基于 EPIC 技术的密码处理器体系结构研究与设计

于学荣,戴紫彬,杨晓辉,马志远
(信息工程大学 电子技术学院,河南 郑州 450004)

摘 要:以分组密码和摘要算法为研究对象,结合处理器体系结构的特点,研究了能够高效灵活实现多种分组密码和摘要算法的处理器体系结构。通过分析现有分组密码算法的运算和结构特点,从实现方式的灵活性和高性能角度出发,提出了一种基于显式并行指令计算结构且性能和灵活性达到了折衷的可编程密码微处理器体系结构。给出了系统的整体架构、可重构运算单元的设计方案及其指令系统的设计,以及用硬件描述语言 Verilog 实现后的硬件测试参数。

关键词:可重构 显示并行结构 对称密码 比特置换

目前,在多数保密通信设备中,主要采用通用 CPU 和专用硬件电路控制密码专用芯片来实现两种方式的密码运算。采用前者控制密码专用芯片时,需要选用一种具有灵活性高、维护容易、升级方便等特点的性能优良的通用微处理器 GPP(General Purpose Processor),但由于通用微处理器指令的局限性,使密码专用芯片达不到其最佳性能,严重影响了保密通信的速度;采用专用硬件电路直接控制密码专用芯片,虽然可使密码专用芯片的性能达到最高,但由于其功能只依赖于密码专用芯片及其外围器件,使得灵活性差、开发周期比较长。

由此可见,无论采用上面哪种方式,由于密码专用芯片的运算处理与控制分离,限制了密码数据处理性能,制约了系统整体速度。针对上述问题,通过分析多种密码算法,本文提出一种基于处理器设计思想的显式并行指令计算结构(EPIC)的可编程密码处理器架构,实现了速度与灵活性的折衷。

1 密码算法分析

1.1 典型的密码算法及其应用

现针对七种分组密码算法和两种杂凑函数即 DES、IDEA、Rijndael、RC6、Serpent、Twofish、Mars、MD5 和 SHA 进行分析。

分组密码算法是一个将比特明文映射成 n 比特密文的双射函数,n 为其分组长度,它的加密与解密过程具有相同的密钥,因此又称为对称密码算法。而杂凑函数是一种将任意长度的消息压缩为某一固定长度的消息摘要的函数,它主要用于数字签名、消息的完整性检测和消息的起源认证检测等方面。

DES 算法(数据加密标准)是第一代公开的完全说明实现细节的被世界公认的分组密码算法[1]。其最初设计者是 IBM 公司,并取得了它的专利权。在随后的二十多年中,DES 算法作为一种典型的分组密码算法,被广泛地应用于保护商业数据的安全(如银行系统等)。

IDEA 算法(国际数据加密算法)公布于 1992 年,是 IPES 标准,因广泛应用于 email 加密认证软件(PGP)中而闻名。

Rijndael 是 1998 年公布的,并于 2000 年在由 NIST(美国国家标准技术研究所)主持的 AES 评选中获胜,此后 Rijndael 算法也称为 AES 算法,成为逐渐代替 DES 的新的加密标准[2]。

RC6、Serpent、Twofish 和 Mars 算法是与 Rijndael 算法一起参评的 AES 候选算法,它们都不同程度地体现了分组密码算法的设计原则,对应用密码学的发展产生了相当大的影响[3-4]。

MD5 消息摘要函数是由 RSA 算法的设计者之一 Rivest 提出的一种单向散列函数,它不基于任何假设和密码体制,采用了直接构造的方法,处理速度非常快。

SHA 是 1993 年公布的联邦信息处理标准(FIPS-180)的安全散列标准,由 NIST 提出并于 1995 年推出了其修订版,通称为 SHA-1。

1.2 密码算法中的基本操作

在分析上述算法的基础上,提取出各个算法的核心操作类型,并总结出它们的基本操作分别为以下六类:S 盒操作[5]、比特置换操作[6]、算术运算、逻辑运算、移位操作和有限域乘法运算[7]。其中算术运算包括模加/减和模乘运算,逻辑运算则由‘与’、‘或’、‘非’和‘异或’组成,表 1 详细列出了它们在各种算法中的具体应用,如 DES 算法中主要使用了 S 盒操作、比特置换、异或和移位操作。

表 1 典型密码算法中的基本操作

操作 / 算法	S 盒	比特置换	移位	有限域乘法	组合逻辑	模运算
DES	6~4	64/64;32/48;32/32;56/48	64		xor	
IDEA			128		xor	$2^{16}+1-\otimes$ $2^{16}-\oplus$
Rijndael	8~8	128	2^8		xor	
RC6			32		xor	$2^{32}-\otimes/\oplus$
Serpent	4~4	128/128	32		xor	
Twofish	4~4		32	2^8	xor	$2^{32}-\otimes/\oplus$
Mars	9~32;8~32		32		xor	$2^{32}-\otimes/\oplus$
MD5			32		xor;and;or;not	$2^{32}-\oplus$
SHA			32		xor;and;or;not	$2^{32}-\oplus$

注：1. x-y 表示 x-bit 输入，y-bit 输出；
2. m/n 表示将 m-bit 置换为 n-bit；
3. $2^n-\oplus$ 表示模 2^n 加法运算；
4. $2^n-\otimes$ 表示模 2^n 乘法运算；
5. 移位操作相应列中的 n 表示移位位宽。

2 可编程密码处理器体系结构设计

在典型的可编程密码处理器结构(AFPC)中，EPIC 结构开发的是标量操作之间的随机并发性，并且增加了功能部件的个数。不相关的指令由编译显式地编入到一个超长的机器指令字中，并发射到流水线，在各个功能部件中并发执行，指令级并行度为 4~8。这种结构的硬件控制相对简单，在计算密集型应用时内在并行性很明显，且不需要很多转移预测。在这种结构上运行指令能够达到较高的实际指令级并行度。正是由于以上特点，EPIC 结构在很大程度上符合了密码算法的需求，即计算密集且顺序执行。

可编程密码处理器体系结构的硬件结构如图 1 所示，整个处理器包括三部分：数据通路、控制单元和输入/输出接口电路。

数据通路是处理器的关键部件之一，包含 FU0~FU5 共 6 个可并行执行的功能单元、32 个 32bit 通用寄存器、4×32 个 32bit 密钥寄存器和回写单元。

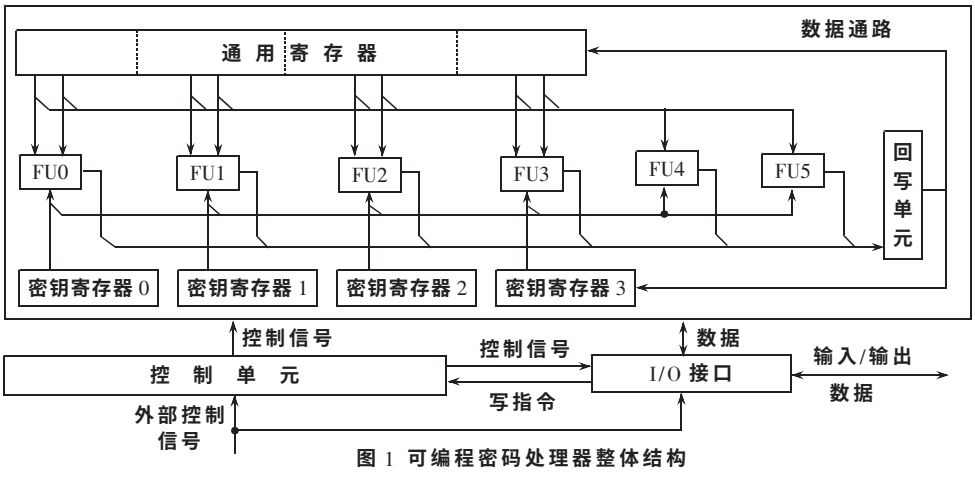

图 1 可编程密码处理器整体结构

功能单元是处理器执行指令运算的核心，由若干个密码运算模块组成。其中，FU0~FU3 内部运算模块的组成与结构完全相同，输入为 3 个 32bit 运算数据，其中 2 个来自通用寄存器堆、1 个来自密钥寄存器堆，输出的运算结果亦为 32bit。FU0~FU3 内部分别设置了 7 个运算模块，分别为 S 盒运算模块、模加/减运算模块、模乘运算模块、32bit 移位运算模块、有限域乘法运算模块、二输入逻辑运算模块、三输入逻辑运算模块。FU4 内部设置了 1 个 128bit 置换运算模块，输入为 12 个 32bit 运算数据，其中 8 个来自通用寄存器堆，4 个来自密钥寄存器堆。FU5 内部设置了 1 个 128bit 移位运算模块，输入也为 12 个 32bit 运算数据，其中 8 个来自通用寄存器堆，4 个来自密钥寄存器堆。

上述这些运算模块功能不是单一的，而是可重构的。表 2 中给出了 4 个可重构运算模块所支持的模式。

除了上述运算模式可重构外，各运算模块根据具体情况还支持运算前增加‘异或’操作、运算后增加‘异或’操作或者运算前后都增加‘异或’操作。由于‘异或’操作延时很小，它的加入并不影响运算的关键路径，这就使得密码运算时减少了单一‘异或’操作的时钟，从而减少了整个运算的时钟数，并且不影响整体性能。表 3 中给出了 Rijndael 算法轮运算流程，采用有限域乘法运算后加入‘异或’操作，时钟周期数由 4 减为 3，10 轮运算将减少 10 个时钟周期。

控制单元完成指令存取、指令译码、指令存储器地址生成等工作，协调处理器内部指令与外部用户命令正确执行。

输入/输出接口电路包括 16 个 32bit 输入寄存器、16 个 32bit 输出寄存器、4 个数据长度计数器、1 个 32bit 命令寄存器等，完成指令、运算数据从 32bit 数据总线装载到指令存储器和输入寄存器以及运算结果从内部通用寄存器写入输出寄存器等操作。

3 指令系统设计

指令系统是算法要素和密码处理器体系结构特性的集中体现，指令系统的设计必须支持硬件的并行执行，即开发指令级并行性(ILP)，指令级并行

表 2 可重构模式举例

功能单元	模式	功能单元	模式	功能单元	类型	功能单元	模式
模加/减运算	模 2^{16} 加法	模乘运算	模 2^{16}	S 盒	8~8 查找表	二输入逻辑运算	与
	模 2^{16} 减法		模 $2^{16}+1$		8~32 查找表		或
	模 2^{32} 加法		模 2^{32}		4~4 查找表		非
	模 2^{32} 减法				6~4 查找表		异或

表 3 Rijndael 算法轮运算流程

Cycle1	Sbox	Sbox	Sbox	Sbox
Cycle2	bit-permutation			
Cycle3	GF	GF	GF	GF
	⊕	⊕	⊕	⊕

注：Sbox 表示 S 盒运算
bit-permutation 表示比特置换(实现行移位)
GF 表示有限域乘法运算
⊕表示‘异或’操作

性的开发程度对发挥密码微处理器的硬件特性，提高程序运行性能至为关键。ILP 技术实际上是指一套完整的处理器设计和编译技术，这些技术通过并行地执行独立的机器操作(如存储器读写、逻辑运算、算术运算等)来加速程序的执行。ILP 的大小可以采用每周期平均执行的指令数(IPC)来衡量，或者采用整个程序的每条指令平均执行的周期数 CPI(CPI=1/IPC)来衡量。在可编程密码处理器体系结构中采用了显式并行指令计算结构，指令级并行数达到 5。

3.1 指令分类

可编程密码处理器体系结构中的指令分为以下几类：

(1) 静态配置指令。它是在密钥生成及加/解密过程中保持不变或者改变次数很少的控制信息配置指令，算法确定后，其 S 盒查找表信息、有限域乘数矩阵和不可约多项式、若干个置换的控制信息等就确定了，它们不会因为操作模式不同而改变。在加/解密过程中采用将配置指令分离出来的方法，可以大大减少执行密码运算时指令的冗余编码，从而缩短了指令字的长度，增加了运算指令字中包含有效操作的个数，有效地提高了加/解密速度并减少了密码程序的代码量。

(2) 短指令。它执行除置换与 128bit 移位运算外的各种密码运算和内部寄存器间的数据传输操作。

(3) 长指令。它执行置换与 128bit 移位运算。

(4) 超长指令。它执行立即数操作和多分支判断操作。

(5) 控制指令。它执行程序跳转、子程序调用及返回、单分支判断等控制操作。

3.2 指令形态

在硬件上，多个功能单元的设置为多条指令的并行执行提供了支持，而哪些指令可以并行执行，哪些指令不能并行执行以及如何将多条指令组装成一条指令的原则，即称为指令的拼装规则。在此设计中有以下几种指令形态：

(1) 静态配置指令。

(2) 超长指令。

(3) 短指令‖短指令‖短指令‖短指令‖控制指令。

(4) 长指令‖控制指令。

其中短指令长度为 37bit，控制指令长度为 32bit，长指令长度为 148bit。无论上述哪种形态，最终的指令字长度都为 192bit（包括指令拼装标识），如四个短指令可以与控制指令拼装成一条指令，长指令也可以与控制指令拼装成一条指令，但静态配置指令与超长指令不能与其他指令拼装，自身组成一个 192bit 的指令字。

4 性能分析

由于可编程密码处理器体系结构支持 5 条指令绑定并行执行，因此其数据路径定义为 5CS (5Combining-Strands)，假设不采用绑定的数据路径定义为 NCS (No-Combining-Strands)，将这两种情况与 Alpha 处理器、CryptoManiac 密码处理器[9]路程进行比较，四种数据路径下加/解密所需时钟数如表 4 所示。分析比较表明可编程密码处理器执行时钟大量减少，尤其与通用处理器 Alpha 相比，加/解时钟数 DES 算法减少 83%，IDEA 算法减少 92%，Rijndael 算法减少 91%，RC6 算法减少 69%，Twofish 算法减少 78%。

表 4 轮运算的加/解密时钟数

	Alpha	CryptoManiac	NCS	5CS
DES	23.56	7	4	4
IDEA	91.95	14	8	7
Rijndael	33.36	9	4	3
RC6	23.24	7	7	7
Twofish	37.36	7	10	8

为了验证可编程密码处理器体系结构实现数据通路和控制通路的正确性，使用 Altera StratixII EP2S180F1508C4 器件作为 FPGA 目标芯片，使用 Altera QuartusII 5.0 工具进行综合，在综合前和综合后使用 Mentor 公司的 ModelSim 5.8c 分别进行功能仿真和时序仿真，结果均正确。其具体资源占用情况如表 5 所示。

密码处理的灵活性和高效性一直是密码算法使用中的制约因素，采用通用微处理器虽然能获得较好的灵活性，但却使一些算法的性能达不到要求；采用专用算

表 5 性能参数列表

器件	Stratix II EP2S180F1508C4	
最高时钟频率	30.53MHz(32.754ns)	
资源	ALUT	30 352
	Memory bit	452 608
	Register	12 986
	DSP block	32

基于 LPS 交换网络的快速比特置换指令系统设计

向 楠[1]，戴紫彬[1]，武清芳[2]
(1. 解放军信息工程大学 电子技术学院，河南 郑州 450004；
2. 解放军 61840 部队，北京 100089)

摘　要：构造了一种基于 LPS 交换网络的比特置换指令系统，可在通用微处理器上进行快速而有效的比特置换。这种指令系统能以不多于 $\log_2 N$ 条的指令完成任意 N 到 N 的比特置换操作，而且非常节省硬件资源。

关键词：比特置换　指令系统　LPS 网络

无论从密码学的角度分析还是从计算机体系结构方面分析，比特置换都具有至关重要的意义。从密码学的角度来说，置换作为扩散的首要手段，在密码算法中得到了广泛应用。例如，在 DES 中有六种不同种类的置换，在 Twofish 中有两种，在 Serpent 中也有两种。比特级的置换提供了字级操作所无法保证的混乱和扩散作用。从计算机体系结构的角度来说，传统的通用微处理器在面向字节处理时性能达到最优，然而在进行短字处理时效率却很低。只有一些指令可以对短字进行处理，最常见的是用逻辑指令(与、或、非等)和移位指令(算术移位和逻辑移位)。因此，为处理器增加面向短字数据操作的新指令越来越迫切。然而，为通用处理器增加新的指令需要考虑到方方面面的因素，如新指令应具有广泛的应用环境，理想的实现复杂度，可接受的资源消耗和良好的性能指标。随着多媒体和信息安全技术的发展，如何提高比特置换的效率将成为面向字节操作的微处理器今后的发展方向[1]。为此，本文综合考虑上述因素，为通用微处理器增加用于有效执行比特置换的指令系统。在

(接上页)

法芯片，在获得很高性能的同时丧失了灵活性。本文针对这一矛盾以 EPIC 结构微处理器构架为出发点，系统地研究了通用的并行分组密码处理器模型、各种密码运算单元、指令集等关键技术，并最终得以实现，达到了实现性能与灵活性之间的良好折衷。

参考文献

[1] MENEZES A J, OORSCHOT P C van, VANSTONE S A. 应用密码学手册[M]. 胡 磊，王 鹏，译. 北京：电子工业出版社，2005:225-234；321-352.

[2] DAEMEN J, RIJMEN V. AES Proposal: Rijndael[S], September 1999. http://www.esat.kuleuven.ac.be/~rijmen/rijndael/.

[3] CHODOWIEC P, GAJ K. Implementation of the twofish cipher using FPGA devices[R]//Technical Report, George Mason University, July 1999. http://www.counterpane.com/twofish.html.

[4] ELBIRT A J, PAAR C. An FPGA implementation and performance evaluation of the serpent block cipher[R]// Eighth ACM International Symposium on Field-Programmable Gate Arrays.Monterey, California:[s.n.],February 10-11, 2000.
http://ece.wpi.edu/Research/crypt/publications/index.html.

[5] FISKIRAN A M, LEE R B. On-chip lookup tables for fast symmetric-key encryption[C]//Proc. IEEE 16th Int. Conf.,Jul.2005:Application-Specific Systems, Architectures, and Processors:ASAP:356-363.

[6] HILEWITZ Y, SHI Z J, LEE R B. Comparing fast implementations of bit permutation instructions[C]// Proceedings of the 38th Annual Asilomar Conference on Signals, Systems, and Computers, Nov. 2004.

[7] HUTTER M, GROBSCHADL J, KAMENDJE G A. A versatile and scalable digit-serial/parallel multiplier architecture for finite fields GF(2^m)[C]// Proceedings of the 4th International Conference on Information Technology: Coding and Computing, 2003:692-700.

[8] WU L, WEAVER C, AUSTIN T. CryptoManiac: A fast flexible architecture for secure communication[J]. the 28th Annual International Symposium on Computer Architecture: ISCA-2001,2001.

(收稿日期：2006-09-25)