(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 112532376 B
(45)授权公告日 2024.11.15

(21)申请号 202011411255.9
(22)申请日 2008.03.25
(65)同一申请的已公布的文献号
申请公布号 CN 112532376 A
(43)申请公布日 2021.03.19
(30)优先权数据
11/729,199 2007.03.28 US
(62)分案原申请数据
200880006495.1 2008.03.25
(73)专利权人 英特尔公司
地址 美国加利福尼亚
(72)发明人 S·格伦 W·K·费加利
V·戈帕尔 M·拉古纳丹
M·G·狄克逊 S·陈努帕蒂
M·E·科纳维斯
(74)专利代理机构 北京东方亿思知识产权代理有限责任公司 11258
专利代理师 姜飞
(51)Int.Cl.
H04L 9/06(2006.01)
(56)对比文件
EP 1586971 A2,2005.10.19
WO 2005006197 A2,2005.01.20
审查员 李怡静

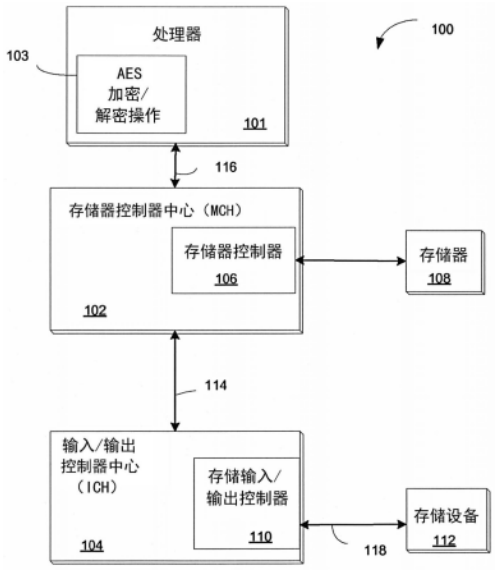
权利要求书1页 说明书11页 附图8页

(54)发明名称

用于高级加密标准(AES)的灵活结构和指令

(57)摘要

提供了用于通用处理器的灵活AES指令集。该指令集包括用于执行AES加密或解密的“一次循环”的指令,并且还包括用于执行密钥生成的指令。可以使用立即数来指示循环数和128/192/256位密钥的密钥生成的密钥长度。由于灵活AES指令集不要求跟踪隐式寄存器,所以可以充分利用管道能力。



CN 112532376 B

1.一种用于处理数据的方法,包括:

解码单循环加密指令以进行高级加密标准AES单循环加密操作,其中所述单循环加密指令来自于AES指令集,该指令集包括单独的AES指令,用于执行加密循环、解密循环、加密最后一个循环、解密最后一个循环,以及用于计算加密循环密钥或解密循环密钥,其中所述单循环加密指令指定用于存储输入数据的目的地寄存器和用于存储循环密钥的源寄存器,其中所述目的地寄存器是128位的寄存器,所述源寄存器是128位的寄存器,并且所述输入数据是128位的值,所述循环密钥是128位的值;并且

执行经解码的单循环加密指令,其中执行单元用于接收所述输入数据和所述循环密钥,并且其中所述执行单元用于使用所述循环密钥对所述输入数据进行所述AES单循环加密操作并在所述目的地寄存器中存储结果数据。

2.根据权利要求1所述的方法,其中,指定的目的地寄存器也是源寄存器。

3.根据权利要求1到2中任一项所述的方法,其中,执行所述经解码的单循环加密指令以进行所述AES单循环加密操作包括:

替换所述输入数据的字节以生成第一中间值;

对所述第一中间值的行进行移位以生成第二中间值;

混合所述第二中间值的列以生成第三中间值;

对所述第三中间值和所述循环密钥进行异或。

4.一种计算机可读介质,具有相关联的指令,其中所述指令在被执行时使得计算机执行如权利要求1到3中任一项所述的方法的步骤。

5.一种处理数据的装置,包括:

寄存器堆;

获取和解码单元,用于解码单循环加密指令,其中所述单循环加密指令来自于AES指令集,该指令集包括单独的AES指令,用于执行加密循环、解密循环、加密最后一个循环、解密最后一个循环,以及用于计算加密循环密钥或解密循环密钥,其中所述单循环加密指令指定所述寄存器堆中的目的地寄存器来存储输入数据并且指定所述寄存器堆中的源寄存器来存储循环密钥,其中所述目的地寄存器是128位的寄存器,所述源寄存器是128位的寄存器,并且所述输入数据是128位的值,所述循环密钥是128位的值;

执行单元,用于执行经解码的单循环加密指令以进行高级加密标准AES单循环加密操作;

其中所述执行单元用于接收所述输入数据和所述循环密钥,并且其中所述执行单元用于使用所述循环密钥对所述输入数据进行所述AES单循环加密操作并且在所述目的地寄存器中存储由所述单循环加密指令的执行产生的结果数据。

用于高级加密标准(AES)的灵活结构和指令

[0001] 本申请是申请日为2008年3月25日、申请号为201710815893.9的同名专利申请的分案申请。

技术领域

[0002] 本公开涉及密码算法,并且具体而言,涉及高级加密标准(AES)算法。

背景技术

[0003] 密码是依靠算法和密钥来保护信息的工具。这种算法是复杂的数学算法,而所述密钥是比特串。密码系统具有两种基本类型:私用密钥系统和公共密钥系统。私用密钥系统又称为对称系统,其具有两方或多方共享的单个密钥(“私用密钥”)。该单个密钥既用于加密信息也用于解密信息。

[0004] 国家标准和技术研究所(NIST)发布的作为联邦信息处理标准(FIPS)197的高级加密标准(AES)是一种私用密钥系统。AES是能够加密和解密信息的对称块加密(block cipher)。

[0005] 加密技术(加密)使用私用密钥(加密密钥)执行一系列的转换,以将被称为“明文”的可理解数据转换为被称为“密文”的无法理解的形式。加密中的转换包括:(1)使用异或(XOR)操作将循环(round)密钥(从加密密钥导出的值)与体(state)(字节的二维阵列)相加;(2)使用非线性字节替换表(S-盒(S-Box))对体进行处理;(3)使用不同的偏移来循环移位所述体的最后三行;以及(4)获取所述体的所有列,并且(相互独立地)混合所述列的数据以产生新的列。

[0006] 解密(反加密)使用加密密钥来执行一系列的转换,以将“密文”块转换为同样大小的“明文”块。反加密中的转换是加密中进行的变换的逆操作。

[0007] 在AES标准中指定Rijindael算法使用长度为128、192和256位的加密密钥来处理128位的数据块。一般将不同的密钥长度称为AES-128、AES-192和AES-256。

[0008] AES算法在10、12或14个连续循环中将明文转换为密文或者将密文转换为明文,其中,循环的次数取决于密钥长度。

附图说明

[0009] 根据下面进行的详细描述以及参考附图,所要求保护主题的实施例的特征将变得明显,其中同样的附图标记代表同样的部件,并且其中:

[0010] 图1是系统的框图,其包括根据本发明的原理用于在通用处理器中执行AES加密和解密的灵活结构和指令的实施例;

[0011] 图2是图1中示出的处理器的实施例的框图;

[0012] 图3是包括图2中示出的用于根据本发明的原理来执行AES加密和解密的执行单元的实施例的框图;

[0013] 图4是示出了通过图3中示出的执行单元的AES加密循环指令流的流程图;

[0014] 图5是示出了通过图3中示出的执行单元的AES加密最后一个循环指令流的流程图;

[0015] 图6是示出了通过图3中示出的执行单元的AES解密循环指令流的流程图;

[0016] 图7是示出了通过图3中示出的执行单元的AES解密最后一个循环指令流的流程图;以及

[0017] 图8示出了可以用来生成循环密钥并执行加密和解密的具有立即(immediate)字节的AES循环指令的实施例。

[0018] 尽管以下详细描述将参考要求保护主题的说明性实施例来进行,但是本领域的技术人员会清楚其多种替换、更改和变化。因此,应当宽泛地看待所要求保护的主题,并且应该仅由所附权利要求对所要求保护的主题进行限定。

具体实施方式

[0019] 高级加密标准(AES)算法是典型地在软件或专用处理器中执行的计算密集算法。因此,典型地,仅将加密操作用于对存储在计算机中的信息的子集进行加密,例如,所述子集为可以归类为“最高机密”的信息。然而,需要对存储在计算机中的更多信息进行加密。例如,如果存储在移动计算机中的所有信息都进行了加密,那么就算该移动计算机被盗,这些信息仍能受到保护。

[0020] AES是利用128、192或256位的密钥长度对128位的比特块进行操作的块加密。根据密钥长度,将一系列的操作反复执行若干个(10、12或14个)循环。

[0021] 通过使用隐式128位寄存器来存储循环密钥,可以动态地(即,刚好在每个循环之前)生成用于每个循环的密钥。然而,由于依赖于先前指令的结果,使用隐式寄存器可能降低x86型基于寄存器的处理器的性能。

[0022] 存在一些应用受益于动态密钥生成,例如,用于处理可能每个流具有不同密钥的网络分组的应用。可能存在要求更高性能的使用单个密钥的其它应用,例如,使用单个密钥来加密/解密磁盘驱动器的内容。因此,出现了对密钥生成的灵活性的要求。本发明的实施例提供了用于在通用处理器中执行AES加密和解密的灵活结构和指令。

[0023] 图1是系统100的框图,其包括根据本发明的原理用于在通用处理器中执行AES加密和解密的灵活结构和指令的实施例。系统100包括处理器101、存储器控制器中心(MCH)或(图形存储器控制器中心(GMCH))102以及输入/输出(I/O)控制器中心(ICH)104。MCH 102包括存储器控制器106,其控制处理器101和存储器108之间的通信。处理器101和MCH 102通过系统总线116进行通信。

[0024] 处理器101可以是多种处理器中的任意一个,诸如:单核Intel®奔腾IV®处理器、单核Intel赛扬处理器、Intel®XScale处理器,或者诸如Intel®奔腾D、Intel®至强®处理器或Intel®酷睿®Duo处理器等多核处理器,或者任何其它类型的处理器。

[0025] 存储器108可以是动态随机存取存储器(DRAM)、静态随机存取存储器(SRAM)、同步动态随机存取存储器(SDRAM)、双数据速率2(DDR2)RAM或者Rambus动态随机存取存储器(RDRAM)或者任何其它类型的存储器。

[0026] ICH 104可以使用高速芯片到芯片互联114(例如,直接介质接口(DMI))来耦合到MCH 102。DMI支持经由两个单向通道的2千兆比特/秒的并发传输速率。

[0027] ICH 104可以包括存储I/O控制器110,用于控制与耦合到ICH 104的至少一个存储设备112的通信。存储设备可以是诸如:磁盘驱动器、数字视频盘(DVD)驱动器、紧致盘(CD)驱动器、独立磁盘冗余阵列(RAID)、磁带驱动器或其它存储设备。ICH 104可以使用串行存储协议通过存储协议互联118与存储设备112进行通信,所述串行存储协议诸如:串行连接小型计算机系统接口(SAS)或者串行高级技术附件(SATA)。

[0028] 处理器101包括AES功能模块103,用于执行AES加密和解密操作。AES功能模块103可以用于对存储在存储器108和/或存储在存储设备112中的信息进行加密或解密。

[0029] 图2是图1中示出的处理器101的实施例的框图。处理器101包括提取和解码单元206,用于对从第一级(L1)指令高速缓存202处接收的处理器指令进行解码。可以将用于执行指令的数据存储在寄存器堆208中。在一个实施例中,寄存器堆208包括多个128位寄存器,AES指令使用这些128位寄存器来存储由AES指令使用的数据。

[0030] 在一个实施例中,寄存器堆是一组128位寄存器,这些128位寄存器类似于在具有流(单指令多数据(SIMD))扩展(SSE)指令集的Intel奔腾MMX处理器中提供的128位MMX寄存器。在SIMD处理器中,以128位的块来处理数据,其中一次载入一个128位的块。

[0031] 提取和解码单元206从L1指令高速缓存202中提取宏指令、解码所述宏指令并且将其分解为称为微操作(μops)的简单操作,这些微操作可以存储在微码只读存储器(ROM)214中。执行单元210调度和执行这些微操作。在示出的实施例中,执行单元210中的AES功能模块103包括用于AES指令集的微操作。退出单元212将所执行指令的结果写入寄存器或存储器。AES指令使用的循环密钥214可以存储在L1数据高速缓存204中,并且可以被载入到执行单元210中以供微操作用于执行AES指令集中的AES指令。将循环密钥214存储在数据高速缓存204中的处理使循环密钥免于受到邻近信道的攻击,例如,尝试获取循环密钥以便能访问存储在系统100中的加密信息。

[0032] 图3是示出了图2中示出的用于根据本发明的原理来执行AES加密和解密的执行单元210的实施例的框图。将结合图2来描述图3。

[0033] 在提取和解码单元206已经将AES指令进行解码之后,执行单元210对AES指令的执行涉及执行可存储在微码ROM 214中的与AES指令相关联的微操作。

[0034] 根据本发明的实施例的灵活AES指令集允许程序员针对待处理的数据量以及存储器带宽和容量来对性能进行折衷。

[0035] 一些应用可以一直使用同一个密钥。在性能至关重要的应用中,可以在每次预先计算密钥的密钥调度(即,每个循环的循环密钥)以及将其存储在存储器中等方面进行折衷。其它应用可能需要将用于存储密钥调度的存储器数量最小化并且仍然能够实现多块操作的良好性能。对于这种应用,可以在处理之前对多个块预先计算密钥调度。通过仅存储加密密钥或反加密密钥以及之后以一些性能为代价导出其它必要的信息,可以进一步使存储器占用量(footprint)最小化。

[0036] 在x86类型的处理器中,可用于AES循环密钥操作和AES调度操作的执行端口的区域和数量限制了AES指令的性能。在每个块加密都需要密钥扩展的系统中,可以通过将AES调度操作和AES循环密钥操作置于单独的执行端口来提高性能。然而,在x86类型的处理器中,可能无法使用单独的执行端口和用于控制该单独端口的附加区域。

[0037] 在一个实施例中,提供了AES指令集,其包括单独的AES指令,用于执行加密循环、

解密循环、加密最后一个循环、解密最后一个循环,以及用于计算加密循环密钥或解密循环密钥。在一个实施例中,AES指令集中存在六个AES指令。每个AES循环指令具有唯一的操作码(opcode)。下面在表1中示出了根据一个实施例的用于固定宽度循环密钥(例如,128位)的AES指令集中的AES循环指令。

[0038] AESENCRYPTRound xmmsrcdst xmm

[0039] 输入: 数据(=目标),循环密钥

[0040] 输出: 使用循环密钥通过AES循环进行转换之后的数据

[0041] AESENCRYPTLastRound xmmsrcdst xmm

[0042] 输入: 数据(=目标),循环密钥

[0043] 输出: 使用循环密钥通过AES最后一个循环进行转换之后的数据

[0044] AESDECRYPTRound xmmsrcdst xmm

[0045] 输入: 数据(=目标),循环密钥

[0046] 输出: 使用循环密钥通过AES循环进行转换之后的数据

[0047] AESDECRYPTLastRound xmmsrcdst xmm

[0048] 输入: 数据(=目标),循环密钥

[0049] 输出: 使用循环密钥通过AES最后一个循环进行转换之后的数据

[0050] AESNextRoundKey xmmsrc1,2xmm dst(immediate)

[0051] 输入: 密钥的低128位,密钥的高128位,循环数的指示符

[0052] 输出: 从输入导出的下一个循环密钥

[0053] AESPreviousRoundKey xmmsrc1,2xmm dst(immediate)

[0054] 输入: 密钥的低128位,密钥的高128位,循环数的指示符

[0055] 输出: 从输入导出的上一个循环密钥

[0056] 表1

[0057] 所述AES指令集包括四个AES循环指令(加密、解密、加密最后一个循环和解密最后一个循环)以及两个AES循环密钥指令(下一个循环密钥以及上一个循环密钥)。所述AES指令集中的AES循环指令包括用于执行加密和解密循环操作的多个单循环操作,其用于除了最后一个循环的所有循环。例如,在表1的AES加密循环(AESENCRYPTRound)单循环指令中,输入数据存储在128位的寄存器(xmmsrcdst)中并且循环密钥存储在另一个128位的寄存器(xmm)中。该指令对存储在128位的xmmsrcdst寄存器中的输入数据(源)执行AES循环操作,并且使用该循环操作的执行结果来改写存储在该128位的xmmsrcdst寄存器中的输入数据。因此,xmmsrcdst先是存储输入数据,而后存储AES循环操作的结果。

[0058] 所述AES指令集还包括:用于最后一个解密循环的AES解密指令,以及用于最后一个加密循环的AES加密指令。例如,在表1的AES加密最后一个循环(AESENCRYPTLastRound)单循环指令中,输入数据存储在128位的寄存器(xmmsrcdst)中并且循环密钥存储在另一个128位的寄存器(xmm)中。该指令对存储在xmmsrcdst寄存器中的输入数据(源)执行AES循环操作,并且使用该循环操作的执行结果来改写存储在xmmsrcdst寄存器中的输入数据。因此,xmmsrcdst先是存储输入数据,而后存储循环操作的结果。xmm寄存器存储用于循环操作的循环密钥。

[0059] 在另一个实施例中,诸如AESENCRYPTRound和AESENCRYPTLastRound的循环指令和

最后一个循环指令可以从存储器(m/128)而不是从寄存器堆304中获取输入,例如,AES循环指令可以是AESENCRYPTRound xmmsrcdst m/128。

[0060] 所述AES指令集中的其它两个AES指令根据密钥的长度(即,128位、192位或256位)来生成用于AES循环的循环密钥。一个AES循环密钥指令生成用于加密操作的循环密钥,以及另一个AES循环密钥指令生成用于解密操作的循环密钥。AES下一个循环密钥(AESNextRoundKey)指令和AES前一个循环密钥(AESPreviousRoundKey)指令中的立即(immediate)字段指定密钥的长度{128,192,256}。

[0061] 在另一个实施例中,不使用立即字段,而是将不同的密钥长度实现为各自具有唯一操作码的单独的指令。在该实施例中,多个AES循环密钥指令包括用于每个循环密钥操作的三个单独的指令,例如,AESNextRoundKey_128、AESNextRoundKey_192和AESNextRoundKey_256,以及,将会具有用于AESPreviousRoundKey的类似的包含三个指令的指令集。在该实施例中,指令集中的指令总数是10个,这与前述实施例中指令总数为6个的情况不同。

[0062] 寄存器堆304具有多个128位的寄存器,其可以由AES指令集中的AES指令使用。所述128位的寄存器可以存储源操作数、循环密钥和AES指令的结果。对于第一循环而言,AES指令接收的源操作数可以是将要进行加密的128位的明文或者是将要进行解密的128位的密文。用于生成128位、192位或256位密钥的密钥调度的密钥可以存储在寄存器堆304中的任何一个128位寄存器308中。循环密钥也可以存储在寄存器堆中的任何128位寄存器308中。所有指令使用寄存器堆中的寄存器并且也可以如前文所述直接地从存储器获取输入。

[0063] 下面在表2中示出了使用表1所示的AES指令集的实施例的源码的实例。在该实例中,在对多个块使用同一个密钥以执行加密的应用中优化了性能。一个此类应用是使用单个密钥来加密磁盘的内容,其中,在将数据存储到磁盘中之前,使用同一个密钥对所有数据进行加密。在该实例中,执行AES-128加密。

[0064] 密钥的长度可以是128位、192位或256位。根据密钥的长度,将要执行的循环次数可以是1、10、12或14,其中,每个循环密钥具有固定的长度(128位)。在循环次数的值为10、12、14的情况下,AES微操作可以针对128位、192位或256位的密钥长度执行标准AES加密和解密。

[0065] 当对多个块使用同一个密钥时,可以预先计算用于每个循环的循环密钥(密钥调度),并且将其存储在存储器(例如,第1级数据高速缓存204)中,从而在对每个块进行加密/解密操作之前不需要再次计算同一个密钥调度。

```
RK[0] = Input Key
For i = 1..10
        RK [i] = AESNextRoundKey (RK[i-1])
End

STATE = Input Block
STATE = STATE xor RK[0]

For i = 1..9
        STATE = AESENCRYPTRound (STATE, RK[i])
End

STATE = AESENCRYPTLastRound (STATE, RK[10])
```

[0066]

[0067] 表2

[0068] 具有10个元素的阵列(RK)用于存储密钥的密钥调度。用于AES-128加密的输入密钥存储在RK[0]中,以及通过从AES指令集中调用AESNextRoundKey指令来预先计算出9个循环密钥RK[0]-RK[1]。AESNextRoundKey指令根据当前循环密钥来计算下一个循环。预先计算的用于密钥调度的循环密钥可以存储在第1级数据高速缓存204的循环密钥214中。

[0069] 在该实例中,作为密钥调度(扩展密钥)的一部分,在进入环路以执行AES循环之前,从寄存器堆304直接输入用于该循环的循环密钥,对所述体和密钥执行异或(XOR)操作。对于循环1到循环9中的每一个循环,从AES指令集调用AESENCRYPTRound指令以对一个循环执行AES循环操作。对于最后一个循环(循环10)而言,从AES指令集调用AESNECYRPTLastRound指令以对最后一个循环执行AES循环操作。

[0070] 在发出第一个AES指令以开始加密或解密操作之前,将待由AES指令进行加密或解密的信息载入到寄存器堆304中的源/目的寄存器306中。用于加密/解密源寄存器306中的信息的密钥存储在寄存器堆304中的一个或多个其它寄存器308中。在128位密钥的情况下,将整个128位密钥存储在寄存器堆304中的任意一个其它128位的寄存器中。对于长度超过128位的密钥而言,将(高于128位的)最高有效位存储在另一个128位寄存器中。

[0071] 在表2所示的实例中,根据密钥来预先计算每个循环的循环密钥,并且在将该循环密钥载入到寄存器堆304的任何一个寄存器308之前,将其存储在第1级数据高速缓存204中。用于每个循环的密钥还可以存储在寄存器堆304中的一个或多个寄存器中,或者可以存储在第1级数据高速缓存204的循环密钥214中。

[0072] AES具有128位的固定块长度以及128、192或256位的密钥长度,并且AES对4x4的字节阵列(即,16个字节(128位的固定块长度))进行操作,将该4x4字节阵列称为“体”。AES算法在10、12或14个连续循环中将128位的明文块转换为128位的密文块(加密)或者将128位的密文块转换为128位的明文块(解密),其中,循环的次数取决于密钥长度(128、192或256位)。

[0073] 在执行每个循环加密操作或解密操作之前,执行单元210接收存储在寄存器堆304中的体和密钥。使用存储在只读存储器(ROM)214的密钥调度器302中的AES指令的微操作来执行每个加密/解密循环操作。在示出的实施例中,体(128位的块体)存储在寄存器306中,并且密钥存储在寄存器堆304的一个或多个其它寄存器308中。在完成AES指令的执行之后,所得到的体存储在寄存器堆304的寄存器306中。所述体可以是由下一个AES循环使用的中间循环数据,或者是AES加密或解密操作的最终结果。

[0074] 在示出的实施例中,密钥调度器302生成在AES循环中使用的循环密钥。密钥调度器302可以实现为多个微码操作,并且可以包括用于执行操作序列的微码操作,所述操作序列用于针对FIPS文档197所定义的128位、197位和256位密钥来生成循环密钥。

[0075] 在另一个实施例中,密钥调度器可以实现为执行单元210中的硬件状态机序列。在另一个实施例中,密钥调度器的一些部分可以实现为存储在微码ROM 214中的微码操作,并且密钥调度器的剩余部分可以实现为执行单元210中的硬件状态机序列。

[0076] 密钥调度器302将密钥的n个字节扩展为扩展密钥(密钥调度)的b个字节,其中,扩展密钥的前面n个字节是原来的密钥。例如,对于128位的密钥而言,将该128位密钥扩展为176字节的扩展密钥,即,11x16字节(128位),其中,前面的16个字节是原来的128位密钥,因此,循环的次数为10。将192位密钥的24个字节扩展为208个字节(13x16字节),以提供12个“循环密钥”,每个循环密钥用于12个循环中的一个,以及,将256位密钥的32个字节扩展为240个字节(15x16字节),以提供14个“循环密钥”,每个循环密钥用于14个循环中的一个。

[0077] 在解码AES指令中的操作码(opcode)之后,将用于控制用于一个AES循环的AES指令中的流的多个参数存储在控制逻辑322中。这些参数包括操作类型(加密或解密)以及该循环是否是最后一个循环。

[0078] AES循环逻辑324可以包括用于以下阶段的微操作:块体314、S-box/逆S-box 316、移位行316和混合逆、混合列或空(称为“混合列”)320和加循环密钥326。

[0079] 在块体314中,使用逐位XOR将对AES循环逻辑324的128位输入(体)与密钥(与该循环关联的扩展密钥的128位的部分)相加,以产生128位的中间值(体)。

[0080] 在S-box/逆S-box 316中,使用可以存储在查询表中并从查询表检索的其它字节值来替换该128位中间值的每个字节,其中,又将所述查询表称为替换盒或“S-Box”。S-box获取一些数量的输入比特m且将其转换为一些数量的输出比特n,并且S-box典型地实现为查询表。典型地使用固定的查询表。通过使用伽罗瓦域(Galois Field)(GF)(2^8)上的逆函数,该操作提供了非线性。例如,可以通过使用m位输入的外部两个比特在查询表中选择行以及通过使用m位输入的内部比特选择列来寻找n位输出。

[0081] 在移位行318中,S-box/逆S-box 316的结果通过比特线性转换,其中,将从替换字节阶段接收的4x4阵列(128位(16字节)的体)的每一行中的字节向左侧进行循环移位。对于4x4阵列中的每一行,对每个字节移位的位数不同。

[0082] 在混合列320中,来自于移位行320的结果通过比特线性转换,其中,将4x4阵列(体)中的每一列视作二进制伽罗瓦域(GF)(2^8)上的多项式,并且之后将其与固定多项式$c(x)=3x^3+x^2+x+2$相乘且对x^4+1取模。最后一个AES循环与其它AES循环的不同之处在于其省略了混合列320。

[0083] 对于该AES循环,混合列阶段320之后的加循环密钥324对来自于扩展密钥的循环

密钥以及移位行318或混合列320的结果执行异或函数。

[0084] 例如,可以发出下述AES指令以执行AES解密的一个循环:

[0085] AESDECRYPTRound xmmsrcdst xmm

[0086] 该实例使用密钥来执行128位的AES加密循环操作,其中,该密钥的扩展密钥由(RK[1],RK[2],…RK[10])来表示。可以通过在发出AES解密循环(AESDECRYPTRound)指令之前发出AESPreviousRoundKey xmmsrc 1,2xmm dst(immediate)指令来生成该循环密钥。可以从第1级数据高速缓存204中直接将循环密钥载入到块体314中,或者可以将循环密钥首先存储在寄存器堆304的寄存器(xmm)中并且之后将循环密钥从寄存器载入到块体314中。

[0087] 当使用不同的密钥来加密/解密每个块时,例如,对于用于加密/解密数据分组的网络接口控制器(NIC)而言,可以在执行加密/解密之前动态地计算每个循环的循环密钥,如下表3中用于AES-128加密的伪码中所示:

[0088]

```
RK[0] = Input Key
STATE = Input Block
STATE = STATE xor RK[0]

For i = 1..9
        RK[i] = AESNextRoundKey (RK[i-1])
        STATE = AESENCRYPTRound (STATE, RK[i])
End
RK [10] = AESNextRoundKey (RK[9])
STATE = AESENCRYPTLastRound (STATE, RK[10])
```

[0089] 表3

[0090] 在该实例中,在将循环密钥用于密钥调度(扩展密钥)的10个循环(即,循环1-9以及循环10(最后一个循环))中的每一个以执行加密之前,生成用于该循环的循环密钥。

[0091] 包括单AES循环指令和单AES循环密钥生成指令的AES指令集允许具有不同次数的循环和密钥调度的AES的变型,即,FIPS文档197未定义的AES的变型。因此,AES指令集中的单个循环AES指令提供了在执行AES加密和解密过程中的灵活性。

[0092] 由于AES指令集所执行的循环的次数是不固定的,所以如果需要的话,可以执行任意次数的循环。例如,如果引入了用于散列或MAC攻击或者AES攻击的新标准,则可以改变循环的次数以支持未来的加密/解密标准。

[0093] 图4是示出了通过图3中示出的执行单元210的AES加密循环指令流的流程图。

[0094] 在方框400处,执行单元210等待AES加密循环指令。如果提取和解码单元206已经解码了AES加密循环指令,则处理继续进行到方框402。如果提取和解码单元206尚未解码AES加密循环指令,则处理保持在方框400中,等待AES加密循环指令。

[0095] 在方框402处,在提取和解码单元206进行指令解码期间,表示将进行加密的指示被存储在控制逻辑322中,并且在执行加密循环过程中使用的循环密钥和128位的块体(源)被从寄存器堆304载入到执行单元210中。处理继续进行到方框404。

[0096] 在方框404处,对128位的块体,即,来自于方框406或418的结果,执行替换操作。使用可以存储在查询表中并从查询表检索的其它字节值来替换该128位的块体的每个字节,其中,又将所述查询表称为替换盒或“S-Box”。S-box获取一些数量的输入比特m且将其转换为一些数量的输出比特n,典型地将S-box实现为查询表。将结果存储为128位的块体。处理继续进行到方框406。

[0097] 在方框406处,128位的块体(4x4阵列)通过比特线性转换,其中,4x4阵列的每一行中的字节向左侧进行循环移位。对于4x4阵列中的每一行,对每个字节移位的位数不同。处理继续进行到方框408。

[0098] 在方框408处,128位的块体(4x4阵列)通过比特线性转换,其中,将4x4阵列(体)中的每一列视作$GF(2^8)$上的多项式,并且之后将其与固定多项式$c(x)=3x^3+x^2+x+2$相乘且对x^4+1取模。处理继续进行到方框410。

[0099] 在方框410处,对于该AES循环,对来自于扩展密钥的循环密钥以及移位行318或混合列320的结果执行异或函数。处理继续进行到方框412。

[0100] 在方框412处,将该循环(128位的块体)的加密操作的结果存储在寄存器堆304的源/目的寄存器302中。用于AES加密指令的处理结束。

[0101] 下面的表4示出了在执行表3所示的伪码之后,使用128位的密钥对128位的块输入执行AES-128加密的结果的实例。

[0102] 128位输入:00112233445566778899aabbccddeeff (十六进制)

[0103] 128位密钥:000102030405060708090a0b0c0d0e0f (十六进制)

[0104] 128位结果:69c4e0d86a7b0430d8cdb78070b4c55a (十六进制)

[0105] 表4

[0106] 图5是示出了通过图3中示出的执行单元210的AES加密最后一个循环指令流的流程图。

[0107] 在方框500处,执行单元等待AES加密最后一个循环指令。如果提取和解码单元206已经解码了AES加密最后一个循环指令,则处理继续进行到方框502。如果提取和解码单元206尚未解码AES加密最后一个循环指令,则处理保持在方框500中,等待AES指令。

[0108] 在方框502处,以结合方框404(图4)所讨论的S-box查询的类似方式来执行最后一个循环的S-box查询。处理继续进行到方框504。

[0109] 在方框504处,与结合方框406(图4)的其它循环所讨论的类似方式来执行最后一个循环的移位行操作。处理继续进行到方框506。

[0110] 在方框506处,对于该AES循环,对来自于扩展密钥的循环密钥以及移位行318或混合列320的结果执行异或函数。处理继续进行到方框508。

[0111] 在方框508处,将加密最后一个循环操作的结果存储在寄存器堆304的源/目的寄存器306中。用于AES指令的处理结束。

[0112] 图6是示出了通过图3中示出的执行单元210的AES解密循环指令流的流程图。

[0113] 在方框600处,执行单元等待AES解密循环指令。如果提取和解码单元206已经解码了AES解密循环指令,则处理继续进行到方框602。如果提取和解码单元206尚未解码AES解密循环指令,则处理保持在方框600中,等待AES解密循环指令。

[0114] 在方框602处,在提取和解码单元206进行指令解码期间,表示将进行解密的指示

被存储在控制逻辑322中,并且在执行解密循环过程中使用的循环密钥和源(128位的块体)被从寄存器堆304载入到执行单元210中。处理继续进行到方框604。

[0115] 在方框604处,将要执行的操作是解密。通过执行AES标准定义的逆S-box查询来对128位的块体执行替换操作。处理继续进行到方框606。

[0116] 在方框606处,按照FIPS文档197的定义执行逆移位行操作。处理继续进行到方框608。

[0117] 在方框608处,按照FIPS文档197的定义执行逆移位行操作。处理继续进行到方框610。

[0118] 在方框610处,对于AES循环,对来自于扩展密钥的循环密钥以及移位行318或混合列320的结果执行异或函数。处理继续进行到方框612。

[0119] 在方框612处,将该循环的解密操作的结果(128位的块体)存储在寄存器堆304的源/目的寄存器302中。用于AES解密循环指令的处理结束。

[0120] 图7是示出了通过图3中示出的执行单元210的AES解密最后一个循环指令流的流程图。

[0121] 在方框700处,执行单元210等待AES解密最后一个循环指令。如果提取和解码单元206已经解码了AES解密最后一个循环指令,则处理继续进行到方框702。如果提取和解码单元206尚未解码AES解密最后一个循环指令,则处理保持在方框700中,等待AES解密最后一个循环指令。

[0122] 在方框702处,通过执行FIPS文档197所定义的逆S-box查询来对128位的块体执行最后一个循环的替换操作。处理继续进行到方框704。

[0123] 在方框704处,按照FIPS文档197的定义对最后一个循环执行逆移位行操作。处理继续进行到方框706。

[0124] 在方框706处,对于AES循环,对来自于扩展密钥的循环密钥以及移位行318或混合列320的结果执行异或函数。处理继续进行到方框708。

[0125] 在方框708处,将解密最后一个循环操作的结果存储在寄存器堆304的源/目的寄存器306中。用于AES解密最后一个循环指令的处理结束。

[0126] 在一个实施例中,图4-图7的流程图中的方框可以实现为执行单元210中的硬件状态机序列。在另一个实施例中,可以将部分方框实现为可以存储在只读存储器(ROM)214中的微程序。将方框实现为硬件状态机序列的实施例可以提供较高的性能。

[0127] 图8示出了可以用来生成循环密钥并执行加密和解密的具有立即字节的AES循环指令的实施例。替代了表1中示出的AES指令集,提供了用于执行AES指令集的功能的单AES循环指令。将单AES指令所执行的特定功能编码在立即字节的比特之中(密钥_选择_修改量(key_select_modifier))。立即字节允许对AES循环指令进行扩展以增加新的特性,取代创建多个各自具有唯一操作码的新指令的方法。

[0128] 可以将AES循环指令符号性地定义为:

[0129] dest:=AES_key_round(source2,source 1),key_select_modifier

[0130] 根据端口号将AES_密钥_循环(AES_key_round)指令发送给特定的执行单元210,以执行AES加密或解密操作。在示出的实施例中,端口号4是指定的用于AES循环指令的执行端口。将执行单元210分为多个并行端口(超标量)。然而,并非所有端口都是等同的。一些端

口具有特定的资源,诸如:大整数乘法器或者浮点乘法器或除法器。在多个端口处支持更简单和更常见的指令,诸如加法、减法和异或,以使性能最大化。因此,对于每个指令或微操作,发出控制逻辑确定用于发出微操作/指令的端口。在该实施例中,总是将AES指令发往端口号4。然而,在其它实施例中,可以使用其它端口号。

[0131] 参考图8,目标字段(dest)存储用于循环N的128位的扩展密钥,源2(source2)字段存储用于循环N-1的128位的扩展密钥,以及源1(souce1)字段存储用于循环N-2的128位的扩展密钥。key_select_modifier是一个8位的立即值,用于提供当前循环数(N)、操作方向(加密/解密)以及AES密钥长度。对于AES-128而言,不需要并且可以省略source1。执行单元是AES单元并且不使用标志(整数或浮点数)。

[0132] 在一个实施例中,立即值的四个最低有效位的位编码指示循环数,例如,AES-128的循环数1-10、AES-192的循环数1-12以及AES-256的循环数2-14。对于AES-128和AES-192而言,由于第一个循环使用未修改的输入密钥,所以循环数0是无效的。对于AES-256而言,由于将未修改的256位的输入密钥用于前两个128位的循环,所以循环数0和1是无效的。

[0133] 立即字节的第4位指示操作方向(加密或解密),例如,在一个实施例中,0=加密且1=解密,以及在另一个实施例中,1=加密且0=解密。立即字节的第5位和第6位指示AES密钥长度。在一个实施例中,按照下面的表5所示来定义AES密钥长度:

[0134]

位[6:5]	密钥长度
00	128
01	192
10	256
11	保留

[0135] 表5

[0136] 在另一个实施例中,值为11的位[6:5]也是128位密钥长度的指示符。在该实施例中,位[6:5]的全部值都是有效的并且可以被解析。

[0137] 本领域的技术人员将理解,本发明的实施例所涉及的方法可以实现在包括计算机可用介质的计算机程序产品中。例如,此类计算机可用介质可以包括只读存储器设备,诸如:存储有计算机可读程序代码的紧致盘只读存储器(CD-ROM)盘或常规ROM设备、或者计算机软盘。

[0138] 虽然已经具体地示出了本发明的实施例并且参考其实施例进行了描述,本领域的技术人员可以理解,可以在不偏离所附权利要求限定的本发明实施例范围的情况下对其形式和细节进行各种改变。

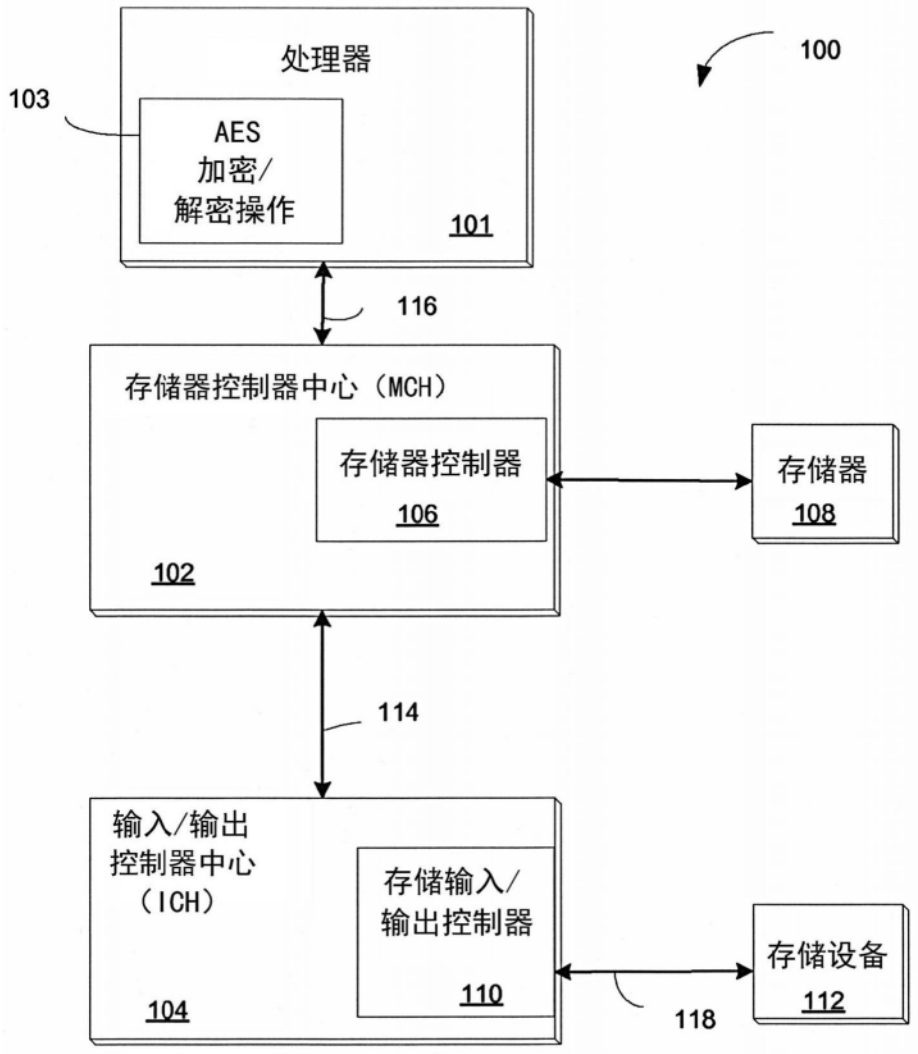

处理器
103
AES
加密/
解密操作
101
100
116
存储器控制器中心（MCH）
存储器控制器
106
存储器
108
102
114
输入/输出
控制器中心
（ICH）
存储输入/
输出控制器
110
存储设备
112
104
118

图1

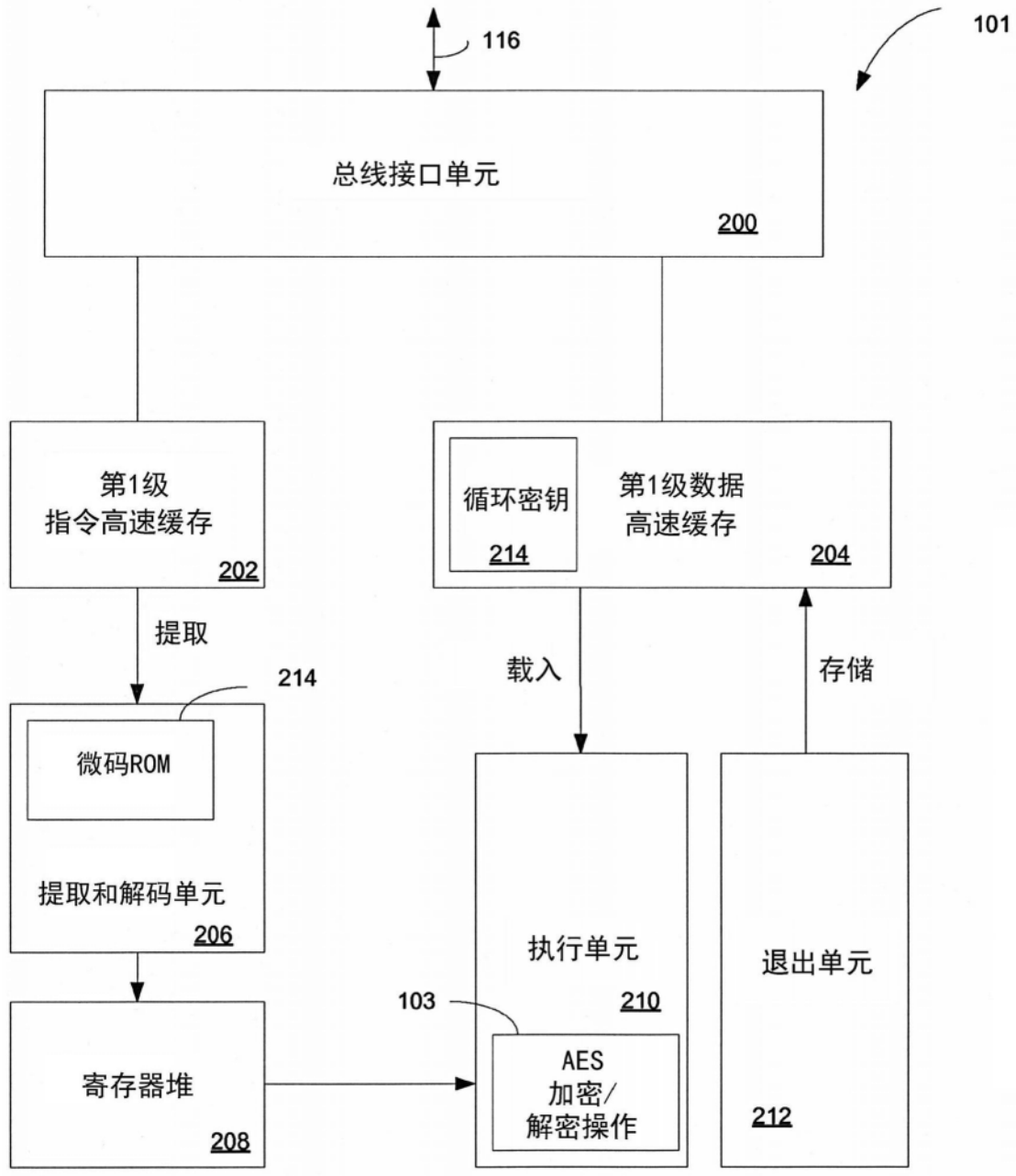

116
101
总线接口单元
200
第1级
指令高速缓存
202
循环密钥
214
第1级数据
高速缓存
204
提取
载入
存储
214
微码ROM
提取和解码单元
206
执行单元
210
103
AES
加密/
解密操作
退出单元
212
寄存器堆
208

图2

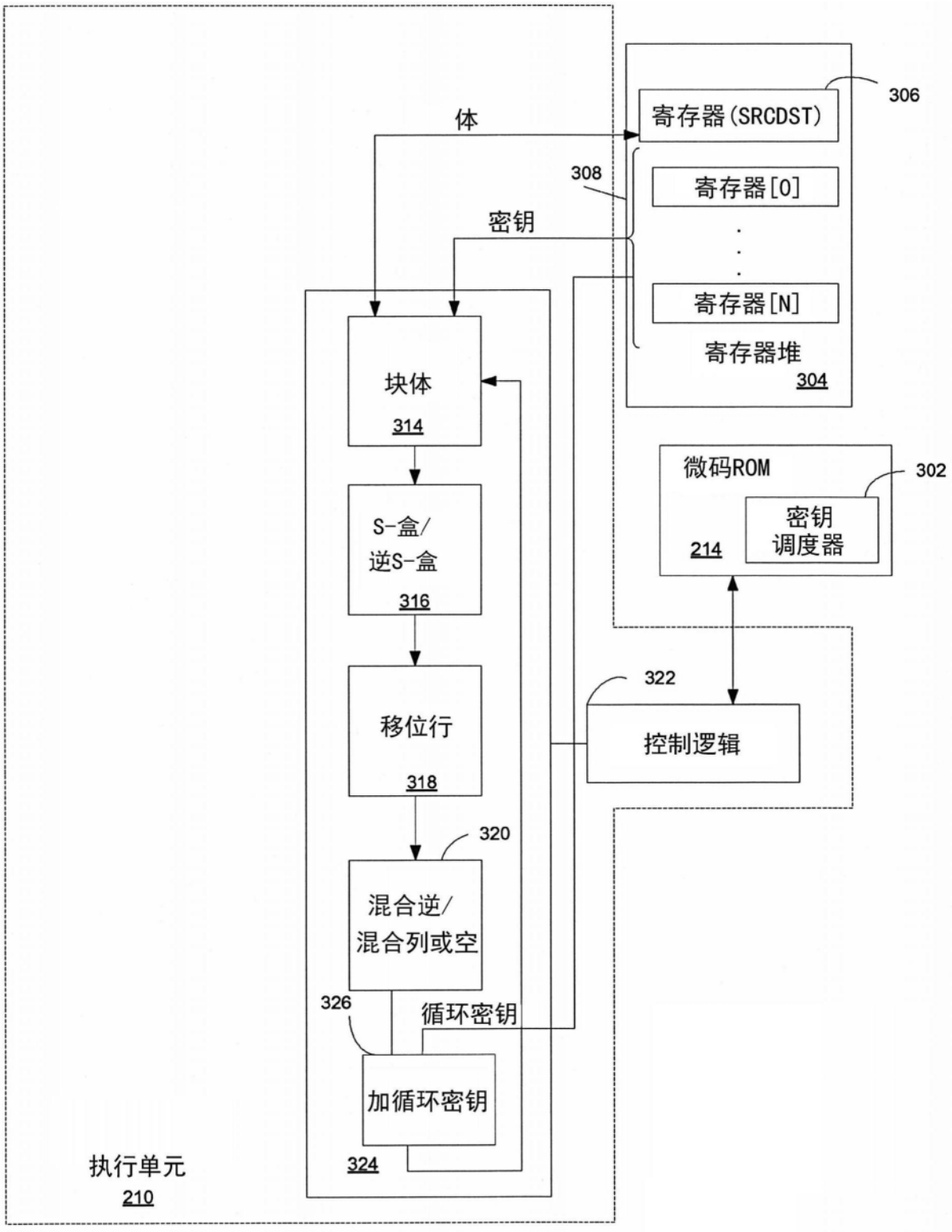

体
密钥
寄存器(SRCDST)
306
308
寄存器[0]
寄存器[N]
寄存器堆
304
块体
314
S-盒/
逆S-盒
316
移位行
318
混合逆/
混合列或空
320
326
循环密钥
加循环密钥
324
执行单元
210
微码ROM
302
密钥
调度器
214
322
控制逻辑

图3

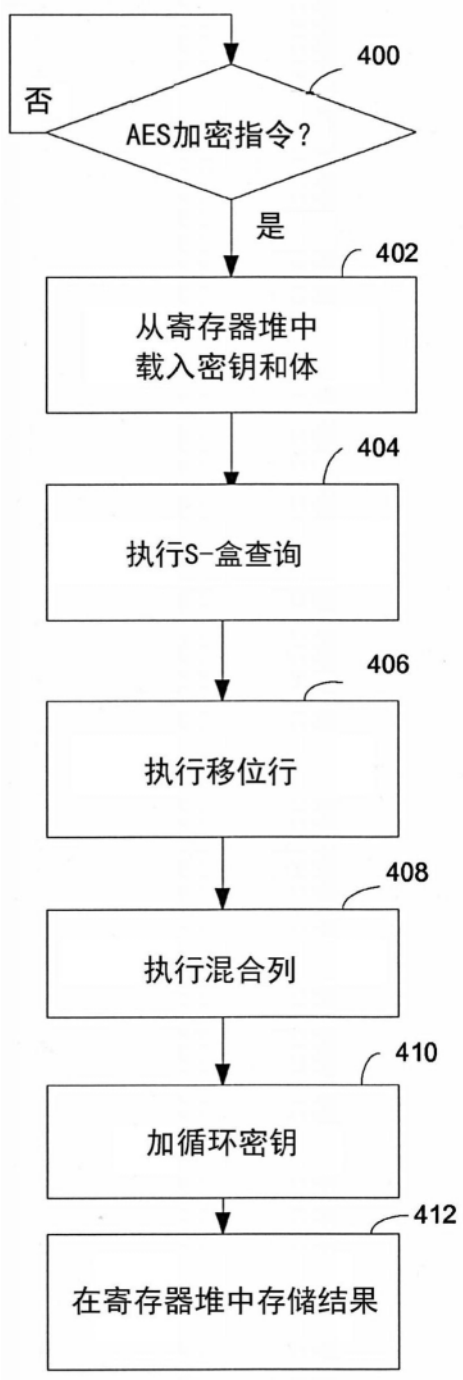

400
否
AES加密指令?
是
402
从寄存器堆中
载入密钥和体
404
执行S-盒查询
406
执行移位行
408
执行混合列
410
加循环密钥
412
在寄存器堆中存储结果

图4

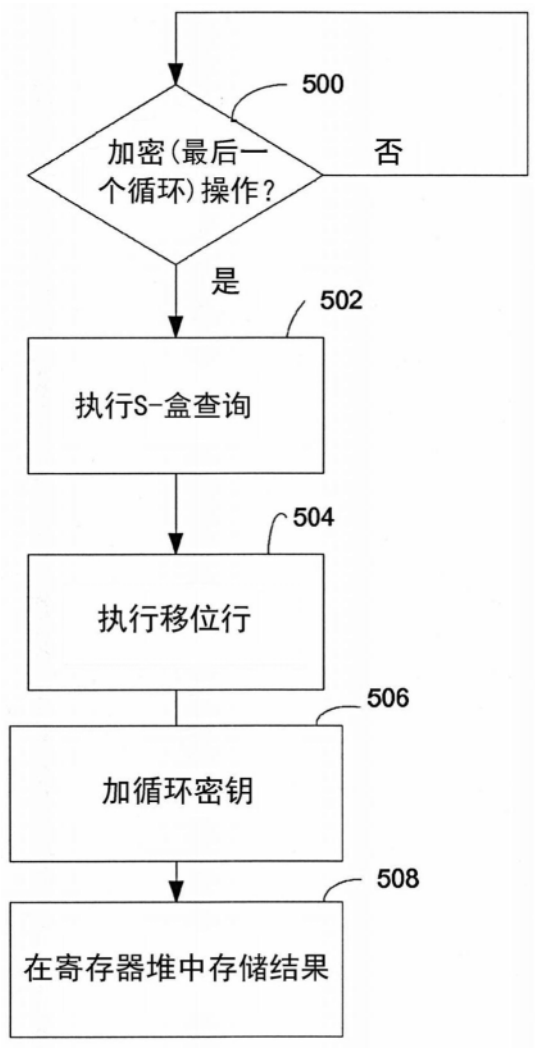

500
加密(最后一个循环)操作?
否
是
502
执行S-盒查询
504
执行移位行
506
加循环密钥
508
在寄存器堆中存储结果

图5

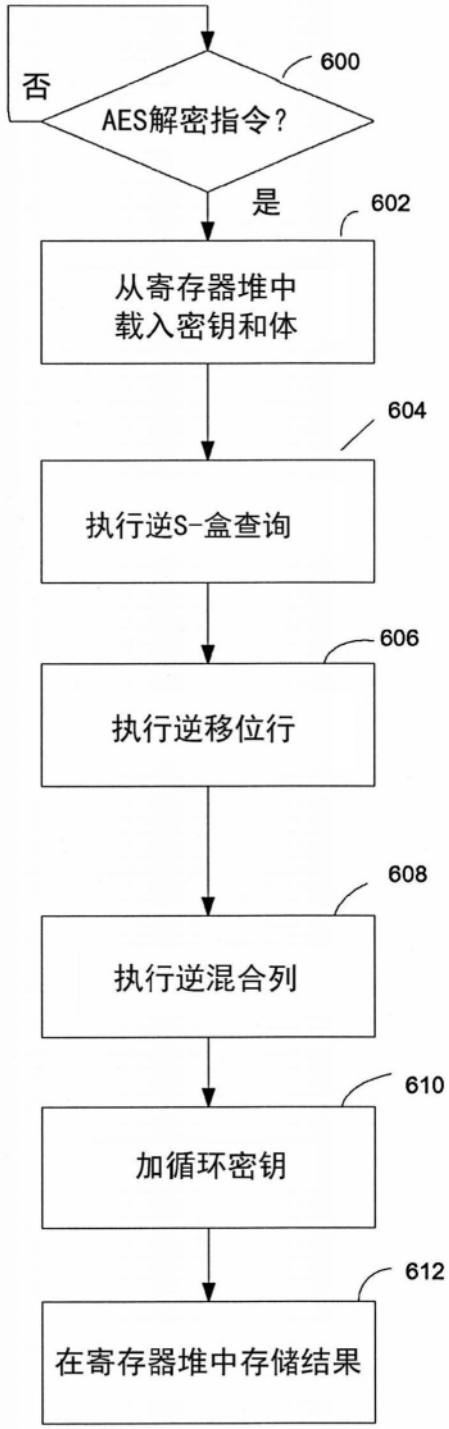

600
否
AES解密指令?
是
602
从寄存器堆中
载入密钥和体
604
执行逆S-盒查询
606
执行逆移位行
608
执行逆混合列
610
加循环密钥
612
在寄存器堆中存储结果

图6

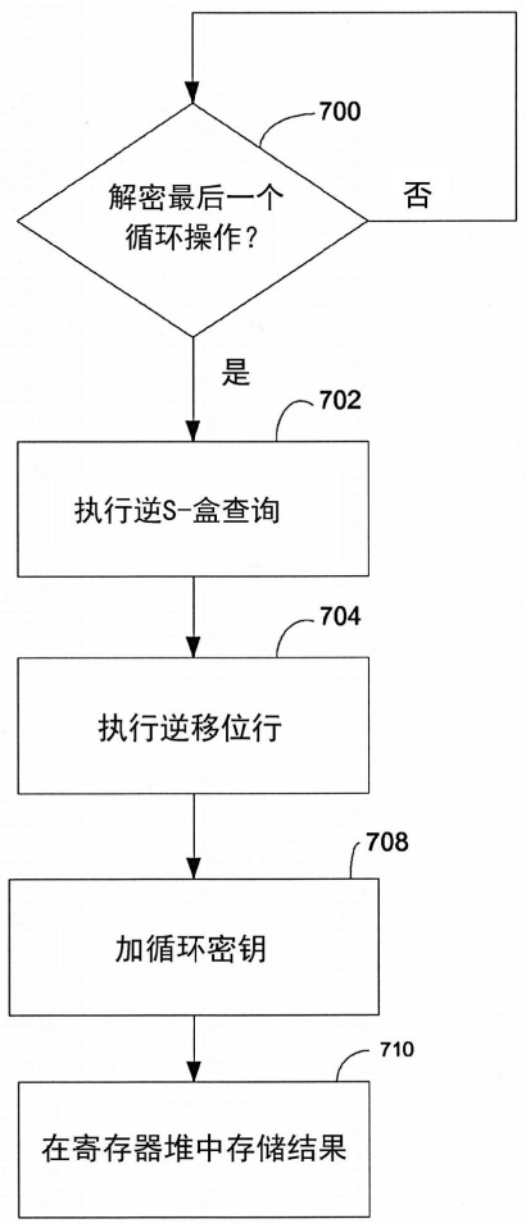

700
解密最后一个
循环操作？
否
是
702
执行逆S-盒查询
704
执行逆移位行
708
加循环密钥
710
在寄存器堆中存储结果

图7

dest:= aes_key_round(source2, source1), key_select_modifier																
端口						整数标志						浮点数标志				
0	1	2	3	4	5	O	S	Z	A	P	C	PreC	C3	C2	C1	C0
-	-	-	-	x	-	-	-	-	-	-	-	-	-	-	-	-

执行单元	AES单元
整数事件	-
操作前事件	N/A
操作后事件	N/A

图8