

SM3 杂凑算法的 ASIC 设计和实现

张 倩, 李树国

(清华大学 微电子所, 北京 100084)

摘 要: 针对国家商用密码 SM3 杂凑算法提出了一种四合一的 ASIC 实现架构. 该架构采用进位保留加法器和循环展开方式, 与单轮结构相比, 时钟周期数减少了 75%, 吞吐率提高了 29.4%. 采用 65 nm 的 SMIC 工艺, 在 125 MHz 的低时钟频率下, 吞吐率达到了 4 Gb/s. 此款 SM3 杂凑算法芯片已经进行了流片, 支持填充和暂停功能.

关键词: SM3; 杂凑算法; ASIC 实现; 填充

中图分类号: TN49

文献标识码: A

文章编号: 1000-7180(2014)09-0143-04

Design and Implementation of SM3 Algorithm in ASIC

ZHANG Qian, LI Shu-guo

(Institute of Microelectronics, Tsinghua University, Beijing 10084, China)

Abstract: Based on the analysis of the National Business password the SM3 hash algorithm, a four-to-one architecture is proposed. The architecture uses carry-save adder and loop unrolling to improve its performance. Compared with single cycle architecture, the four-to-one architecture reduces the number of clock cycles by 75% and improves the throughput by 29.4%. In SMIC's 65 nm process, the throughput of SM3 has achieved a high performance of 4Gbps at 125MHz. This SM3 hash algorithm chip is being taped. It supports byte padding and has pause function.

Key words: SM3; hash algorithm; ASIC implementation; Padding

1 引言

国家密码管理局于 2010 年 12 月 22 日发布了使用于商用密码应用的 SM3 密码杂凑算法. 它在消息认证、完整性保护、数字签名中有重要的应用^[1], 并且可以用作入侵检测、病毒检测或者构建随机函数或用作伪随机数发生器^[2]. SM3 函数是一种 HASH 函数, 现有 HASH 函数的实现方法已有相关论述^[4-5]. 文献[6]中总结了 HASH 函数的四种硬件实现方式, 并用两种迭代方式对 SM3 算法进行了实现, 完成一次 SM3 压缩过程分别需要 131 和 64 个周期. 文献[7]中较详细给出了 SM3 设计的整体框架, 从消息扩展模块到压缩模块, 用 69 轮迭代实现. 文献[8]中给出了包括控制电路在内的整体结构, 用 64 轮迭代实现. 但是, 在文献 [6-8] 中, SM3

杂凑函数都是用 FPGA 实现的, 并且硬件结构都是不完整, 它们的填充都是用软件实现的.

本文为了实现高性能的 SM3 算法, 提出一种四合一架构, 该架构在循环展开的基础上, 针对关键路径和面积进行了逐一的改进化并且给出从填充到控制电路, 消息扩展模块到压缩模块的完整实现.

2 SM3 杂凑算法概述

SM3 算法的输入是一个小于 2^{64} 比特长度的任意二进制序列, 输出是一个 256 比特的杂凑值, 图 1 中给出了 SM3 算法的具体处理过程. 首先输入一个小于 2^{64} 比特的消息, 经过填充后, 依次分成数据块 $B_1, B_2, \dots, B_n$, 每个数据块 B_i 含有 512 比特的消息. 之后, 每个数据块 B_i 都要经过 64 轮的迭代压缩, 最后与上一个数据块 B_{i-1} 的结果相加, 当然与第一个

收稿日期: 2013-11-17; 修回日期: 2014-01-10

基金项目: 国家“八六三”计划(2012AA012402); 国家自然科学基金(61073173); 清华大学自主研发计划(2011Z05116)

数据 B_i 相加的是初始值 IV. 当最后一个数据块操作结束之后, 得出的就是 256 比特的杂凑值.

SM3 算法的迭代压缩有 64 轮, 在图 2 中给出了具体的描述, 随着轮数的不同, 布尔函数 FF 和 GG 取两种不同的表达式, T 取两个不同的常量. 输入数据 W_j 则每一轮都不同, P_0, P_1 是两个置换函数, 每一轮都相同. rotl_n 表示循环左移 n 位.

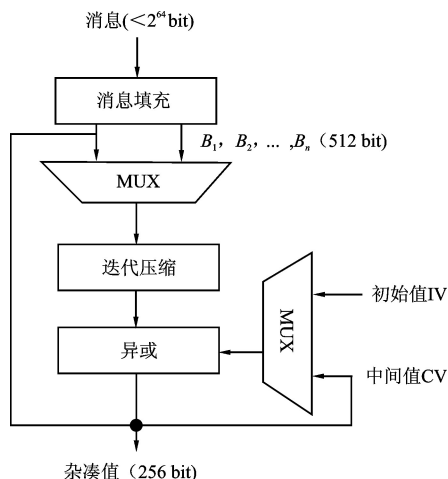


图 1 SM3 算法框图

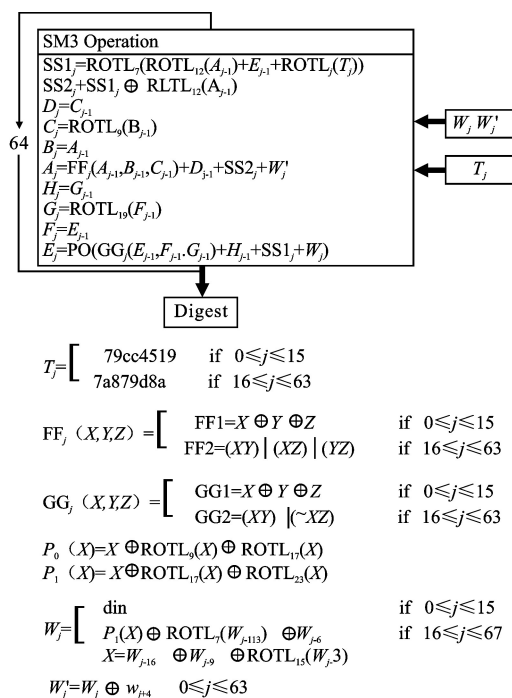


图 2 SM3 压缩过程

3 SM3 杂凑算法 ASIC 实现

如图 2 所示, SM3 算法的压缩过程是该算法的核心部分. 它的迭代压缩有 64 轮, 如果 1 轮迭代用一个周期实现, 实现迭代要用 64 个周期; 如果 2 轮

压缩用一个周期实现, 就需要 32 个周期; 依次类推, 可以有多种实现方案. 经过面积和性能的折中, 本文选取 4 轮用一个周期实现的方案, 下文简称四合一架构.

3.1 四合一架构

为了在低频下提高吞吐率, 采用四合一的化简方式. 在现有的 SM3 论文中, 对于压缩部分的化简都是基于进位保留加法器 (CSA), 本文在采用 CSA 的基础上, 利用四合一的特点, 采用预计算的方法, 改变关键路径, 从而缩短延时.

SM3 算法经过压缩后, 每一个周期执行四次 A-H 的计算, 针对四次 A-H 的运算, 分别进行了压缩, 采用了两种结构, 第一次压缩采用了结构 A, 第二、三、四次压缩采用结构 B. 因此每个周期的关键路径为: $A+3B$.

图 3 是结构 A 及其关键路径图. 为了不增加寄存器, 在这一轮中没有采用预计算. 此时的关键路径用黑粗虚线表示出来, 即为 GG, MUX, CSA, +, +, P_0 . 其中 GG 函数通过 flag 来确定表达式取 G_1 还是 G_2 . flag 即是图 2 中描述的 if 条件的两种情况. 图中 CSA 为进位保留加法器, 它的结构如图 4 所示. S 和 C 分别为三输入加法器中的和与进位值.

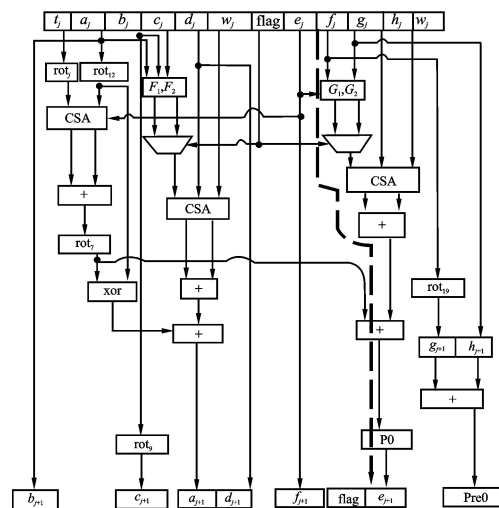


图 3 结构 A 及其关键路径图

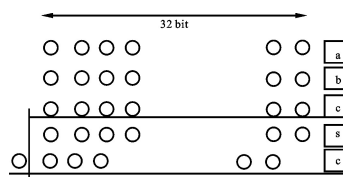


图 4 CSA 原理图

为了进一步缩短关键路径, 在第二次计算 A-H 时, 预先计算 $H+W$, 即图 3 中的 Pre0, 在图 4 中用

到,由图4可以看出,通过预计算,使两条主要路径(A和E的计算过程)长度最接近,这时关键路径也发生了转变.图5中粗线表示出来的第二轮的关键路径,此时的关键路径为:rot_{j+1},rot7,CSA,+,+,P₀.循环左移函数rot_{l_n},在硬件中使用连线实现,其时间与其他逻辑相比可以忽略.因此可以看出结构B比结构A少了GG布尔函数的计算过程.在实际中与之前论文中的关键路径相比,减少了一个CSA和MUX的计算过程.

这样在一个周期中实现A+3B的结构,关键路径为GG,MUX,4CSA,8+,4P₀.

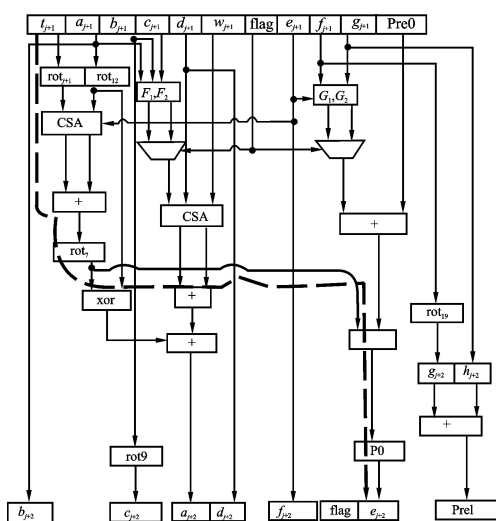


图5 结构B及其关键路径图

3.2 扩展模块

图2的压缩过程中显示,每一轮都需要一个 W_j 和 W'_j 的输入,二者的计算公式也在图2中给出.对于扩展模块,要结合16个周期的时序,在每次四合一结构运算前,所需要的四个 $W_j, W_{j+1}, W_{j+2}, W_{j+3}$ 和 $W'_j, W'_{j+1}, W'_{j+2}, W'_{j+3}$ 要计算出来.在第一个512比特数据输入(对应十六个周期)时,只存入寄存器 $W_0 \sim W_{15}$,不处理数据;在第二个512比特数据输入的过程中,在将这些数据存入寄存器 $W_0 \sim W_{15}$ 的过程中,同时计算第一个数据块的 $W_{16} \sim W_{67}$,用于计算第一个数据块的压缩运算.在计算 $W_{16} \sim W_{67}$ 的过程中,可以重复利用寄存器 $W_{16} \sim W_{31}$,用已有的寄存器存放新的值,每个周期顺次移动四次,如图6所示.方框表示寄存器,nw32~nw35表示利用 $W_{16} \sim W_{31}$ 计算出来的新值.

3.3 填充模块

填充模块完成SM3的消息填充部分,可以用两个状态机实现.一个是计数器的状态机用来计算输入的消息长度,这个计数器分成四部分:有3比特的

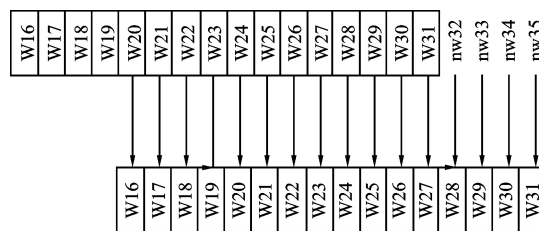


图6 寄存器的复用

0表示消息最小的度量单位是字节;还有2位表示32比特消息中的哪些字节有效,由输入决定;4位表示32比特消息的输入;每次这4位的计数器满16个后,一个55位的计数器加1.总共64比特的计数器,最终的值会添加到消息的末尾表示输入消息的长度.

另一个状态机输出的是填充后的消息,传递给运算模块.设计的关键是利用流处理,接口每输入一个消息,在下一个周期就输出填充后的32比特数据作为数据通路的输入.根据输出不同可以分成不同的状态,这时,只需要一个32位的寄存器就可以传递填充后的数据,减少了芯片的面积.

同时在四合一的架构上,填充模块的处理变得容易,由于输入一个消息块(512比特)需要16个周期,处理一个消息块也是16个周期,因此,芯片的处理速度跟接口同步,减少了一个和填充部分握手的信号.

3.4 控制模块

由于SM3杂凑函数主要以运算处理数据为主,因此控制模块较简单.如图7所示,控制模块产生两个信号来控制运算模块,一个是continue信号,控制SM3运算模块中的状态继续运算或者暂停;另一个是last信号,控制SM3运算模块状态的结束,也标志着即将输出最终的杂凑值.由这两个信号控制的SM3运算模块,按照迭代压缩的流程,包含32个状态.前16个状态用来预输入 $W_0 \sim W_{15}$,后面的16个状态是循环执行的模块,在这16个状态中,除了四合一的运算外,要同时预输入 $W_0 \sim W_{15}$,也需要进行扩展运算,为下个周期的W输入做准备.

3.5 SM3 算法实现结构

图7中给出了SM3杂凑密码芯片的实现框图,从中可以看出这款芯片包含了运算模块、填充模块、扩展模块和控制模块.参考文献[8]中SHA256的接口设计将SM3算法模块的输入数据宽度设定为32比特,输出设定为256比特.由于输入的序列被划分成512比特的数据块,因此,该芯片需要分16个周期来接收数据.SM3算法的IP核的引脚图如

图 7 所示.

SM3 压缩的消息,通过数据接口 din 来传输. 在同一个周期,与 din 相匹配输入的信号有 load data 信号, last data 信号和 last byte 信号. 其中,load data 信号表示 din 值是否有效;last data 表示输入的 din 值是否为当前正在压缩的消息的最后一个有效输入;last byte 表示输入的最后一个有效值中有效的字节数. 当一条消息计算结束后,杂凑函数值由 digest 输出同时 finish 信号有效.

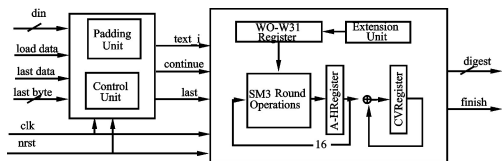


图 7 SM3 实现框图

4 功能验证结果及性能分析

4.1 RTL 级仿真

在该芯片的设计中,采取两种方式进行测试,一种是自主设计特殊向量,可以测试状态机中的特殊跳转. 特殊向量指输入 1 个字节到 68 个字节之间的 68 种情况,主要针对填充模块中的状态机,保证这些状态机都可以正常工作.

在特殊向量测试达标之后,要进行大量测试,尽可能遍历所有情况,确保芯片功能的正确性. miracle 库中的 strong_bigrand 函数,它在大数范围内产生随机数,其随机性比 C 语言中的 random 函数好,利用它来产生大量的随机消息. 把随机产生的消息作为 SM3 算法的 C 语言程序的输入,产生大量的测试结果. 把这些正确的结果与采用 Verilog 语言实现的 SM3 算法结果进行比对,若错误,则修改硬件程序;若正确,则进行下一组. 最终测试了百万组向量,全部正确.

4.2 性能分析

表 1 中给出了 SM3 实现的性能指标. 四合一架构下,SM3 算法的吞吐率在频率为 125 MHz 的情况下,可以达到了 4 Gb/s. 现有论文中,最快的 SM3 算法^[8]是用 Virtex-5 实现的,在 197.2 MHz 的时钟频率下其吞吐率是 1.483 Gb/s. 相比文献^[8],本设计性能提升了 169.7%. 为了在相同工艺下比较四合一架构的优势,本文采用单轮结构设计了 SM3 算法,关键路径化简采用结构 A,用 69 个周期实现,与这个单轮结构相比,四合一架构的性能提升了 $(4 - 3.09)/3.09 = 29.4\%$. 四合一架构用 16 个周期实现一次循环,相比单轮结构周期数减少了约 3/4.

为了对比四合一架构在低时钟频率下的高性能,在不同的接口时钟频率下给出了单轮结构和四合一结构的吞吐率对比. 可以看出,在低于 125 MHz 的时钟频率下,四合一结构实现的 SM3 性能都是单轮结构性能的四倍以上. 因此,本文的设计可以达到低频下的高性能,如表 2 所示.

表 1 SM3 三种实现方式性能对比 (SMIC 65 nm)

结构	时钟/ns	频率/MHz	吞吐率/(GB/s)	功耗/mW	面积/ μm^2
迭代 ^[3]	5.1	197.2	1.483	—	328 slice
单轮	2.4	416	3.09	7.218	29 625
四合一	8	125	4.00	2.801	64 483

表 2 低时钟频率下 SM3 两种实现方式性能比较

频率/MHz	40	50	80	100	125
单轮/(Gb/s)	0.297	0.371	0.594	0.743	0.928
四合一/(Gb/s)	1.28	1.6	2.56	3.2	4.00

图 8 是高性能密码芯片 SM234 的最终版图,该芯片中,由于接口的限制,SM3 的吞吐率为 3.2 Gb/s,但是内核的吞吐率最高可以达到 4 Gb/s. 该款芯片已送出流片.

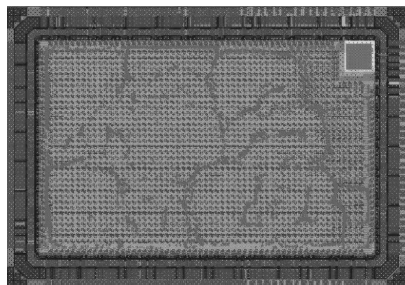


图 8 高性能密码芯片 SM234 最终版图

5 结束语

用 ASIC 实现的本款芯片,支持最小输入单位为字节的填充,支持随时暂停功能,可以处理消息长度小于 2^{64} 比特的任意输入. SM3 杂凑算法的四合一架构,在低频下性能提升的优势非常明显. 由于芯片工作频率低,可以直接与外部接口进行通信,不需要锁相环来提高时钟频率就可以达到需要的性能,适应高性能 SM2 算法的数字签名工作.

参考文献:

- [1] 国家商用密码第 22 号文件[EB/OL]. [2013-11-01]. <http://www.oscca.gov.cn/file/201012221418577862010>.
- [2] William Stallings. 密码编码学与网络安全——原理与实践[M]. 5 版. 北京:电子工业出版社, 2012.

(下转第 152 页)

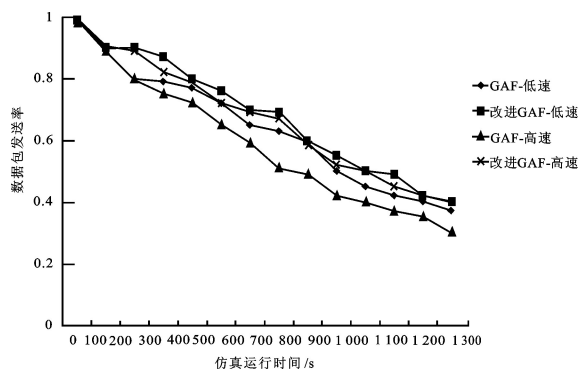


图7 数据包发送率对比

的次数被改进 GAF 算法要多. 为此, 改进 GAF 算法的平均延迟优于 GAF.

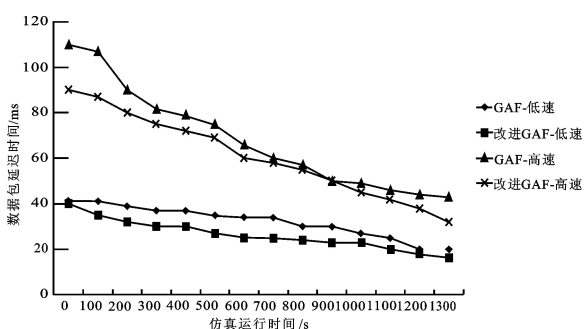


图8 平均数据传输延迟对比

5 结束语

本文提出一种改进的 GAF 算法, 仿真实验表明本文提出的改进 GAF 算法的网络生存时间、数据包平均传达率、平均的数据传输延迟比 GAF 协议要高.

参考文献:

- [1] G J Pottie, W J Kaister. Wireless integrated network sensors[J]. Communications of the ACM, 2000, 43(5):51-58.
- [2] Akyildiz I F, Weilian S, Sankarasubramanian Y, et al. A Survey on Sensor Networks[J]. IEEE Communication Magazine, 2002, 48(8):1-14.
- [3] Heinzelman WR, Chandrakasan A, Balakrishnan H. Energy-Efficient communication protocols for wireless microsensor network[C]// Proc of the Hawaaiian Int'l Conf on Systems Science. Hawaii:IEEE, 2000.
- [4] Bao L, Garcia-Luna-Aceves JJ. Topology management in ad hoc networks[C]//Proc of the MobiHoc. [s. l.]: IEEE, 2003.
- [5] Y Xu, J Heidemann, D Estrin. Geography-in form ed energy conservation for Ad Hoc Routing[C]// Proceedings of the 7th Annual Internati onal Conference on Mobile Computing and Networking. [s. l.]: IEEE, 2001.
- [6] 刘曙, 刘林华, 陶军. 一种基于蜂窝结构的改进 GAF 算法[J]. 计算机技术与发展, 2009, 39(1):39-43.
- [7] P Baul, V N Padmanabhan. User Location and Tracking in An In-building Radio Network[R]. Technical Report MSR-TR-99-12, Microsoft Research, Feb. 1999.

作者简介:

孔德川 男, (1977-), 硕士, 讲师. 研究方向无线传感器网络、嵌入式系统.

(上接第 146 页)

- [3] 郭文平, 刘政林, 陈毅成, 等. 高吞吐率、低能耗的 SHA₁ 加密算法的硬件实现[J]. 微电子学与计算机, 2008, 25(5):76-79.
- [4] Ling Bai, Shuoguo Li. VLSI implementation of high-speed SHA-256[C]// IEEE 8th International Conference on ASIC. Changsha, Hunan: IEEE, 2009: 131-134.
- [5] Eun-Gu Jung, Daewan Han. Low area and high speed SHA-1 implementation[C]// International Soc Design Conference. Jeju: IEEE, 2011:365-367.
- [6] 丁冬平, 高献伟. SM3 算法的 FPGA 设计与实现[J]. 微型机与应用, 2012, 31(5):26-27.
- [7] 王晓燕, 杨先文. 基于 FPGA 的 SM3 算法优化设计与实现[J]. 计算机工程, 2011, 37(19):1-4.

- [8] 刘宗斌, 马原, 荆继武, 等. SM3 哈希算法的硬件实现与研究[J]. 信息安全, 2011(9):191-193.

作者简介:



张倩 女, (1988-), 硕士研究生. 研究方向为杂凑密码芯片设计.

李树国 男, (1963-), 博士生导师. 研究方向为信息安全算法及其大规模集成电路设计与实现.