

基于 MIPS 处理器的 AES 算法指令集扩展方法与实现

李爱国, 冯国松

(解放军信息工程大学 电子技术学院, 河南 郑州 450004)

摘 要: 由于 MIPS 处理器数据总线宽度的限制, 其扩展的 AES(高等加密标准)指令集无法有效实现其并行性的特点. 为了提高 AES 扩展指令集的并行处理能力, 利用 MIPS 处理器中乘法结果寄存器, 可以一次实现对 64 比特数据的 AES 处理, 有效利用处理器自身资源提高指令集的并行处理能力. 同时, 利用 MIPS 处理器的空闲流水周期可以流水化 AES 中的关键运算, 缩短其关键路径以降低扩展执行单元对流水周期的影响. 对不同实现方式的性能进行比较, 结果表明该方法缩短了 AES 算法中复杂运算的关键路径长度从而使处理器的工作频率不受增加的功能单元的影响, 同时有效地减少了芯片面积, 并且继承了软件编程灵活性的优点.

关键词: MIPS; 高等加密标准(AES); 流水线; 指令集扩展

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2012)06-0126-04

Method and Implementation of Instruction Set Extension for AES on MIPS Processor

LI Ai-guo, FENG Guo-song

(Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004, China)

Abstract: Because of the limit of the data bus of the MIPS Processor, the instruction extension for Advanced Encryption Standard(AES) algorithms on MIPS Processor can't well implement the parallel processing of AES. Taking advantage of the result register of multiplier on MIPS Processor, it can implement 64bit encryption/decryption to increase its parallel processing capacity, this method can't raise the number of the registers on MIPS. And the pipeline processing of the critical byte substitution phase and mix-column phase of the AES algorithms could use the idle cycle of the MIPS's pipeline cycle, it decrease the critical path of the extension module. It compares the performance of different implementation and the results show that this method reduces the critical path to let the processor run at a high frequency, but it decreases a large area of the chip, also it has the advantage of the flexible programming.

Key words: MIPS; Advanced Encryption Standard(AES); pipeline; Instruction Set Extension(ISE)

1 引言

以密码学算法为基础的各种安全协议在信息服务中得到了广泛的发展和应用, 但是密码算法往往需要高强度的复杂运算, 而以软件实现的密码运算会严重消耗嵌入式系统有限的存储资源、计算能力和功耗, 并且严重影响了在高速应用环境中的应用体验; 为了改善在高速环境中的应用体验, 开发密码硬件加速单元来提高主处理器的密码运算能力可以

解决处理器的密码处理瓶颈, 但是硬件加速单元缺乏相应的灵活性并且消耗大量的逻辑资源^[1]; 因此在通用处理器中通过指令集扩展而实现相应的密码运算成为两者折中的方法为许多的研究者所采纳^[2-4]. 专用指令集扩展的方法实现了对密码运算中影响系统性能的关键步骤的硬件支持, 并且使用软件完成算法控制, 因此, 相对于纯软件的方法, 提高了其运算性能, 相对于 AISC 实现方法降低了其硬件复杂度. 本文提出了在 MIPS 处理器上实现 AES

算法的指令集扩展的方法。

2 AES 算法及 MIPS 特点

2.1 AES 算法

高级加密标准(AES)是美国国家标准和技术研究所公布的用以取代 DES 的对称加密算法,该算法具备简洁、高效和安全的特性^[5]。AES 算法的分组长度为 128 bit,可以分别使用 128 bit、192 bit 和 256 bit 的密钥进行加解密运算,根据密钥长度的不同 AES 运算的迭代轮数分别为 10、12 和 14 轮,除最后一轮外其他的轮运算都进行相同的运算,每一轮运算都是以字节或字为单位进行并行操作的。图 1 所示为的轮运算的基本过程,由图中可以看出 128 bit 明文分组按字节进行移位操作之后以字节为单位输入替换单元(替换单元是 AES 算法中唯一的非线性变换),每个字节替换单元都进行相同的操作,然后以字为单位经过列混淆单元(因此只需要四个列混淆单元即可完成一个分组的操作)输出,最后进行轮密钥加(即异或运算)操作,解密过程与加密过程类似,只不过是进行相应的逆运算。这些运算可以并发地执行并且控制相同,因此指令产生的译码信号可以控制多路的功能单元并行执行。

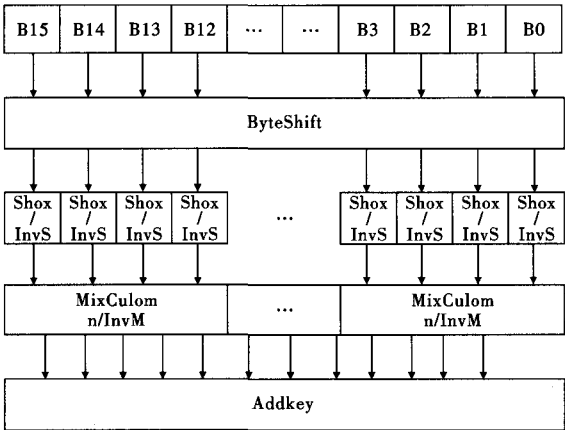


图 1 AES 算法的轮运算

然而,行移位操作是对输入的 16 个字节的重新排列而不仅仅是简单的移位操作,基于软件的实现方式需要花费多条指令完成相应的操作,而这跟机器字长和 AES 的状态矩阵的存储方式有关。AES 的状态矩阵是 16 个字节组成的 4×4 矩阵,移位操作是对矩阵的不同行进行不同的移位操作,而列混淆操作是对状态矩阵的列进行的运算,因此对于 32 位字长的 RISC 处理器来说,如果状态矩阵按列存储的话,则除了行移位外其余的运算指令都仅需 1

个或 2 个操作数,符合一般 RISC 处理器的指令格式,但是行移位需要从每一列取数(相当于四个源操作数),这对于指令编码是困难的。同样,如果状态矩阵按行存储时,列混淆操作需要从四个行(即四个寄存器)进行取数,与按列存储情况类似。本文中的状态矩阵按列存储,同时根据轮运算的特性将字节交换拆分到字节替换和列混淆操作中。

2.2 MIPS 器的指令和流水线

MIPS 处理器^[6]是 RISC(精简指令系统计算机)处理器,本文中使用 MIPS32 位系列处理器,其流水线结构与经典的 RISC 五级流水线稍有不同,其执行级 E 和译码级 D 被合并为一级,对逻辑路径较长的执行级操作被拆分为 2 次操作,分别占据执行级后半周期和访存级的前半个周期,其第四级为对齐级,负责把从外部存储器取回的数据对齐到字的边界,对于普通的非访存指令该级不进行任何操作。

本文的指令集扩展方法利用 MIPS32 位系列处理器流水线的空闲周期即对齐级,从而将扩展的 AES 运算单元中的逻辑路径较长的字节替换运算和列混淆运算进行流水化处理,分别占据 E 级后半周期、M 级和 A 级,这种方法的优点就是通过对字节替换运算以及列混淆运算的流水线处理大大缩小其内部的延时,使扩展的硬件单元的延迟不影响处理器的关键延迟,从而不影响处理器的流水线周期。

3 AES 指令集扩展方法实现

3.1 拆分的行移位运算

前文已经提到根据状态矩阵的存储方式不同行移位操作和列混淆操作的难度是不同的,本文采用按列存储的方式(这样有利于后面的列混淆操作)。由于行移位运算和字节替换运算都是对字节的操作,因此可以将这两步合并起来,如图 2 所示,其基本思想是扩展的字节替换指令从两个源操作数选取其中合适的四个字节组成新的中间状态矩阵列经过字节替换后存储在目的寄存器中,然后再选取合适的中间状态矩阵列作为列混淆指令的源操作数,对源操作数中相应的字节进行交换后再来完成列混淆操作并存储运算结果,在所有字节完成列混淆操作的同时也完成了移位操作。这样做的优点是可以不必扩展专用的行移位指令,同时也满足 MIPS 处理器的对于指令集格式的要求。但是这对于具备并行运算特性的 AES 算法来说浪费了输入的另外的 32 bit 数据和译码产生的控制信号。

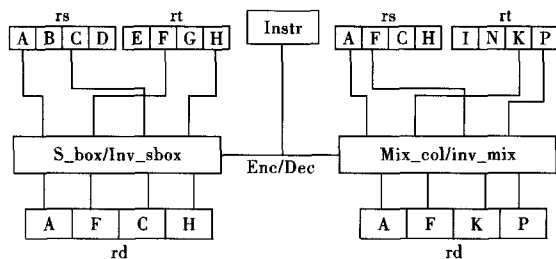


图2 字节交换表示的行移位

3.2 SIMD 技术

单指令流多数据流(SIMD)技术以开发应用中的并行性为手段来提高处理器的运算性能. 由于多路并行的数据通路只需要单一的控制通路, 因此在实现上相对较容易. 由于 MIPS 处理器内部寄存器组的特性, 即使完成 64 bit 数据的处理也无法存储其结果. 文献[5]中实现的 SIMD 技术需要增加存储单元, 本文利用 MIPS 存储器里的特殊寄存器对-乘法结果寄存器对(HI 和 LO), 该寄存器对可以一次性存储 64 bit 的数据, 同时支持寄存器对与通用寄存器之间的双向传输. 因此利用乘法结果寄存器实现对图 2 中运算过程的改进, 如图 3 所示, 我们只将 32 bit 的字节替换和列混淆运算单元扩展为 64 bit 的执行单元, 这样指令就实现了 SIMD 技术, 提高扩展指令的处理能力.

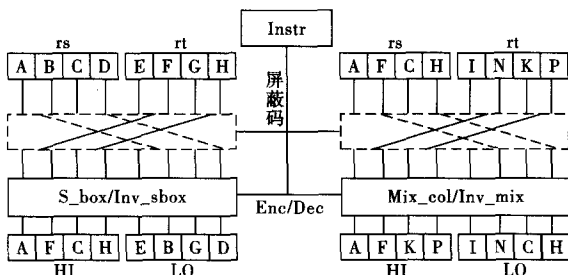


图3 扩展功能单元的 SIMD 架构

前文已经描述了相应的扩展功能单元, 图 4 所示为扩展功能单元的流水线操作, 通过流水化操作可以缩小扩展功能单元的逻辑路径长度, 同时可以有效利用处理器流水线的空闲周期, 提高流水线的利用效率.

3.3 扩展的指令集

在表 1 中列出了具体扩展的 AES 算法专用指令集, 每一条 AES 扩展指令都具有 SIMD 技术特点. AES_sbox_e 和 AES_sbox_d 指令分别完成字节替换操作及其逆运算, 同时完成行移位的第一步运算, AES_mixad_e 和 AES_mixab_e 都是完成列混淆运算, 只不过相应的行移位操作有所不同, AES

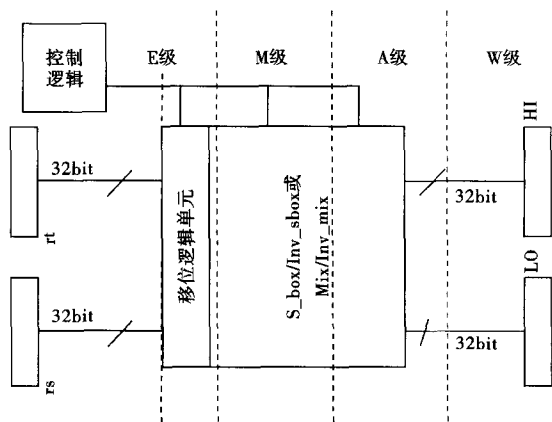


图4 扩展指令集的指令流水结构图

_mixad_d 和 AES_mixab_d 完成相应的列混淆逆运算. 在软件实现算法控制时, 由于指令隐含着等效解密的方法, 因此建议采用等效解密的方法完成解密操作, 可以节省相应的代码空间.

表1 AES 算法扩展指令集

指令名称	功能描述
AES_sbox_e	字节替换操作, 源操作数来自通用寄存器
AES_sbox_d	逆字节替换操作, 源操作数来自通用寄存器
AES_mixad_e	列混淆操作, 源操作数来自通用寄存器并进行字节移位
AES_mixad_d	列混淆逆操作, 源操作数来自通用寄存器并进行字节移位
AES_mixab_e	列混淆操作, 源操作数进行字节移位
AES_mixab_d	列混淆逆操作, 源操作数进行字节移位

4 仿真与分析

AES 算法中字节替换的逻辑资源量和延迟都影响着整体功能电路的面积和延迟. 文献[7]中介绍了 AES 算法中的字节替换和逆字节替换的高效实现方式, 本文采用改进型查表法并如前所述流水化其执行单元, 可以节省一半的存储块.

文献[8]中的硬件加速单元是以轮间的流水和轮内子流水的方式实现 AES 运算, 整个加速单元的吞吐率高达 39.93 Gbit/s, 但是其消耗的逻辑资源也较多, 另外由于使用全流水的方式限制了加密算法的加密模式. 表 2 中给出了本文实现的 AES 算法扩展功能单元、纯软件实现和文献[8]实现方式的性能对比, 纯软件代码使用 MIPS 公司的 SDE 工具编译, 与指令集扩展在相同 CPU 中完成测试, 表中硬件加速单元以及指令集扩展功能单元均使用 Synplicity tool 在 Xilinx 的 FPGA 上完成综合, 由表 2

可以看出 ISE 实现方法的运算能力介于软件方式和硬件加速单元之间,并且 ISE 方式继承了软件实现方式编程灵活性的优点,可以根据不同的密钥长度和加密模式的实际需求来实现系统应用,并不需要再增加额外的硬件单元.从表中可以看出软件实现方式的硬件代价是 0,只是软件的运行需要占用系统大量的资源,而硬件加速单元实现需要大量的逻辑资源,ISE 则介于两者之间,因此 ISE 最大限度的节省了逻辑资源.

表 2 预计算轮密钥和 AES 加密的性能对比

实现方式	密钥扩展	加密		逻辑资源 (slice)
	Cycles	Cycles	Speedup	
pure SW ^[10]	739	1637	1.00	N/A
本文	206	289	5.66	1789+1BRAMs
Asic ^[9]	N/A	1	1637	10563+21BRAMs

5 结束语

本文根据 MIPS 处理器的五级流水线特点,利用 MIPS 处理器的两个特殊寄存器 HI 和 LO 使扩展指令实现 SIMD 技术,另外利用处理器的空闲流水周期流水化组合实现的 AES 算法中的关键运算步骤,通过上述技术以较少的硬件资源实现处理器对 AES 算法的指令集扩展,在提高了软件实现 AES 性能的同时继承了软件实现方式灵活性的优点.如何通过合理的硬件代价使扩展的指令能够支持多种算法是下一步的研究方向,同时如何降低扩展单元的功耗以使其在嵌入式系统中获得更好的应

用也是下一步需要研究的.

参考文献:

[1] 蔡宇东,沈海斌,严晓浪. AES 算法的高速实现[J]. 微电子学与计算机,2004,21(1):83-85.

[2] Elbirt A J. Fast and efficient implementation of AES via instruction set extensions[C]// Proc of AINAW'07. Niagara Falls, Canada:[s. n.],2007.

[3] Tillich S,Groszschaeidl J. Instruction set extension for efficient AES implementation on 32-bit processors[C] //Proc of CHES'06. Yokohama, Japan:[s. n.],2006.

[4] Intel Corp. Intel Advanced Encryption Standard (AES) Instructions Set. [EB/OL]. [2011-09-22]. <http://www.intel.com>.

[5] National Institute of Standards and Technology (NIST). Federal Information Processing Standards Publication 197[S]. 2001.

[6] MIPS32 Architecture for programmers volume III [EB/OL]. (2005-07-01)[2011-03-11]. <http://www.mips.com>.

[7] 高磊,戴冠中. AES 算法中 SubBytes 变换的高速硬件实现[J]. 微电子学与计算机,2006,23(7):48-49.

[8] 何德彪,胡进,陈建华. 基于 FPGA 的高速 AES 实现[J]. 华中科技大学学报:自然科学版,2010,38(2):101-104.

作者简介:

李爱国 男,(1986—),硕士研究生. 研究方向为集成电路设计.

冯国松 男,(1985—),硕士研究生. 研究方向为集成电路设计.

(上接第 125 页)

统网络学习平台的基础上,以建构主义学习理论为指导思想,将 Agent 技术应用于个性化网络教学系统的构建和研究,为加快现代远程教育发展提供了一个新的契机.随着人工智能技术和网络技术研究的不断深入,基于 Agent 技术的网络教学研究必将有着广阔的应用前景.

参考文献:

[1] 林君芬,余胜泉. 关于我国网络课程现状与问题的思考 [EB/OL]. [2011-03-06]. <http://www.edu.cn/20020305/3021721.shtml>,2007.

[2] Wooldridge M J,Jennings N R. Intelligent agents: theory and practice[J]. The Knowledge Engineering Review,1995,10(2):115-152.

[3] 李学峰,李彩清,王文杰. 网络故障监测中多智能 Agent 模型的研究与实现[J]. 微电子学与计算机,2007,24(1):59-62.

[4] 建构主义学习理论[EB/OL]. [2011-05-06]. <http://www.lnnu.edu.cn/xdjyxx/xsqy/xsqy-hkk/jiangouzhu-yi.htm>.

[5] 申瑞民,许彦青,张同珍,等. 基于多代理的智能型远程教学环境研究[J]. 计算机工程与应用,2002(4):253-256.

[6] 鲁俊. 基于多 Agent 技术下的智能决策支持系统研究[J]. 微电子学与计算机,2003,20(10):192-194.

作者简介:

徐林 男,(1979—),硕士,讲师. 研究方向为远程网络教育研究.