一种基于 FPGA 的可重构密码芯片的设计与实现

杨晓辉， 戴紫彬
解放军信息工程大学 电子技术学院，河南 郑州 450004
来源：电子技术应用

摘 要： 介绍了 SHA-1、SHA224 及 SHA256 三种安全杂凑算法的基本流程，采用可重构体系结构的设计思想和方法设计出一款可实现这三种算法的可重构密码芯片，并对关键路径进行了优化设计。最后给出了基于 Altera 公司的 Cyclone 系列 FPGA 的可重构密码芯片的实现结果。

关键词： SHA-1/SHA-224/SHA-256 可重构密码芯片 FPGA

目前，国内外广泛使用的密码处理芯片大都是实现某种特定密码算法的专用芯片，如 MD5 芯片、SHA-1 芯片等。由于专用密码芯片实现的密码算法是确定的且不可更改的，因此难以满足不同密码用户多层次的安全性需要。为克服这一缺陷，本文设计一种新型的密码处理芯片——可重构密码芯片。

可重构密码芯片是采用可重构体系结构设计而成的用于对数据进行加/解密处理的集成电路芯片。其内部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，从而能够灵活、快速地实现多种不同的密码算法[1]。此外，由于可重构体系结构设计是建立在某些硬件资源能够被不同应用需求重复使用的基础之上的，所以其消耗的硬件资源要比只实现某种算法的专用芯片所占用的硬件资源的总和要少得多。可重构密码芯片不仅可以灵活实现多种密码算法，还可以更加有效地利用硬件资源以达到节约逻辑资源的目的。本文在分析 SHA-1/SHA-224/256 [2]算法的基础上，选用 Altera 公司的 Cyclone 系列器件，采用 VHDL 语言进行描述，并给出一种能实现该可重构密码芯片的电路设计方案。

1 算法简介

1.1 SHA-1 算法介绍[3]

SHA-1 算法输入报文的最大长度不超过 2^{64}bit，输入按 512bit 分组进行处理，产生的输出是一个 160bit 的报文摘要。该算法处理包括以下几个步骤：

(1)附加填充比特。对报文进行填充使报文长度与 448 模 512 同余(长度=448 mod 512)，填充的比特数范围从 1 到 512，填充比特串的最高位为 1，其余位为 0。

(2)附加长度值。将用 64bit 表示的初始报文(填充前)的位长度附加在步骤(1)的结果后(低位字节优先)。

(3)初始化缓存。使用一个 160bit 的缓存存放该散列函数的中间值及最终结果。该缓存的值分别表示为 A=67452301，B＝EFCDAB89，C=0x98BADCEF，D=0x10325476，E=C3D2E1F0。

(4)处理 512bit(16 个字)报文分组序列。算法的核心是一个包含四个循环的模块，每个循环由 20 个处理步骤组成。四个循环有相似的结构，但每个循环使用不同的逻辑函数，分别表示为 f_1、f_2、f_3、f_4。每个循环都以当前正在处理的 512bit 和 160-bit 缓存值 A、B、C、D、E 为输入，然后更新缓存内容。每个循环还使用一个额外的常数值 K_t，对应的四轮 K_t取值及逻辑函数 f_t如表 1 所示。第四循环最后一步的输出与第一循环的输入进行模 2^{32}相加后得到下一个 512bit 分组计算所需的 A、B、C、D、E 值。

(5)所有的 512bit 分组处理完毕后，最后一个分组产生的输出便是 160bit 的报文摘要。图 1 说明了 SHA-1 算法中每一处理步骤所包含的操作。

表 1 SHA-1 运算常数、函数

步数	常数 K_t	函数 f_t
$0 \leqslant t \leqslant 19$	0x5A827999	$(B \wedge C) \vee (\overline{B} \wedge D)$
$20 \leqslant t \leqslant 39$	0x6ED9EBA1	$B \oplus C \oplus D$
$40 \leqslant t \leqslant 59$	0x8F1BBCDC	$(B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$
$60 \leqslant t \leqslant 79$	0xCA62C1D6	$B \oplus C \oplus D$

1.2 SHA-224/SHA-256 算法介绍

SHA-224/SHA-256 算法输入报文的最大长度不超过 2^{64}bit，输入按 512bit 分组进行处理，产生的输出是一个 224bit 或 256bit 的报文摘要。该算法处理包括以下几个步骤：

(1)和(2)与 SHA-1 算法的前两步相同。

(3)初始化缓存。使用一个 256bit 的缓存存放该散列函数的中间值及最终结果。当执行 SHA-224 算法时,该缓存的值分别表示为 A= 0xC1059ED8, B＝0x367CD507, C=0x3070DD17,D=0xF70E5939, E=0xFFC00B31,F=0x68581511,G=0x64F98FA7,H=0xBEFA4FA4；当执行 SHA-256 算法时，该缓存的值分别表示为 A=0x6A09E667,B=0xBB67AE85,C=0x3C6EF372,D=0xA54FF53A,E=0x510E527F,F=0x9B05688C,G=0x1F83D9AB, H=0x5BE0CD19。

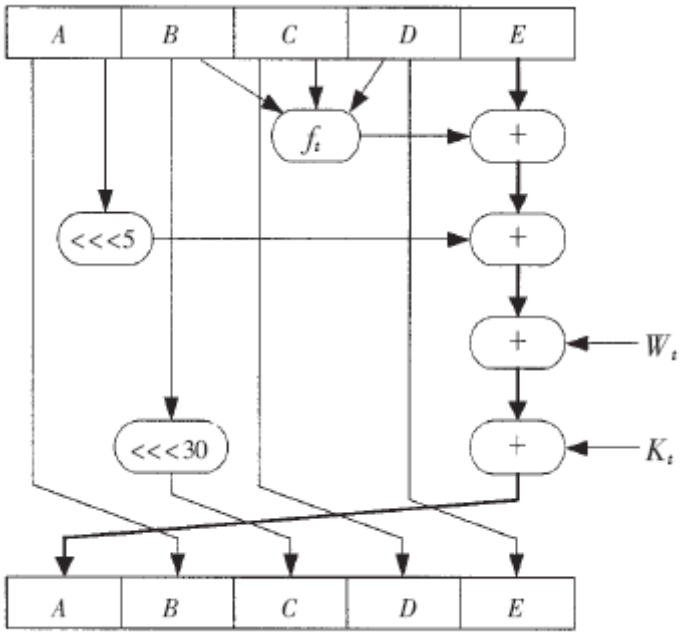

图 1 基本的 SHA-1 操作(单步)

(4)处理 512bit(16 个字)报文分组序列。该算法使用六种基本逻辑函数，由 64 步迭代运算组成。每步都以 256bit 缓存值 A、B、C、D、E、F、G、H 为输入，然后更新缓存内容。每步使用一个 32bit 常数值 K_t 和一个 32bit W_t。六种基本函数如下：

$$Ch(x,y,z)=(x \wedge y)\oplus(x \wedge z)$$
$$Maj(x,y,z)=(x \wedge y)\oplus(x \wedge z)\oplus(y \wedge z)$$
$$\sum_0(x)=ROTR^{28}(x)\oplus ROTR^{34}(x)\oplus ROTR^{39}(x)$$
$$\sum_1(x)=ROTR^{14}(x)\oplus ROTR^{18}(x)\oplus ROTR^{41}(x)$$
$$\sigma_0(x)=ROTR^{1}(x)\oplus ROTR^{8}(x)\oplus SHR^{7}(x)$$
$$\sigma_1(x)=ROTR^{19}(x)\oplus ROTR^{61}(x)\oplus SHR^{6}(x)$$

W_t是由当前的输入分组(512bit 长)导出的 32bit 长的数。在所有 64 次运算完成之后，将其输出 A、B、C、D、E、F、G、H 与第一步的输入 A、B、C、D、E、F、G、H 的值对应进行模 2^{32} 相加。然后将其结果作为下一分组数据 A、B、C、D、E、F、G、H 的值继续运行算法。

(5)所有的 512bit 分组处理完毕后，对于 SHA256 算法,最后一个分组产生的输出便是 256bit 的报文摘要;若是SHA-224 算法,则最后一个分组产生的输出取前 224bit 作为报文摘要。图 2 说明了 SHA-224/SHA-256 算法每一处理步骤所包含的操作。

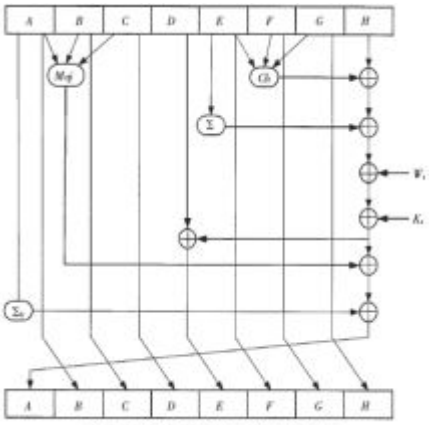

图 2 基本的 SHA-224/SHA-256 单步运算

2 电路结构

2.1 总体结构设计

本设计采用可重构密码芯片的设计思想,通过对 SHA-1、SHA-224、SHA-256 三种算法分析可以看出,这三种算法的 W_t 生成电路和移位存储模块是可重用的部件。数据通路中的 CSA 加法器、存放杂凑值的移位寄存器以及常数值存储模块 K_t 也都是可重用的部件。按照上述分析，把该芯片分为三大模块：存储模块、控制模块、可重用处理模块。其中，存储模块用于存储各种算法所需的常数值。控制模块用于接收外部的控制信号和选择算法信号，控制各种算法的存储和运算。可重用处理模块用于对各种算法进行可重构计算。根据整体算法要求，又将其再往下划分为七个功能子模块。本设计充分利用 FPGA 可重构计算的特点，对可重用的模块进行可重构计算以实现对 FPGA 资源的灵活有效利用。 SHA-1/SHA-224/SHA-256 可重构运算电路 IP CORE 由以下七个子模块构成：控制电路、W_t 生成电路、K_t 常数寄存器、填充电路、运算电路、HASH 值暂存寄存器、移位寄存器。本设计总体结构框图如图 3 所示。

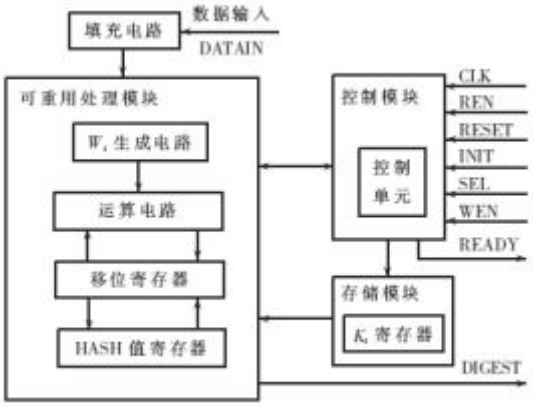

图 3 SHA-1/SHA-224/SHA-256 可重构芯片电路设计整体架构

其中，K_t常数寄存器子模块和 HASH 值暂存寄存器子模块属于存储模块；W_t生成电路子模块、填充电路子模块、运算电路子模块、移位寄存器子模块属于可重用处理模块。控制电路子模块属于控制模块。每个子模块具体功能为：W_t生成电路负责对每组 512bit 的输入数据生成 64 个或 80 个 32bit 的字并送入运算通路。K_t常数寄存器用于存储 64 个或 80 个 32bit 的常量。控制电路负责接收外部的控制信号，并产生所有的内部控制信号，采用计数器电路生成。填充电路接收输入数据，产生每个 512bit 分组并送入 W_t生成电路。运算电路负责计算多个模 2^{32}加法。然后将其结果送入移位寄存器。HASH 值暂存寄存器用于存储 5 个或 8 个 32bit 寄存器的初始值以及每个 512bit 分组运算完毕后的 5 个或 8 个 32bit 寄存器的临时值。移位寄存器负责对每步运算的数据进行移位存储。当 HASH 运算结束时，移位寄存器将其本身值与 HASH 值暂存寄存器中的值相加。当外部读信号有效时，电路实现串行移位功能，在控制信号作用下，将寄存器内的数据顺序读出。

CLK 是系统时钟信号；RESET 是复位信号，RESET 有效时，所有寄存器复位；INIT 是初始化信号，INIT 有效时，初始化摘要计算，在每一批数据进行摘要运算之前，首先执行初始化操作，然后按 512bit 分组写入数据；WEN 是写使能信号，WEN 有效时，在 CLK 时钟上升沿将 512bit 的数据分 16 次由 32bit 数据总线写入芯片内部数据寄存器；REN 是读使能信号，REN 有效时，将内部运算结果读出数据端口；DATAIN 是数据输入，即输入的 512bit 分组数据。SEL 是选择算法指令信号，用来选择所需的算法。READY 是运算状态信号，每 512bit 分组运算完毕后， READY 变为有效，等待读出或外部数据继续输入。DIGEST 是摘要值输出，即 REN 信号有效时输出的相应摘要值。

2.2 控制电路设计

控制电路的核心是一个 7 位计数器，每来一个时钟信号，便进行加 1 操作，根据不同的计数值可给出不同的控制信号。依照算法要求，每处理一组 512 bit 分组数据时，若执行 SHA-1 算法则需要 80 步运算；若执行 SHA-224/SHA-256 算法则需要 64 步。为实现逻辑资源的可重构，达到计数器资源重用的目的，就需要计数器在不同的算法下产生不同的控制信号。本控制电路采用增加选择算法信号(SEL)来实现，其结构如图 4 所示。其中 READY 信号表示每一分组运算完成信号。ADD 信号表示分组运算的最后一步各寄存器的值与第一步计算时的各寄存器的输入值相加信号。START 信号表示移位寄存器移位信号。

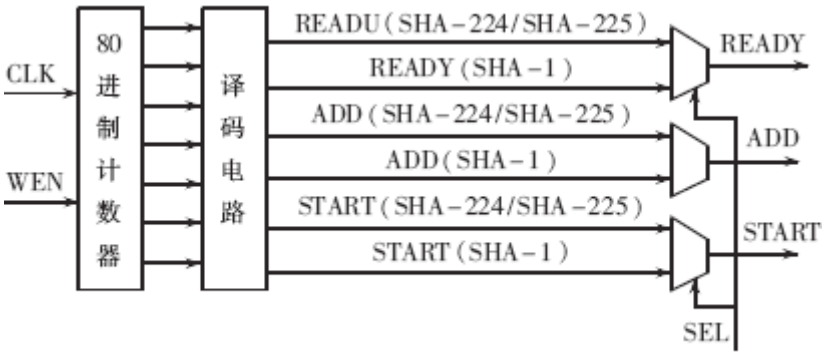

图 4 SHA-1/SHA-224/SHA-256 控制电路设计

2.3 数据路径优化设计

SHA-1 运算模块使用五个寄存器(见图 1)存放散列函数每一步运算的中间结果；SHA-224/SHA-256 运算模块则采用八个寄存器(见图 2)存放散列函数每一步运算的中间结果。为实现可重构计算以使不同算法逻辑单元重用，本设计采用八级移位寄存器 A、B、C、D、E、F、G、H 实现，每个寄存器 32bit 位宽。当执行 SHA-1 算法时，使用前五个 32bit 移位寄存器；当执行 SHA-224/SHA-256 算法时，使用全部八个移位寄存器。当 RESET 信号有效时，寄存器初始化，电路将根据不同算法给寄存器赋初值。数据路径设计的关键是计算每步 A 寄存器的值。当执行 SHA-1 算法时，$A_{t+1}=(E_t+f_t(B_t,C_t,D_t)+A_t(〈〈〈(5)+W_t+K_t) \bmod 2^{32}$ ；当执行 SHA-224/SHA-256 算法时，$A_{t+1}=(H_t+\Sigma 1(E_t)+Ch(E_t,F_t,G_t)+K_t+W_t+\Sigma 0(A_t)+Maj(A_t,B_t,C_t) \bmod 2^{32}$。

式中：A_t、B_t、C_t、D_t、E_t、F_t、G_t、H_t、W_t、K_t是第 t 时刻的各寄存器的值、消息分组和常数值。A_{t+1}是第 t+1 时刻的 A 寄存器的值。

SHA-1/SHA-224/SHA-256 运算模块的关键路径的设计是计算 A_{t+1}，在这一路径中，需要完成多个多变量逻辑函数和多个连续 32 位加法的运算。可以看出，A_{t+1}计算主要包括非线性函数运算、加法运算和移位。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；而加法运算由于进位会在电路上产生延迟，因此应尽量对其进行优化，否则会影响电路运算速度。因此在电路的设计上采用保存进位加法器(CSA)，以减少延迟。由于 SHA-1 算法执行的是五个连续 32 位加法，而 SHA-224/SHA-256 执行的是七个连续 32 位加法，而且 SHA-1 与 SHA-224/SHA-256 所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，再根据选择的算法对不同的值进行选择，然后送入 CSA 加法器的输入端。

本设计电路由五级保存进位加法器(CSA)和两个串行进位加法器(CPA)构成，实现了保存进位加法器对不同算法的重用，其结构如图 5 所示。

3 性能评估

以上设计采用 VHDL 语言描述，在 QuartusII 4.2 环境下编译综合，选用 Altera Cyclone[4]系列器件为目标器件进行整体综合、仿真和底层布局，采用 FIPS 180-2 给出的测试数据进行仿真，采用单个分组和多个分组分别进行测试，均得到正确结果。

在本设计中，当电路执行 SHA-1 算法时运算一个 512bit 分组需要 82 个时钟周期，其中第一个为数据准备周期，第 2~81 个为运算周期，第 82 个为移位寄存器的值与 HASH 值暂存寄存器的值相加并将其结果送入移位寄存器的周期；当执行 SHA-224/SHA-256 算法时运算一个 512bit 分组需要 66 个时钟周期，其中第一个为数据准备周期，第 2~65 个周期为运算周期，第 66 为移位寄存器的值与 HASH 值暂存寄存器的值相加并将其结果送入移位寄存器的周期。该模块的运算速度可以通过下面公式得出：

运算速度=(分组长度/运算耗用的时钟周期数)×系统时钟频率，单位为 Mbps

将实现的可重构算法分别下载到 Altera Cyclone 系列器件后测得的主要性能指标在表 2 中给出。

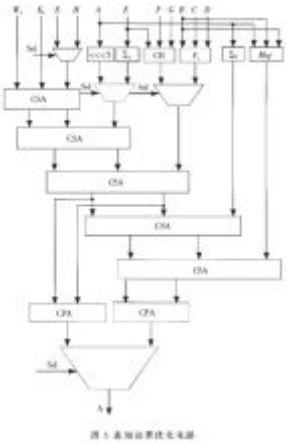

表 2 可重构密码芯片硬件算法性能指标

器件类型	最高时钟（MHz）	占用资源			最高运算速度(Mbps)	
		逻辑单元	引脚	存储单元	SHA-1	SHA-224/SHA-256
EP1C20F324C6	71.81	2628	72	192	448.37	557.07
EP1C20F324C7	62.21	2630	72	192	388.43	482.60
EP1C20F324C8	54.14	2630	72	192	338.04	419.99

表 3 SHA-1/SHA-224/SHA-256 算法专用芯片分别与可重构 SHA-1/SHA-224/SHA-256 算法芯片的参数比较

比较项目 \ 算法类型	SHA-1	SHA-224	SHA-256	SHA-1/SHA-224/SHA-256
所需逻辑单元	1141	1624	1625	2628
所需引脚	72	70	70	72
所需储存单元/bit	192	2048	2048	192
最高时钟频率/MHz	161.79	89.81	87.60	71.81

本设计的创新点是利用可重构设计思想对 SHA-1、SHA-224、SHA-256 三种不同算法的可重用模块进行可重构计算，通过 FPGA 实现时既能灵活实现不同算法，又能实现资源的充分利用，节约大量逻辑资源。为进一步说明本设计对 FPGA 资源利用情况，下面将这三种算法的专用芯片与可重构 SHA-1/SHA-224/SHA-256 算法芯片的一些参数进行比较，如表 3 所示。这里要说明的是这三种算法的参数均是采用类似 SHA-1/SHA-224/SHA-256 的总体电路架构设计得到的。选择的器件均是 EP1C20F324C6。

本文在分析 SHA-1/224/256 三种不同的杂凑算法的基础上，通过 Altera 公司的 Cyclone 系列 FPGA 设计了一款可重构密码芯片。可重构密码芯片是一种创新性的密码芯片，它很好地克服了传统的密码芯片只能实现特定密码算法的弊端，使得密码使用者能够在它上面很方便地选择所需要的密码算法，从而大大提高了密码系统的灵活性。可重构密码芯片可作为构建密码系统的核心部件而被广泛应用于保密通信、网络终端加密设备等领域，因此该研究方向具有重要的政治、军事和经济意义[5]。

参考文献

1 曲英杰. 可重构密码协处理器的组成与结构. 计算机工程与应用, 2003;23:32～34

2 National Institute of Standards and Technology. Announcing the Secure Hash Standard. FIPS 180-2, 2002 August 1:9～20,71～72

3 杜艳华，戴紫彬. 安全散列算法 SHA-1 的 IP CORE 的设计与实现. 电子技术学院学报， 2004；16(1):17～19

4 Altera Corporation. Cyclone Device Handbook .Volume 1.available at http://www.Altera.com.

5 曲英杰，刘卫东，战嘉瑾. 可重构密码协处理器简介及其特性. 计算机工程， 2004;30(13)