

SHA-2(256,384,512)系列算法的硬件实现

刘政林, 董 馨, 李东方

(华中科技大学 电子科学与技术系, 湖北 武汉 430074)

摘 要: 在同一系统中存在着对安全性要求不同的应用, 可能需要对 SHA-256、SHA-384、SHA-512 算法进行选择, 目前大部分研究只是对这几种算法单独地进行了硬件实现. 本文提出了一种 SHA-2(256,384,512)系列算法的 VLSI 结构, 基于这种结构, 根据不同的要求, 每一种 SHA-2 算法都可以单独灵活地执行. 本文还对该系列算法和各个独立 SHA-2 算法的 FPGA 实现进行了比较, 结果表明, 在面积较 SHA-256 实现增加 40%, 而与 SHA-384/512 基本相同的情况下, 频率可达到 74MHz.

关键词: 哈希函数; 安全性; 密码学; SHA-2(256 384 512); 硬件实现

中图分类号: TP309.7

文献标识码: A

文章编号: 1000-7180(2012)12-0051-04

On the Hardware Implementations of the SHA-2(256,384,512) Hash Function

LIU Zheng-lin, DONG Xin, LI Dong-fang

(Department of Electronics Science and Technology, Huazhong University
of Science and Technology, Wuhan 430074, China)

Abstract: In a system, applications with different security requirements need selection among algorithm SHA-256, SHA-384, SHA-512. However, these types of algorithms were implemented in hardware separately in current studies. In this paper, a VLSI architecture for the SHA-2 family is proposed, in order to meet the needs of different options. The proposed architecture supports a multi-mode operation in the sense that it performs all the three hash functions (256, 384, 512) of the SHA-2 standard. The proposed system is compared with the implementation of each hash function in FPGA device. The results show that the frequency of the introduced system can be achieved 74MHz, while the area of the implement only increase 40% over SHA-256 and basically the same as SHA-384/512.

Key words: Hash functions; security; cryptography; SHA-2(256,384,512); hardware implementation

1 引言

安全哈希标准(Secure Hash Standard, SHA)是由美国国家标准技术研究所(NIST)和美国国家安全局(NSA)设计的杂凑算法.

随着对安全性要求的提高, SHA-2 系列算法的硬件实现成为了研究的热点, 已经提出了诸多的硬件实现方案^[1-7]. 但是目前所提出的大多数实现方案都是针对单个算法的实现, 不能满足对安全性

可选择需要, 能够实现 SHA-2(256,384,512)系列算法的硬件结构是我们研究的目标. 在许多应用场合, 速度是一个很重要的因素. 另外, 随着市场对器件体积和便携式的要求, 体积也成考虑的重要条件之一. 本文兼顾了这两种要求.

2 SHA-2 系列算法简介

SHA-2 系列算法包括 SHA-224、SHA-256、SHA-384、SHA-512. 这些算法的主要不同

收稿日期: 2012-04-18; 修回日期: 2012-05-26

基金项目: 国家自然科学基金项目(60973034); 新世纪优秀人才支持计划项目(NCET-07-0328)

在于操作数长度、初始向量值、常量值和最后产生信息摘要的长度. 表 1 列出了 SHA-1 和 SHA-2 系列算法的主要区别.

表 1 哈希算法特性对比

SHA 标准	攻击复杂度 (bits)	信息块大小 (bits)	循环次数	字长 (bits)	哈希值 (bits)
SHA-1	2^{128}	$<2^{64}$	80	32	160
SHA-224	2^{128}	$<2^{64}$	64	32	224
SHA-256	2^{128}	$<2^{64}$	64	32	256
SHA-384	2^{192}	$<2^{128}$	80	64	384
SHA-512	2^{256}	$<2^{128}$	80	64	512

SHA-2 系列算法的主要计算步骤基本相同, 不同在于功能函数的定义和循环次数. SHA-2 系列算法主要步骤:

(1) 预处理

预处理过程包括三步:

① 信息填充. 根据 FIPS 180-2 中的方法, 对于 SHA-256 算法, 将输入数据填充为 512 bit 的整数倍, 而对于 SHA-384 和 SHA-512, 将输入数据填充为 1024 bit 的整数倍.

② 分割填充的信息. 将填充后的信息分解成 m 个 512 bit (SHA-256) 或者 1024 bit (SHA-384/512) 的信息块, 用于哈希计算.

③ 设置初始向量值 $H^{(0)}$.

(2) 哈希循环计算

计算 SHA-256 时, $W=32$ bit, 循环次数 $t_{\max}=63$; 而计算 SHA-384 和 SHA-512, $W=64$ bit, 循环次数 $t_{\max}=79$. 哈希循环计算主要包括四个步骤:

经过预处理分解的信息块依次进行计算.

① 准备信息表, $\{W_t\}$ 值:

$$W_t = \begin{cases} M_t^{(i)}, & 0 \leq t \leq 15 \\ \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}, & 16 \leq t \leq t_{\max} \end{cases}$$

其中 $M_t^{(i)}$ 为信息块分解成 16 个 32 bit (SHA-256) 或者 64 bit (SHA-384 512) 初始值.

② 设置每一次哈希循环的 8 个工作常量: a, b, c, d, e, f, g, h .

$$a || b || c || d || e || f || g || h = H^{(i-1)}$$

③ $t=0$ 到 $t_{\max}$, 执行循环:

{

$$T_1 = h + \sum_1(e) + Ch(e, f, g) + K_t + W_t$$

$$T_2 = \sum_0(a) + Maj(a, b, c)$$

$$g = f \quad f = e \quad e = d + T_1$$

$$d = c \quad c = b \quad b = a$$

$$a = T_1 + T_2$$

}

④ 计算第 i 次的最终哈希值 $H^{(i)}$

$$H_0^{(i)} = a + H_0^{(i-1)}$$

$$H_1^{(i)} = b + H_1^{(i-1)}$$

$$H_2^{(i)} = c + H_2^{(i-1)}$$

$$H_3^{(i)} = d + H_3^{(i-1)}$$

$$H_4^{(i)} = e + H_4^{(i-1)}$$

$$H_5^{(i)} = f + H_5^{(i-1)}$$

$$H_6^{(i)} = g + H_6^{(i-1)}$$

$$H_7^{(i)} = h + H_7^{(i-1)}$$

然后重复以上步骤 N 次, 得到最终的信息摘要 M :

$$H_0^{(N)} || H_1^{(N)} || \dots || H_7^{(N)}$$

3 SHA-2 的硬件结构

为了实现 SHA-2 系列的三种算法, 本文提出了一种通用的硬件结构, 如图 1 所示. SHA-256、SHA-384、SHA-512 计算过程中的主要不同在于循环次数和功能函数. 在提出的硬件结构中, 通过一个配置寄存器来控制循环次数和对功能函数进行选择. 当配置寄存器设置为计算 SHA-256 时, 循环次数为 64, 选择定义 SHA-256 的功能函数进行中间值计算; 而当配置寄存器设置为 SHA-384 或 SHA-512 时, 循环次数则为 80, 选择 SHA-384 和 SHA-512 的功能函数. 此外, 在此硬件结构中, 为了满足 SHA-2(256, 384, 512) 的各种算法, 其进行操作的输入和输出数据都为 64 bit.

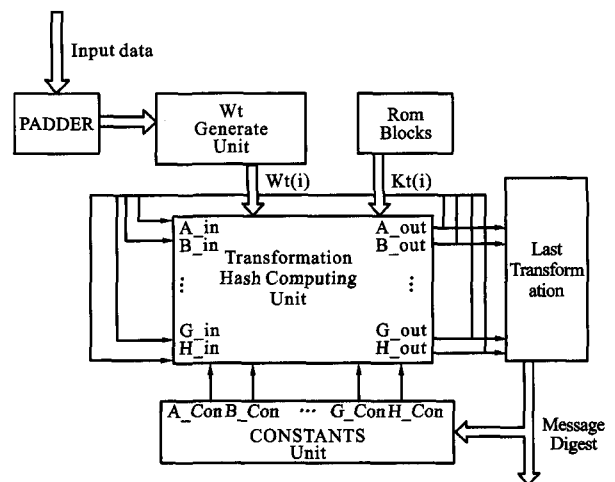


图 1 SHA-2 的硬件结构

3.1 WT 模块

Wt 模块的结构如图 2 所示. Wt 模块的输入为经过填充的信息. 在 Wt 模块中,经过填充的信息被分成 N 个相等的数据块: $M^1, \dots, M^N$, 它们依次传入 Wt 模块中进行计算. 经过分组的每个数据块 M^i , 分成 16 个数据, 依次存储在 $W_0, \dots, W_{14}$ 和 Wt 中. Wt 模块的输出依据以下两个等式进行进算:

$$W_t = \begin{cases} M_t^{(i)}, & 0 \leq t \leq 15 \\ \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}, & 16 \leq t \leq t_{\max} \end{cases}$$

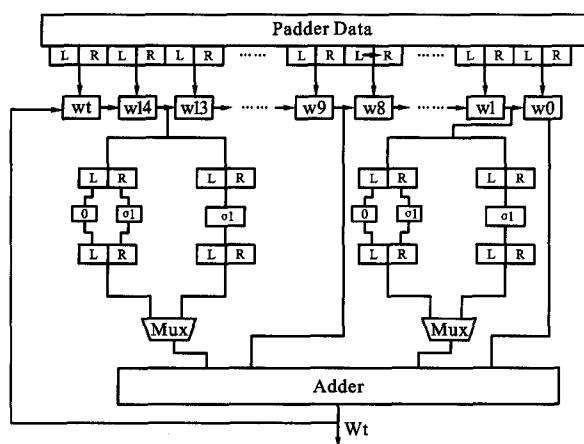


图 2 WT 模块结构

当执行 SHA-256 计算时, Wt 的操作数为 32 bit, 而 SHA-2 结构的输入都为 64 bit, 所以通过配置寄存器控制的多路选择器将原始输入的 16 个数据块左边 32 bit 置为 0; 执行 SHA-384 和 SHA-512 计算时, Wt 的操作数为 64 bit 不需要进行操作. 在计算 Wt 所需的中间值时, 通过多路选择器来选择 SHA-256 和 SHA-384/512 的功能函数.

3.2 ROM 模块

在 ROM 模块中(如图 3 所示)存储了特殊的常量 $K_t(i)$. 对于 SHA-2 而言, 它们有不同的 $K_t(i)$ 值. SHA-384 和 SHA-512 的 $K_t(i)$ 为相同的 80 个 64 bit 的常量, 而 SHA-256 的 $K_t(i)$ 为 64 个 32 bit 的常量, 并且这 64 个常量和 SHA-512 $K_t(i)$ 值的前 64 个的高 32 位相同.

因此, 在此模块中定义了两个 80×32 bit 的 ROM 来存储常量, 这样可以减小资源面积. 当计算 SHA-384、SHA-512 时, ROM1 中的 32 bit 为 K_t 的左边部分, ROM2 中的 32 bit 为 K_t 的右边部分; 而计算 SHA-256 时, ROM1 中的 32 bit 为 K_t 的左边部分, 通过选择器控制右边部分的输出则为 0.

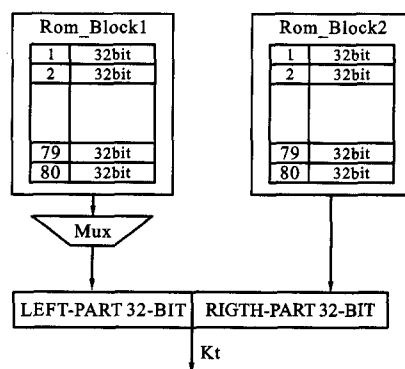


图 3 ROM 模块结构

3.3 哈希循环计算模块

哈希循环计算模块结构如图 4 所示. SHA-2 的循环算法和步骤一样, 不同的只是功能函数定义和循环次数. 哈希计算模块有 8 个基本数据 ($A_{in}, \dots, H_{in}$) 输入, 并且产生了 8 个数据输出 ($A_{out}, \dots, H_{out}$). 8 个常量值 ($A_{con}, \dots, H_{con}$), 在初始化过程中就已经输入, 此模块还有两个输入, $K_t(i)$ 和 $W_t(i)$. $K_t(i)$ 常量由 ROM 模块输出, 而 $W_t(i)$ 则是由 Wt 模块输出. 哈希循环计算模块主要是计算 T1、T2 值, 和最后的模加法. 当进行 SHA-256 计算时, 通过配置寄存器将 8 个初始化常量的高 32 位定义为 0. 在 SHA-256 的整个计算过程中只有右边的 32 bit 进行操作, 所以加法器为 2^{32} 模加法器. 计算 SHA-384, SHA-512 时, 加法器则为 2^{64} 模加法器. 为了使所提出的硬件结构可以正确执行 SHA-2(256, 384, 512) 系列的所有哈希计算, 此模块中每个 64 bit 的数据都被分成了两个 32 bit 的数据, 以便通过配置寄存器对 SHA-256 和 SHA384/512 计算进行控制.

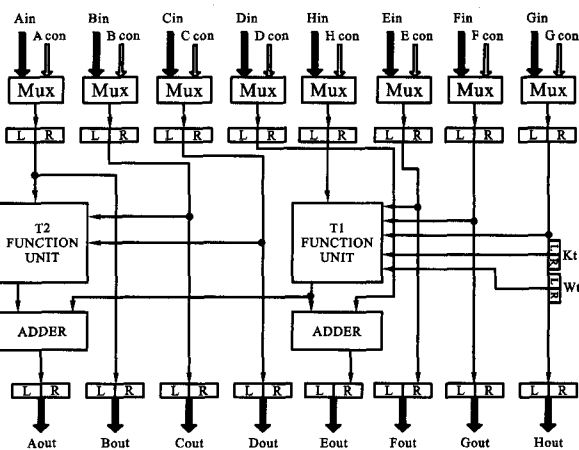


图 4 哈希循环计算模块

经过多次哈希循环计算, 由最后操作模块得到计算结果. 在此模块中, 输入输出都为 64 bit. 只是

SHA-2(256)产生的信息摘要为 8 个 64 bit 计算结果的右边 32 bit ($H_0^{(N)}[31:0] || \dots || H_7^{(N)}[31:0]$); SHA-2(512)产生的信息摘要为 8 个 64 bit 的计算结果($H_0^{(N)} || \dots || H_7^{(N)}$); 而 SHA-2(384)产生的信息摘要为 8 个 64 bit 计算结果的左边 6 个 ($H_0^{(N)} || H_1^{(N)} || H_2^{(N)} || H_3^{(N)} || H_4^{(N)} || H_5^{(N)}$).

4 性能分析与比较

使用 Verilog 硬件描述语言,对本文提出的 SHA-2 结构进行实现设计. 根据 FIPS 180-2 中所给的数据,通过 Modelsim Simulator 验证环境仿真,结果和所给数据相符,功能正确无误. 并且在 ISE 环境编译综合,选择 Xilinx VIRTEX-II 系列器件为目标器件进行整体综合、仿真和底层布局. 最后将所得结果与 SHA-256、SHA-384、SHA-512 和 SHA-2 相比较后,得到如表 2 所示的结果.

表 2 性能比较

Reference Architecture	Area /slice	Frequency /MHz
Our work SHA-2	2458	74
SHA-2 ^[4]	2951	50
SHA-256 ^[4]	1306	77
SHA-384 ^[4]	2581	69
SHA-512 ^[4]	2545	69
SHA-384/512 ^[4]	2573	66

本文将 SHA-256、SHA-384、SHA-512 设计在一个 VLSI 结构上,虽然面积较实现 SHA-256 大 40%,但是却比实现 SHA-512 所需的面积更小,并且可达到 74 MHz 的频率.

5 结束语

本文提出了一种 SHA-2(256,384,512)系列算法的硬件结构. 基于这种结构,每一种 SHA-2(256,384,512)算法都可以单独灵活地执行. 实验结果表明,这种方法在面积较 SHA-256 实现增加 40%,而较 SHA-384/512 基本相同的情况下,可以实现 SHA-256、SHA-384 和 SHA-512 三种算

法,并且最大频率可达到 74 MHz. 能满足安全加解密设备对小面积和高性能的要求,可用于无线网络加密系统、移动数字电视收费点播数字认证系统设计等.

参考文献:

- [1] Sklavos N, Koufopavlou O. On the hardware implementation of the SHA-2 hash functions [C]//proc. of IEEE International symposium on Circuits and systems. UR; Bristol, 2003: 25-28.
- [2] McEvoy R P, Crowe F M, Murphy C C, et al. Optimization of the SHA-2 family of hash functions on FP-GAS[C]//IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures (ISVLSI'06). Germany; Karlsruhe, 2006: 317-322.
- [3] McLoone M, McCanny J. Efficient single-chip implementation of SHA-384 and SHA-512[C]//IEEE Proc., International Conference on Field-Programmable Technology. China; Hongkong, 2002: 311-314.
- [4] Ryan Glabb, Laurent Imbert, Graham Jullien. Multi-mode operator for SHA-2 hash functions[J]. Journal of systems architecture, 2007, 53(2/3): 421-425.
- [5] Dadda L, Macchetti M, Owen J. An ASIC design for a high speed implementation of the hash function SHA-256 (384, 512)[J]. ACM Great Lakes Symposium on VLSI, 2004: 421-425.
- [6] 杨晓辉, 戴紫彬. 基于 FPGA 的 SHA-256 算法实现[J]. 微计算机信息, 2006, 22(4/2): 146-148.
- [7] 郭文平, 刘政林, 陈毅成, 等. 高吞吐率、低能耗的 SHA-1 加密算法的硬件实现[J]. 微电子学与计算机, 2008, 25(5): 76-79.
- [8] William Stallings. 密码编码学与网络安全[M]. 孟庆树, 傅建明, 译. 北京: 电子工业出版社, 2006.
- [9] FIPS PUB 180-2, Secure Hash Standard (SHA-2). National institute of standards and technology (NIST) [S/OL]. August 1, 2002. available from <http://www.itl.nist.gov/fipspubs/>.

作者简介:

刘政林 男, (1968-), 博士, 教授. 研究方向为数字电路设计.