仿真结果,可以得出采用结构 3,即硬件实现点乘及其以下运算模块,其他模块采用软件实现可使得算法性能最佳。

3.2　算法实现与资源开销比较

根据 2.1 节所述,将 ESL 模型转化成对应 RTL 状态机时,通过比较,其周期数相似度达 95% 以上,证明了该 ESL 模型的准确性。

算法硬件加速模块在提升运算性能的同时也导致芯片面积和功耗增加。对经转化得到的 RTL 模型,在 Design Compiler 和 GSMC 180 nm CMOS 工艺,1.62 V 供电条件下,通过编写相应约束文件进行逻辑综合,得到不同模块面积和功耗如表 4 所示。

表 4　各模块面积/功耗比较

模块名	面积/μm^2	功耗/mW
MODN	109 544	0.033
PADD	3 524 840	1.338
DOUBLEP	1 880 673	0.747
Invmod	292 930	0.078
Mulmod	271 231	0.123
kP_TOP	5 580 892	2.229

表 4 数据显示点加和倍点运算占用芯片大部分面积,分别是模运算的 7 倍、12 倍,两者功耗和则占模块总功耗 94%。因此对于成本受限或功耗要求严格的系统,可选择在牺牲相关性能的情况下,采用结构 1 与结构 2 结合,硬件实现模运算,同时对点加和倍点进行划分,采用软件或者软硬结合的方式实现。

4　结语

本文基于椭圆曲线算法实现原理,对集成于 ARM 处理器的 SM2 算法提出几种软硬件划分方式,通过电子系统级设计与统一建模语言对其进行周期精确级建模与仿真,快速有效地评估不同架构下的运行效率,同时将所得 ESL 模型转化为寄存器传输级模型,比较不同结构的面积和功耗,进而判断性能与资源开销的平衡。对于采用基于椭圆曲线加解密、密钥交换等算法的嵌入式 SoC 芯片开发的用户,可以在本文的分析基础上,用户可以结合自己实际需求,选择合适的架构并予以实现。

未来可进一步关注算法硬件加速模块 VLSI 实现架构的优化,并将软硬件集成到处理器内核对多种应用进行验证。

为了将其封装为通用IP,还需考虑不同架构下的通用性及可配置性。

参考文献:

[1] AL-SOMANI T F, KHAN E A, QAMAR-UL-ISLAM A M, *et al*. Hardware/Software co-design implementation of elliptic curve ccryptosystems[J]. Information Technology Journal, 2009, 8(4): 403 - 410.

[2] TAN F. High-speed coprocessor implementation of ECC algorithm [D]. Xi' an: Xidian University, 2009. (谈飞洋. 高速ECC算法协处理器设计[D]. 西安: 西安电子科技大学, 2009.)

[3] ZHOU F, SHI Z, GUO W, *et al*. High parallel configurable elliptic curve cryptographic processor over GF(p) field[J]. Computer Engineering, 2012, 38(16): 142 - 144. (周发旺, 史再蜂, 郭炜, 等. 高并行可配置的GF(p)域ECC处理器[J]. 计算机工程, 2012, 38(16): 142 - 144.)

[4] XU J, WANG Z, YAN Y. HW/SW co-design of ECC ASIP[J]. Computer Engineering and Design, 2012, 33(3): 916 - 920. (徐劲松, 王志新, 严迎建. ECC专用指令处理器软硬件协同设计[J]. 计算机工程与设计, 2012, 33(3): 916 - 920.)

[5] ZHANG L, CHEN J, HUANG Y, *et al*. Research and design of elliptic curve cryptosystem SoC[J]. Journal of Huazhong University of Science and Technology: Natural Science Edition, 2008, 36(11): 52 - 55. (张丽娜, 陈建华, 黄尹, 等. 椭圆曲线密码SoC的研究与设计[J]. 华中科技大学学报: 自然科学版, 2008, 36(11): 52 - 55.)

[6] SAKIYAMA K, BATINA L, PRENEEL B. HW/SW co-design for public-key cryptosystems on the 8051 micro-controller[J]. Computers and Electrical Engineering, 2007, 33(5/6): 324 - 332.

[7] State Cryptography Administration. Public key cryptographic algorithm SM2 based on elliptic curve[EB/OL]. [2010 - 12 - 10]. http://www.oscca.gov.cn/. (国家密码管理局. SM2椭圆曲线公钥密码算法总则[EB/OL]. [2010 - 12 - 10]. http://www.oscca.gov.cn/.)

[8] YOU Y. SoC design based on ESL design methodology[J]. China Integrated Circuit, 2011(9): 29 - 35. (游余新. 基于ESL设计方法学的SoC设计[J]. 中国集成电路, 2011(9): 29 - 35.)

[9] LI H. CHEN X. Electronic system level design based on SystemC [M]. Beijing: Science Press, 2010: 9 - 11. (李挥, 陈曦. SystemC电子系统级设计[M]. 北京: 科学出版社, 2010: 9 - 11.)

[10] HAU Y W, KHALIL-HANI M. SystemC-based HW/SW co-simulation platform for System-on-Chip (SoC) design space exploration [J]. International Journal of Information and Communication Technology, 2009, 2(1/2): 108 - 119.

[11] HAU Y W, KHALIL-HANI M, MARSONO M N. SystemC-based hardware/software co-design of elliptic curve cryptographic system for network mutual authentication[J]. Malaysian Journal of Computer Science, 2011, 24(2): 111 - 130.

[12] SUN R, CAI C, ZHOU Z, *et al*. The comparision between digital signature based on SM2 and ECDSA[J]. Network Security Technology and Application, 2013(2): 60 - 62. (孙荣燕, 蔡昌曙, 周洲, 等. 国密SM2数字签名算法与ECDSA算法对比分析研究[J]. 网络安全技术与应用, 2013(2): 60 - 62.)

(上接第1384页)

[8] ZHAO Y, ZHOU F, FAN X, *et al*. IDSRadar: a real-time visualization framework for IDS alerts [J]. Science China Information Sciences, 2013, 56(8): 1 - 12.

[9] HUMPHRIES C, PRIGENT N, BIDAN C, *et al*. ELVIS: Extensible Log VISualization [C]// Proceedings of the 10th Workshop on Visualization for Cyber Security. New York: ACM, 2013: 9 - 16.

[10] ALSALEH M, ALQAHTANI A, ALARIFI A, *et al*. Visualizing PHPIDS log files for better understanding of Web server attacks [C]// Proceedings of the 10th Workshop on Visualization for Cyber Security. New York: ACM, 2013: 1 - 8.

[11] LI B, SPRINGER J, BEBIS G, *et al*. A survey of network flow applications [J]. Journal of Network and Computer Applications, 2013, 36(2): 567 - 581.

[12] LAI J, WANG H, JIN S. Study of network security situation awareness system based on Netflow [J]. Application Research of Computers, 2007, 24(8): 167 - 172. (赖积保, 王慧强, 金爽. 基于Netflow的网络安全态势感知系统研究 [J]. 计算机应用研究, 2007, 24(8): 167 - 172.)

[13] SCARFONE K, MELL P. Guide to Intrusion Detection and Prevention Systems (IDPS) [K/OL]. [2014 - 06 - 20]. http://wenku.baidu.com/link?url=JICw2pW5FMQQpKbU1TBfcFGrz551tCfIIbjAysZsRQJDfiG3-BOxvOZeczL1_sNflAq_r9J0iDEoz-K8Mwsnt4ofLO6_rxqNUZO9fbUSZ_.

[14] NEWMAN R. Computer security: protecting digital resources [M]. Burlington: Jones & Bartlett Publishers, 2009: 273 - 275.

[15] GUERRA-GOMEZ J, BUCK-COLEMAN A, PACK M, *et al*. TreeVersity: Interactive visualizations for comparing hierarchical datasets [J/OL]. [2014 - 06 - 20]. http://hcil2.cs.umd.edu/trs/2012-14/2012-14.pdf.

[16] ZHANG X, YUAN X. Treemap visualization [J]. Journal of Computer-Aided Design and Computer Graphics, 2012, 24(9): 1113 - 1124. (张昕, 袁晓如. 树图可视化 [J]. 计算机辅助设计与图形学学报, 2012, 24(9): 1113 - 1124.)

[17] KRSTAJIC M, KEIM D. Visualization of streaming data: observing change and context in information visualization techniques [C]// Proceedings of the 2013 IEEE International Conference on Big Data. Piscataway: IEEE, 2013: 41 - 47.

[18] ROPINSKI T, OELTZE S, PREIM B. Survey of glyph-based visualization techniques for spatial multivariate medical data [J]. Computers and Graphics, 2011, 35(2): 392 - 401.

[19] CHEN Y, HU H, LI Z. Performance compare and optimization of ractangular treemap layout algorithms [J]. Journal of Computer-Aided Design and Computer Graphics, 2013, 25(11): 1623 - 1634. (陈谊, 胡海云, 李志龙. 树图布局算法的比较与优化研究 [J]. 计算机辅助设计与图形学学报, 2013, 25(11): 1623 - 1634.)

[20] KRSTAJIC M, BERTINI E, KEIM D. Cloudlines: Compact display of event episodes in multiple time-series [J]. IEEE Transactions on Visualization and Computer Graphics, 2011, 17(12): 2432 - 2439.

[21] SHI C, CUI W, LIU S, *et al*. RankExplorer: visualization of ranking changes in large time series data [J]. IEEE Transactions on Visualization and Computer Graphics, 2012, 18(12): 2669 - 2678.