第 36 卷 第 6 期 计 算 机 工 程 2010 年 3 月
Vol.36 No.6 Computer Engineering March 2010
·安全技术· 文章编号：1000—3428(2010)06—0131—02 文献标识码：A 中图分类号：TP309

可重构散列函数密码芯片的设计与实现

李 淼，徐金甫，戴紫彬，杨晓辉
(解放军信息工程大学电子技术学院，郑州 450004)

摘 要：根据不同环境对安全散列算法安全强度的不同要求，采用可重构体系结构的思想和方法，设计一种可重构的散列函数密码芯片。实验结果表明，在 Altera Stratix II 系列现场可编程门阵列上，SHA-1, SHA-224/256, SHA-384/512 的吞吐率分别可达到 727.853 Mb/s, 909.816 Mb/s 和 1.456 Gb/s。

关键词：可重构密码芯片；安全散列算法；现场可编程门阵列

Design and Implementation of Reconfigurable Hash Function Cryptographic Chip

LI Miao, XU Jin-fu, DAI Zi-bin, YANG Xiao-hui
(Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004)

【Abstract】According to different needs to security hash algorithms under different circumstances, this paper adopts the thought and method of the reconfigurable architecture, and designs a reconfigurable hash cryptographic chip. Experimental results based on FPGA of the family of Stratix II of Altera Corporation show that the proposed system reaches throughput values equal to 727.853 Mb/s for SHA-1, 909.816 Mb/s for SHA-224/256, and 1.456 Gb/s for SHA-384/512 respectively.

【Key words】reconfigurable cryptographic chip; security hash algorithms; Field Programmable Gate Array(FPGA)

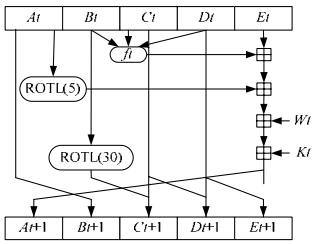
1 概述

1995 年，美国国家标准与技术研究所(NIST)公布了新的安全散列算法 SHA-1，该算法替代了 1993 年颁布的散列函数标准算法 SHA；2001 年，为了满足更高的安全等级，颁布了 3 个新的散列函数 SHA-256, SHA-384 和 SHA-512，散列值长度分别为 256 bit, 384 bit 和 512 bit；2004 年，又增加了 SHA-224。5 种散列函数一起作为安全散列标准[1]。

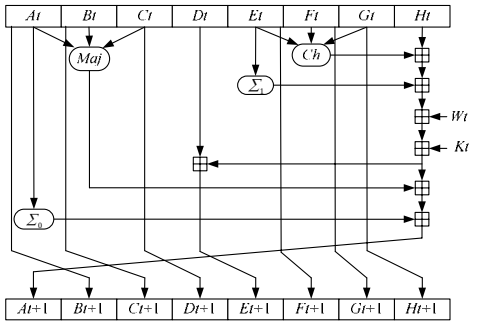
可重构密码芯片是采用可重构体系结构的思想和方法进行设计、用于对数据进行加/解密处理的集成电路芯片。其内部的逻辑电路能够根据不同密码算法的需求，重新组织，构成不同的电路结构，实现不同的功能，从而匹配不同的密码算法，达到灵活、快速地实现多种密码算法的目的[2]。此外，由于可重构体系结构设计建立在某些硬件资源能够被不同应用需求重复利用基础上，因此其消耗的硬件资源要比所能实现的各个算法专用芯片占用硬件资源的总和要少得多。可重构密码芯片不仅能灵活实现多种密码算法，还能有效利用硬件资源，达到节约逻辑资源的目的。

2 安全散列算法简介

SHA-1/224/256/384/512 是 5 种安全散列算法，用于产生输入消息的一定长度的消息摘要。这些算法能够确保消息的完整性：对于输入的任何微小的改变都会引起输出的很大差异。这种特性被用于数字签名、消息认证和随机数的产生等应用中。5 种算法都包含消息预处理和散列值计算 2 个部分。消息预处理分为消息填充、消息分割和寄存器初始化 3 个部分。散列值计算通过压缩函数，将分割后的消息块压缩成一定长度的摘要值。图 1 为 5 种安全散列算法的轮运算电路结构[3]。



(a)SHA-1 轮运算结构



(b)SHA-224/256/384/512 轮运算结构

图 1 5 种安全散列算法的轮运算电路结构

基金项目：国家“863”计划基金资助项目(2008AA01Z0103)
作者简介：李 淼(1983－)，女，硕士研究生，主研方向：专用集成电路设计；徐金甫，副教授、博士；戴紫彬，教授、博士；杨晓辉，博士研究生
收稿日期：2009-09-20 **E-mail：**limiao830226@yahoo.cn

3 可重构设计

通过对 SHA-1/224/256/384/512 5 种算法的分析可以发现，初始常数 *Kt* 存储模块以及存放散列值的移位寄存器、基本函数、*Wt* 生成电路和数据通路中的 CSA 加法器都是可重用的部件。本文利用 FPGA 可重构计算的特点，对可重用的模块进行可重构设计，以达到对 FPGA 资源灵活有效利用的目的。

对于常数 *Kt* 和初始散列值，只存储 SHA-1 的 4 个 32 bit 初始常数值和 SHA-384/512 的 80 个 64 bit 初始常数值以及 SHA-1 的 5 个 32 bit 初始散列值、SHA-384/512 的 16 个 64 bit 初始散列值，就可以实现初始常数 *Kt* 和初始散列值的复用存储，大大节省了存储空间。

5 种算法的基本函数 $Ch(x,y,z)$ 和 $Maj(x,y,z)$ 结构相同，只是参数的位长不同。SHA-1 其余 2 个基本函数相同，可归结为 1 个函数 $Parity(x,y,z)$。SHA-224/256/384/512 其余 4 个基本函数 $\Sigma_0^{(256)}(x)$，$\Sigma_1^{(256)}(x)$，$\sigma_0^{(256)}(x)$ 和 $\sigma_1^{(256)}(x)$ 结构相同，只是参数的位长和具体的移位位数不同。通过算法选择信号“Sel”可以使不同算法选择对应的函数、参数位长和移位位数，实现对基本函数的可重构设计。

本文将详细介绍 *Wt* 生成电路和数据通路中 CSA 加法器的可重构设计。

3.1 *Wt* 生成电路

Wt 的前 16 个字直接取自当前分组中的 16 个字，即 $Wt = Mt$，其余 *Wt* 的值由不同算法来决定。对 SHA-1 算法，$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) <<< 1$；对 SHA-224/256/384/512 算法，$Wt = \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}$。SHA-1/384/512 算法生成 80 个 *Wt*，而 SHA-224/256 算法生成 64 个 *Wt*。SHA-1/224/256 算法的 *Wt* 为 32 bit，SHA-384/512 的 *Wt* 为 64 bit。

Wt 生成电路由移位寄存器、数据选择器、模 $2^{32}/2^{64}$ 加、异或电路、循环移位电路和 Wt 寄存器构成[4]。通过数据选择器实现了 16 个 64 bit 寄存器的重构设计。前 16 步，外部数据 *Mt* 经由数据选择器送入 16 级移位寄存器和 *Wt* 寄存器；从第 16 步以后，移位寄存器的外部输入是之前寄存器值的函数运算，并同时送入 *Wt* 锁存。而后 *Wt* 寄存器输出数据送入运算模块，直接参与每一步运算。

进一步对算法中模加运算单元进行优化。采用二级 CSA(进位保留加法器)、一级 CLA(超前进位加法器)的级联结构，将原来 9 级 CSA 的延迟(3 个 CLA 级联，一级 CLA 的延时相当于三级 CSA 的延时)缩短到 5 级，提高了 *Wt* 生成电路的运算速度。图 2 为采用 CSA 的 *Wt* 生成电路结构。

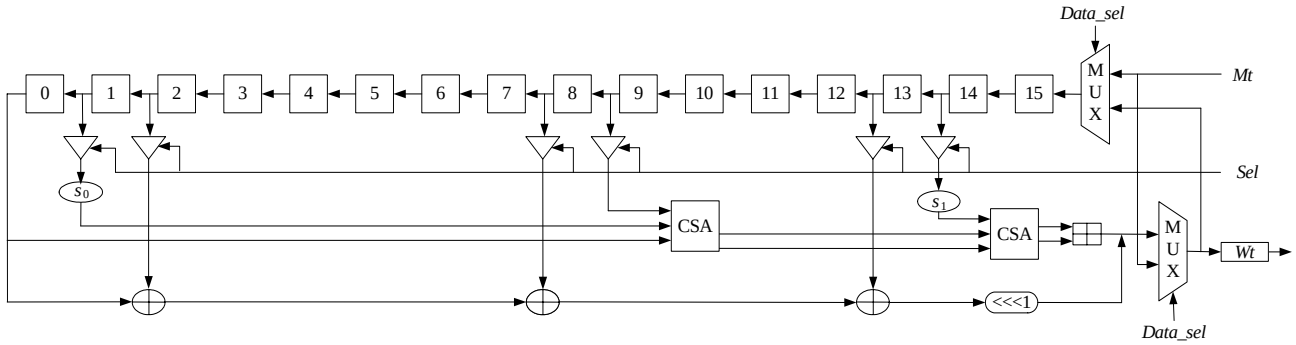


图 2 采用 CSA 的 *Wt* 生成电路结构

3.2 数据路径

本文采用 8 个 64 bit 位宽的移位寄存器 ABCDEFGH 来实现对不同算法逻辑单元的重构设计。对 SHA-1，使用寄存器 *ABCDE* 的低 32 bit，高 32 bit 置 0；对 SHA-224/256，使用 8 个寄存器的低 32 bit，高 32 bit 置 0；对 SHA-384/512，使用 8 个寄存器的全部 64 bit。当复位信号 RESET 有效时，寄存器将根据不同算法进行初始化。

数据路径设计的关键是计算每步寄存器 A 的值，主要包括非线性函数运算、加法运算和移位运算。其中，非线性函数运算只是完成信号在不同输入输出之间的切换，只需用组合逻辑电路设计，不会产生太大的延迟；移位只占用布线资源，同样不会对电路的速度有影响；而由于加法运算的进位会产生延迟，因此应尽量对其进行优化，否则会影响电路的运算速度。

SHA-1 执行 5 个连续 32 bit 加法，SHA-224/256 执行 7 个连续 32 bit 加法，SHA-384/512 执行 7 个连续 64 bit 加法，而且 5 种算法使用的逻辑函数和输入寄存器的值不同，这就需要将各个不同的函数变换的值提前计算出来，再根据算法选择进入 CSA 加法器的输入端。

本文设计的连加运算电路采用 CSA 和一级 CLA 的级联结构，同时还完成了 Et+1 的计算，实现了 CSA 对不同算法、不同寄存器的重用，有效地节省了资源。采用 CSA 的连加运算电路结构如图 3 所示。

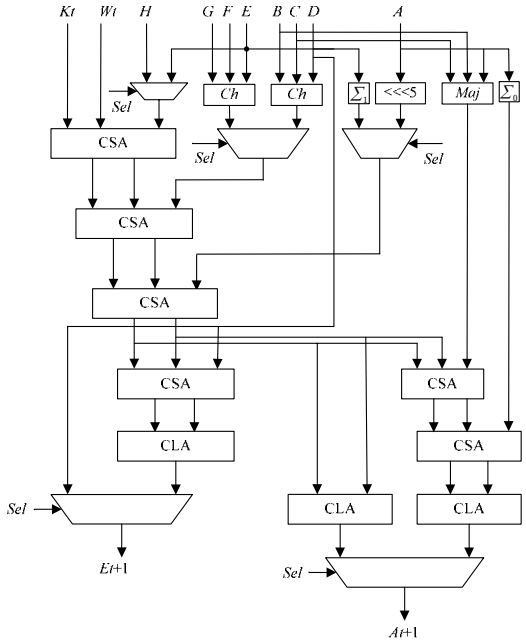


图 3 采用 CSA 的连加运算电路结构

(下转第 136 页)