(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 108959168 B
(45)授权公告日 2020.09.18

(21)申请号 201810587475.3

(22)申请日 2018.06.06

(65)同一申请的已公布的文献号
申请公布号 CN 108959168 A

(43)申请公布日 2018.12.07

(73)专利权人 厦门大学
地址 361005 福建省厦门市思明南路422号

(72)发明人 李晓潮 张琪 林少宇 黄鹭 王炫榕

(74)专利代理机构 厦门南强之路专利事务所（普通合伙）35200
代理人 马应森

(51)Int.Cl.
G06F 15/78(2006.01)

(56)对比文件
US 2015281254 A1,2015.10.01
US 2016119132 A1,2016.04.28
CN 104158648 A,2014.11.19

审查员 梁滔

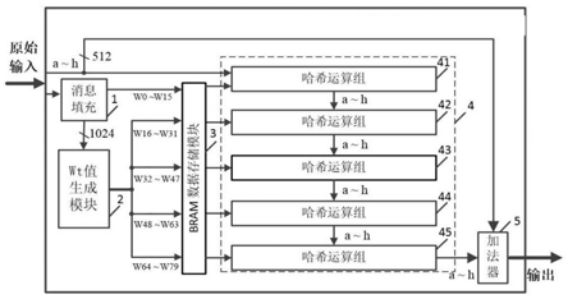
权利要求书2页 说明书5页 附图1页

(54)发明名称

基于片上内存的SHA512全流水电路及其实现方法

(57)摘要

基于片上内存的SHA512全流水电路及其实现方法，SHA512全流水电路设有消息填充模块、W_t值生成模块、BRAM数据存储模块、全流水哈希运算模块和加法器模块，消息填充模块、W_t值生成模块、BRAM数据存储模块、全流水哈希运算模块和加法器模块按顺序连接。在全流水线架构中使用片上BRAM存储模块进行W_t值的数据存储，整个电路系统由依次连接的消息填充模块、W_t值生成模块、全流水哈希运算模块、BRAM存储模块以及加法器模块组成。提高SHA-512算法在FPGA上的吞吐率，平衡FPGA内部资源的分配，提高算法的效率。具有高吞吐率、高单位资源吞吐率的特点，可应用于基于FPGA的SHA-512算法实现。



CN 108959168 B

1.基于片上内存的SHA512全流水电路，其特征在于设有消息填充模块、W_t值生成模块、BRAM数据存储模块、全流水哈希运算模块和加法器模块，所述消息填充模块、W_t值生成模块、BRAM数据存储模块、全流水哈希运算模块和加法器模块按顺序连接；所述W_t值生成模块将读取扩充后的原始数据，依次生成5组共80个W_t值，所述读取扩充后的原始数据构成SHA-512运算所需要的5组数据，其中第1组是消息填充模块的输出数据，剩余4组是通过W_t值生成模块输出的数据，所述5组数据均存储至BRAM数据存储模块中；所述全流水哈希运算模块设有5个哈希运算组，每组16轮哈希运算，整个模块共将实现80轮哈希运算的全流水运算过程，除第一轮哈希运算会读取8个初始哈希值a～h外，后续每轮哈希运算将会读取BRAM存储中的数据和上一轮的哈希运算数据进行计算；

基于片上内存的SHA512全流水的实现方法包括以下步骤：

1)消息填充模块对原始输入数据进行读取，并将消息数据转化为二进制，在消息的结束位置加上结束标志“1”，填充多个“0”，最后加上128位的消息长度信息进行填充，使消息长度为1024的整数倍数，则之后进入SHA-512算法进行运算的数据，位数均为1024的整数倍数；

2)W_t值生成模块将读取填充后的原始数据，将这1024位数据分为每块64位的16个小块，即第一组哈希运算所需的W_t值W_0～W_{15}，之后经过移位、异或的非线性函数计算依次生成后续4组W_t值，即W_{16}～W_{31}、W_{32}～W_{47}、W_{48}～W_{63}、W_{64}～W_{79}；

3)W_t值生成模块生成的W_t值将存入BRAM数据存储模块；

4)全流水哈希运算模块实现80轮哈希循环运算过程；

5)加法器模块的一个输入为哈希运算的最后输出，加法器模块的另一个输入为原始输入数据中的8个初始哈希值a～h，两者相加，即得到了SHA-512算法电路的最终512位信息摘要输出。

2.如权利要求1所述基于片上内存的SHA512全流水电路，其特征在于所述消息填充模块读取原始输入数据，并将消息数据填充至1024位的整数倍数。

3.如权利要求1所述基于片上内存的SHA512全流水电路，其特征在于所述加法器模块的一个输入与哈希运算的输出连接，加法器模块的另一输入为原始输入数据中的8个初始哈希值a～h，将两者相加后得到SHA-512算法的512位信息摘要输出。

4.基于片上内存的SHA512全流水的实现方法，其特征在于采用如权利要求1～3中之一所述电路，所述实现方法包括以下步骤：

1)消息填充模块对原始输入数据进行读取，并将消息数据转化为二进制，在消息的结束位置加上结束标志“1”，填充多个“0”，最后加上128位的消息长度信息进行填充，使消息长度为1024的整数倍数，则之后进入SHA-512算法进行运算的数据，位数均为1024的整数倍数；

2)W_t值生成模块将读取填充后的原始数据，将这1024位数据分为每块64位的16个小块，即第一组哈希运算所需的W_t值W_0～W_{15}，之后经过移位、异或的非线性函数计算依次生成后续4组W_t值，即W_{16}～W_{31}、W_{32}～W_{47}、W_{48}～W_{63}、W_{64}～W_{79}；

3)W_t值生成模块生成的W_t值将存入BRAM数据存储模块；

4)全流水哈希运算模块实现80轮哈希循环运算过程；

5)加法器模块的一个输入为哈希运算的最后输出，加法器模块的另一个输入为原始输

入数据中的8个初始哈希值a～h，两者相加，即得到了SHA-512算法电路的最终512位信息摘要输出。

5.如权利要求4所述基于片上内存的SHA512全流水的实现方法，其特征在于在步骤3)中，所述W_t值生成模块生成的W_t值将存入BRAM数据存储模块的具体方法为：将片上内存BRAM配置为简单双端口，64×256模式，此模式下，允许在同一个有效时钟内，同时对BRAM进行读操作和写操作，每个有效时钟来临时，地址A将会增加1，最新一个W_t值将会被写入上一个时钟周期W_t值的相邻的位置上；当地址A大于256时，其将会被重新置为0，进行循环利用，地址B表示BRAM的读地址，是由同一时钟周期内的地址A与W_t值从写入BRAM到被相应的一轮哈希运算使用所经过的时钟周期数相加而得到的，$W_{t,x}$表示在第x个有效时钟内被写入BRAM的64位W_t值，$W_{t,x+Delay}$表示在同一周期内被读出至哈希运算模块的W_t值；Delay即为W_t值从写入BRAM到被相应的一轮哈希运算使用所经过的时钟周期数，在每个有效时钟周期内，都会有最新的W_t值依据地址A被写入BRAM，同时，哈希运算模块会根据地址B读取该轮运算所需的对应的W_t值，BRAM数据存储模块中将会保存W_0～W_{79}的值，总共80个64位的数据，这些数据经由BRAM存储。

6.如权利要求4所述基于片上内存的SHA512全流水的实现方法，其特征在于在步骤4)中，所述全流水哈希运算模块实现80轮哈希循环运算过程为：每一轮哈希运算将会读取BRAM数据存储模块中的W_t数据，除第一轮哈希运算会另外读取原始输入中的8个初始哈希值之外，之后每轮哈希运算将会读取上一轮的哈希运算数据与W_t数据一起进行计算；输入a_t、b_t、c_t、d_t、e_t、f_t、g_t、h_t为第t轮哈希运算的8个64位哈希值，W_t是由BRAM数据存储模块中读取出，K_t为SHA-512算法K_t常量表中的常量；计算过程中，Maj、Ch、Σ_0、Σ_t为四个非线性计算函数，+为加法器，进位保留加法器为适用于多个加数并缩短延时的加法器；输出a_{t+1}、b_{t+1}、c_{t+1}、d_{t+1}、e_{t+1}、f_{t+1}、g_{t+1}、h_{t+1}为经过一轮哈希运算后新生成的8个64位哈希值；10个中间寄存器δ、a’、b’、c’、d’、e’、γ、f’、g’、λ将原来必须在一个时钟周期内完成的哈希运算分为两个时钟周期完成，第一个时钟周期的运算结果会存入中间寄存器，第二个时钟周期内将从中间寄存器读取数据进行计算，共同完成一轮哈希运算，这就使得关键路径由4个64位的加法运算变成2个64位的加法运算，缩短关键路径提升工作频率；利用进位保留加法器缩短多个数相加产生的延时，把2个64位加法运算的延时缩短为1个非线性函数、1次移位运算以及1次64位加法运算的延时。

基于片上内存的SHA512全流水电路及其实现方法

技术领域

[0001] 本发明涉及信息安全技术领域，尤其是涉及一种高吞吐率、高单位资源吞吐率、高效率的基于片上内存的SHA512全流水电路及其实现方法。

背景技术

[0002] 在信息安全领域，SHA-512算法常用于对信息的完整性和准确性进行验证，是广泛应用于和安全相关的协议和软件中的散列函数之一。SHA-512算法接收少于2的128次方比特的任意长输入信息，并生成固定为512位的信息摘要输出。SHA-512是单向散列函数，是不可逆的字符串变换算法，即无法从一个SHA-512信息摘要逆推得到原始的信息。

[0003] 现场可编程门阵列(Field-Programmable Gate Array，FPGA)是由程序驱动的可编程逻辑器件，在应用方面具有良好的可定制性和灵活性。同时，FPGA具有很高的运算性能，它支持深度可变的流水线结构，并提供大量的并行计算资源，在每个时钟周期内可以完成非常复杂的计算。

[0004] BRAM(Block RAM)是FPGA上的块随机存储单元，可应用于构造数据高速缓冲存储器、深的FIFO和缓冲器等。每块BRAM均可被配置为单端口RAM或双端口RAM，并支持级联。对其进行适当的使用能够极大的节约FPGA上的slice资源，优化设计结构。

[0005] 对现有已公开的技术、文章和发明专利的检索发现，专利公开号CN107612682A的“一种基于SHA512算法的数据处理方法、装置及系统”在FPGA开发板上使用SHA-512四轮分组压缩迭代算法对数据进行处理。在Integration-the VLSI Journal期刊的第47卷4期的On the development of high-throughput and area-efficient multi-mode cryptographic hash designs in FPGAs的文中，作者在FPGA平台上实现了4级流水线式SHA-512算法。在IET Computers&Digital Techniques第8卷第2期的Optimising the SHA-512cryptographic hash function on FPGAs文中，作者在FPGA平台上利用寄存器实现了SHA-512算法的全流水线结构，资源占用巨大。综上所述，现有技术未涉及基于FPGA片上内存BRAM的SHA-512算法全流水线电路的实现方法。

发明内容

[0006] 本发明的目的在于提供基于片上内存的SHA512全流水电路及其实现方法。

[0007] 本发明利用片上内存BRAM(BLOCK RAM)对电路的架构进行优化，从而降低对FPGA上寄存器资源的占用，解决一般流水线设计中存在的寄存器占用多、工作频率低等问题，极大地提高电路的吞吐率以及单位资源吞吐率(Throughout Per Slice，TPS)。

[0008] 所述基于片上内存的SHA512全流水电路设有消息填充模块、W_t值生成模块、BRAM数据存储模块、全流水哈希运算模块和加法器模块，所述消息填充模块、W_t值生成模块、BRAM数据存储模块、全流水哈希运算模块和加法器模块按顺序连接。

[0009] 所述消息填充模块读取原始输入数据，并将消息数据填充至1024位的整数倍数。

[0010] 所述W_t值生成模块将读取扩充后的原始数据，依次生成5组共80个W_t值，所述读取

扩充后的原始数据构成SHA-512运算所需要的5组数据，其中第1组是消息填充模块1的输出数据，剩余4组是通过W_t值生成模块2输出的数据，所述5组数据均存储至BRAM数据存储模块3中。

[0011] 所述全流水哈希运算模块设有5个哈希运算组，每组16轮哈希运算，整个模块共将实现80轮哈希运算的全流水运算过程，除第一轮哈希运算会读取8个初始哈希值a～h外，后续每轮哈希运算将会读取BRAM存储中的数据和上一轮的哈希运算数据进行计算。

[0012] 所述加法器模块的一个输入与哈希运算的输出连接，加法器模块的另一输入为原始输入数据中的8个初始哈希值a～h，将两者相加后得到SHA-512算法的512位信息摘要输出。

[0013] 所述Wt值的参考定义为：输入消息经过填充以及非线性变换后，划分成为的64位数值，用作后续哈希运算的输入值。SHA512算法基于的FIPS(联邦信息处理标准)中给出的说明是Wt：the t‘th word ofthe message schedule直译为消息清单的第t个单元。W为word(unit of language语言的单元)，t为序号。这个值是加密算法中常用的固定的中间值，至今未见对Wt的明确定义。

[0014] 所述基于片上内存的SHA512全流水的实现方法包括以下步骤：

[0015] 1)消息填充模块对原始输入数据进行读取，并将消息数据转化为二进制，在消息的结束位置加上结束标志“1”，填充多个“0”，最后加上128位的消息长度信息进行填充，使消息长度为1024的整数倍数，则之后进入SHA-512算法进行运算的数据，位数均为1024的整数倍数；

[0016] 2)W_t值生成模块将读取填充后的原始数据，将这1024位数据分为每块64位的16个小块，即第一组哈希运算所需的W_t值W_0～W_{15}，之后经过移位、异或等各种非线性函数计算依次生成后续4组W_t值，即W_{16}～W_{31}、W_{32}～W_{47}、W_{48}～W_{63}、W_{64}～W_{79}；

[0017] 3)W_t值生成模块生成的W_t值将存入BRAM数据存储模块；

[0018] 在步骤3)中，所述W_t值生成模块生成的W_t值将存入BRAM数据存储模块的具体方法可为：将片上内存BRAM配置为简单双端口，64×256模式，此模式下，允许在同一个有效时钟内，同时对BRAM进行读操作和写操作，每个有效时钟来临时，地址A将会增加1，最新一个W_t值将会被写入上一个时钟周期W_t值的相邻的位置上；当地址A大于256时，其将会被重新置为0，进行循环利用，地址B表示BRAM的读地址，是由同一时钟周期内的地址A与W_t值从写入BRAM到被相应的一轮哈希运算使用所经过的时钟周期数相加而得到的，$W_{t,x}$表示在第x个有效时钟内被写入BRAM的64位W_t值，$W_{t,x+Delay}$表示在同一周期内被读出至哈希运算模块的W_t值；Delay即为W_t值从写入BRAM到被相应的一轮哈希运算使用所经过的时钟周期数，在每个有效时钟周期内，都会有最新的W_t值依据地址A被写入BRAM，同时，哈希运算模块会根据地址B读取该轮运算所需的对应的W_t值，BRAM数据存储模块中将会保存W_0～W_{79}的值，总共80个64位的数据，这些数据经由BRAM存储。

[0019] 4)全流水哈希运算模块实现80轮哈希循环运算过程；

[0020] 在步骤4)中，所述全流水哈希运算模块实现80轮哈希循环运算过程可为：每一轮哈希运算将会读取BRAM数据存储模块中的W_t数据，除第一轮哈希运算会另外读取原始输入中的8个初始哈希值之外，之后每轮哈希运算将会读取上一轮的哈希运算数据与W_t数据一起进行计算；输入a_t、b_t、c_t、d_t、e_t、f_t、g_t、h_t为第t轮哈希运算的8个64位哈希值，W_t是由BRAM

数据存储模块中读取出，K_t为SHA-512算法K_t常量表中的常量；计算过程中，Maj、Ch、

[0021] Σ_0、Σ_t为四个非线性计算函数，+为加法器，进位保留加法器为适用于多个加数并可以缩短延时的加法器；输出a_{t+1}、b_{t+1}、c_{t+1}、d_{t+1}、e_{t+1}、f_{t+1}、g_{t+1}、h_{t+1}为经过一轮哈希运算后新生成的8个64位哈希值；10个中间寄存器δ、a'、b'、c'、d'、e'、γ、f'、g'、λ将原来必须在一个时钟周期内完成的哈希运算分为两个时钟周期完成，第一个时钟周期的运算结果会存入中间寄存器，第二个时钟周期内将从中间寄存器读取数据进行计算，共同完成一轮哈希运算，这就使得关键路径由4个64位的加法运算变成2个64位的加法运算，缩短关键路径提升工作频率；利用进位保留加法器缩短多个数相加产生的延时，把2个64位加法运算的延时缩短为1个非线性函数、1次移位运算以及1次64位加法运算的延时。

[0022] 5)加法器模块的一个输入为哈希运算的最后输出，加法器模块的另一个输入为原始输入数据中的8个初始哈希值a～h，两者相加，即得到了SHA-512算法电路的最终512位信息摘要输出。

[0023] 本发明相对于现有技术具有如下优点：

[0024] 现有技术不采用BRAM，假设每轮哈希运算需要i个时钟周期来完成，则需要至少3160*i个64位寄存器来对Wt值存储，这极大地占用了FPGA资源。本发明所述的BRAM数据存储模块4可对计算过程中的W_t值进行存取。设计为在全流水线结构合适的位置上，使用BRAM模块存储W_t值，能够节约这些寄存器，并简化全流水线结构。本发明中使用的片上内存BRAM均采用简单双端口配置，此模式下，允许在同一个有效时钟内，同时对BRAM进行读操作和写操作。在每个有效时钟到来时，将会有一个64位的W_t值被写入BRAM中，与此同时，全流水哈希运算模块会从BRAM中读取一轮计算所需的W_t值。故本发明使用片上内存BRAM构造了BRAM数据存储模块存取计算过程中的W_t值，不但减少了FPGA片上寄存器资源的占用，而且还提高了全流水线结构的性能，提高了算法电路的吞吐率和单位资源吞吐率。

[0025] 本发明基于FPGA的高运算性能，首次在SHA-512算法的全流水线架构中使用BRAM数据存储模块，减少电路系统对寄存器的使用，实现了高运行频率、高吞吐率和高单位资源吞吐率的SHA-512全流水线电路。

[0026] 采用上述技术方案和方法后，本发明在FPGA开发平台上进行了具体的实现，使用的FPGA芯片为Xilinx公司出品的Kintex-7系列中的XC7K325T-FFG676-1。通过采用片上内存BRAM对SHA-512算法的全流水线结构中数据进行存储和读取，均衡和优化FPGA中的资源占用，最终成果可提高SHA-512全流水线架构电路的运行频率至268.8M，其吞吐率至275Gbps，单位资源吞吐率至8.66Mbps/slice。

附图说明

[0027] 图1为本发明所述基于片上内存的SHA512全流水电路实施例的结构组成示意图。

[0028] 图2为本发明实施例的BRAM数据存储模块结构图。

[0029] 图3为本发明实施例的一轮哈希运算实现框图。

具体实施方式

[0030] 下面将结合实施例及附图对本发明的技术模块作进一步详细说明。

[0031] 本实施例为基于FPGA片上内存BRAM的SHA-512全流水线电路实现，在全流水线架

构中创新性的使用BRAM数据存储模块存取计算过程中的W_t值，从而减少FPGA上寄存器的占用，实现SHA-512算法的全流水线架构。基于这个设计思想对电路系统进行设计，实现了高效率、高吞吐率和高单位资源吞吐率的SHA-512算法全流水线电路系统。

[0032] 如附图1所示，本发明基于FPGA片上内存BRAM的SHA-512全流水线电路，包含顺序连接的模块分别为：消息填充模块1、W_t值生成模块2、BRAM数据存储模块3、全流水哈希运算模块4以及加法器模块5。算法实施具体步骤如下：

[0033] 消息填充模块1对原始输入数据进行读取，并将消息数据转化为二进制。进一步在消息的结束位置加上结束标志“1”，填充多个“0”，最后加上128位的消息长度信息进行填充，使消息长度为1024的整数倍数。则之后进入SHA-512算法进行运算的数据，位数均为1024的整数倍数。

[0034] W_t值生成模块2将读取填充后的原始数据，将这1024位数据分为每块64位的16个小块，即第一组哈希运算所需的W_t值W_0～W_{15}，之后经过移位、异或等各种非线性函数计算依次生成后续4组W_t值，即W_{16}～W_{31}、W_{32}～W_{47}、W_{48}～W_{63}、W_{64}～W_{79}。

[0035] W_t值生成模块2生成的W_t值将存入BRAM数据存储模块3。本发明中将片上内存BRAM配置为简单双端口，64x256模式，此模式下，允许在同一个有效时钟内，同时对BRAM进行读操作和写操作。如附图2所示，地址A表示BRAM的写地址。每个有效时钟来临时，地址A将会增加1，最新一个W_t值将会被写入上一个时钟周期W_t值的相邻的位置上。当地址A大于256时，其将会被重新置为0，进行循环利用。地址B表示BRAM的读地址，是由同一时钟周期内的地址A与W_t值从写入BRAM到被相应的一轮哈希运算使用所经过的时钟周期数相加而得到的。$W_{t,x}$表示在第x个有效时钟内被写入BRAM的64位W_t值，$W_{t,x+Delay}$表示在同一周期内被读出至哈希运算模块的W_t值。Delay即为W_t值从写入BRAM到被相应的一轮哈希运算使用所经过的时钟周期数。在每个有效时钟周期内，都会有最新的W_t值依据地址A被写入BRAM，同时，哈希运算模块会根据地址B读取该轮运算所需的对应的W_t值。BRAM数据存储模块中将会保存W_0～W_{79}的值，总共80个64位的数据。这些数据经由BRAM存储，可以大大减少电路中寄存器的使用，简化了全流水线设计，使得算法电路的吞吐率以及单位资源吞吐率得到了很大的提升。

[0036] 全流水哈希运算模块4实现80轮哈希循环运算过程。每一轮哈希运算将会读取BRAM数据存储模块中的W_t数据，除第一轮哈希运算会另外读取原始输入中的8个初始哈希值之外，之后每轮哈希运算将会读取上一轮的哈希运算数据与W_t数据一起进行计算。每一轮哈希运算的内部连接如附图3所示，输入a_t、b_t、c_t、d_t、e_t、f_t、g_t、h_t为第t轮哈希运算的8个64位哈希值，W_t是由BRAM数据存储模块中读取出，K_t为SHA-512算法K_t常量表中的常量。计算过程中，Maj、Ch、Σ_0、Σ_t为四个非线性计算函数，+为加法器，进位保留加法器为适用于多个加数并可以缩短延时的加法器。输出a_{t+1}、b_{t+1}、c_{t+1}、d_{t+1}、e_{t+1}、f_{t+1}、g_{t+1}、h_{t+1}为经过一轮哈希运算后新生成的8个64位哈希值。10个中间寄存器δ、a’、b’、c’、d’、e’、γ、f’、g’、λ将原来必须在一个时钟周期内完成的哈希运算分为两个时钟周期完成，第一个时钟周期的运算结果会存入中间寄存器，第二个时钟周期内将从中间寄存器读取数据进行计算，共同完成一轮哈希运算，这就使得关键路径由4个64位的加法运算变成2个64位的加法运算，缩短关键路径提升工作频率。此外，利用进位保留加法器缩短多个数相加产生的延时，把2个64位加法运算的延时缩短为1个非线性函数、1次移位运算、以及1次64位加法运算的延时，进一步缩短了关键路径，使得本设计的工作频率和吞吐率得到提高。

[0037] 加法器模块5的一个输入为哈希运算的最后输出，另一个输入为原始输入数据中的8个初始哈希值a～h，两者相加，即得到了SHA-512算法电路的最终512位信息摘要输出。

[0038] 综上所述，上述实施例公开的基于FPGA片上内存BRAM的SHA-512全流水线电路实现方法，首次在全流水线架构中使用片上内存BRAM进行W_t值的数据存储，减少了对FPGA上寄存器的占用，同时提高了SHA-512算法的工作频率、吞吐率和单位资源吞吐率，解决了SHA-512算法在实际应用中低效率的问题，具有高工作频率、高吞吐率和高单位资源吞吐率的特点。

[0039] 本发明首次在全流水线架构中使用片上BRAM存储模块进行W_t值的数据存储，减少了FPGA上寄存器资源的占用，提高了单位资源吞吐率。整个电路系统由依次连接的消息填充模块、W_t值生成模块、全流水哈希运算模块、BRAM存储模块以及加法器模块组成。这种实现方法不仅极大的提高了SHA-512算法在FPGA上的吞吐率，同时平衡了FPGA内部资源的分配，提高了算法的效率。本发明具有高吞吐率、高单位资源吞吐率的特点，可应用于基于FPGA的SHA-512算法实现。

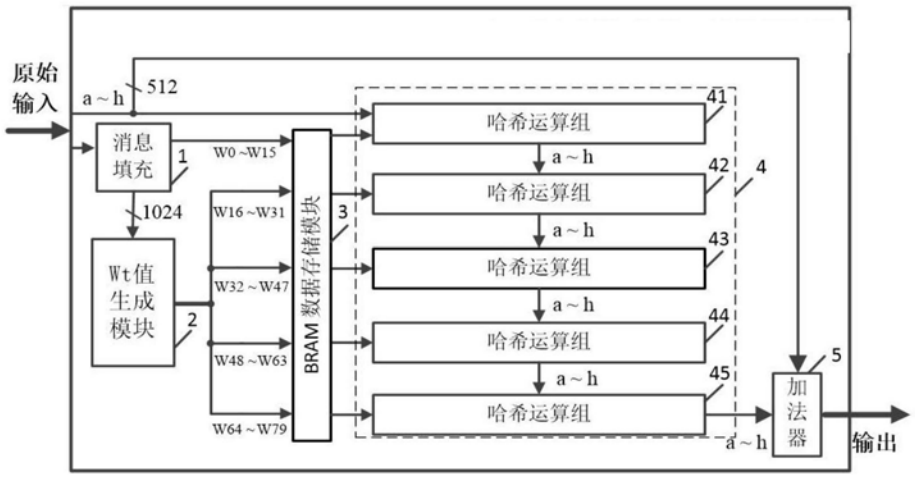

原始输入
a～h
512
消息填充
1
1024
Wt值生成模块
2
W0～W15
W16～W31
W32～W47
W48～W63
W64～W79
BRAM 数据存储模块
3
哈希运算组
41
42
43
44
45
4
a～h
加法器
5
输出

图1

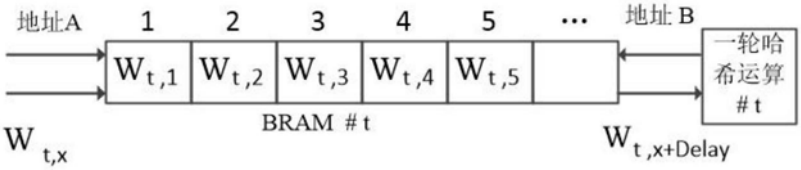

地址A
1
2
3
4
5
…
地址B
$W_{t,1}$
$W_{t,2}$
$W_{t,3}$
$W_{t,4}$
$W_{t,5}$
一轮哈希运算 # t
BRAM # t
$W_{t,x}$
$W_{t,x+Delay}$

图2

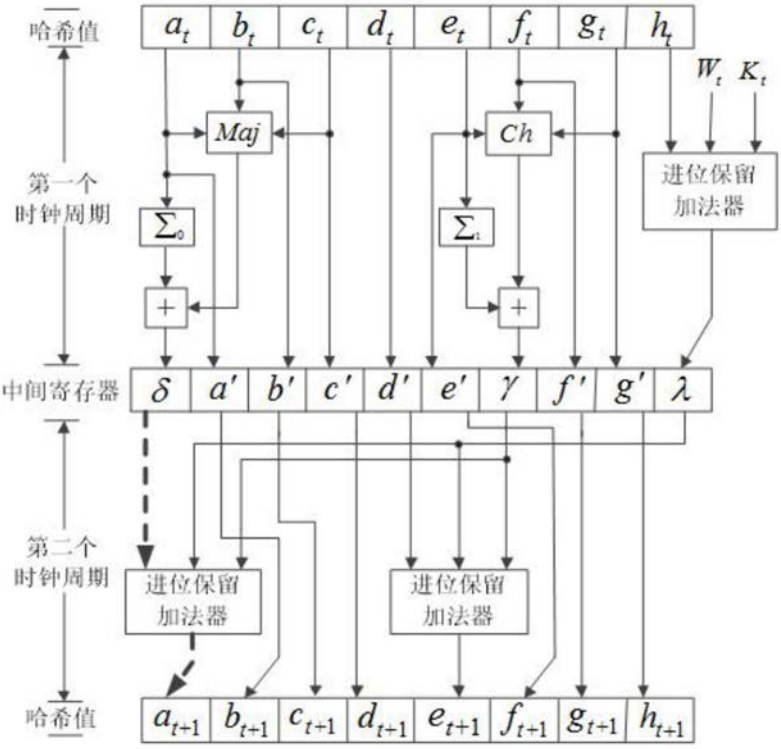

哈希值
a_t b_t c_t d_t e_t f_t g_t h_t
W_t K_t
Maj
Ch
进位保留加法器
第一个时钟周期
Σ_0
Σ_1
+
+
中间寄存器
δ a' b' c' d' e' γ f' g' λ
第二个时钟周期
进位保留加法器
进位保留加法器
哈希值
a_{t+1} b_{t+1} c_{t+1} d_{t+1} e_{t+1} f_{t+1} g_{t+1} h_{t+1}

图3