



(12) 发明专利

(10) 授权公告号 CN 101178758 B

(45) 授权公告日 2012.12.26

(21) 申请号 200710140450.0

说明书第 0088 段, 说明书第 0140 段, 说明书第 0142 段.

(22) 申请日 2005.06.30

审查员 李小青

(30) 优先权数据

2004-194951 2004.06.30 JP

(62) 分案原申请数据

200510080503.5 2005.06.30

(73) 专利权人 富士通半导体股份有限公司

地址 日本神奈川县

(72) 发明人 后藤诚司 蒲田顺 田宫大司

(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

代理人 宋鹤

(51) Int. Cl.

G06F 21/02 (2006.01)

(56) 对比文件

US 2004/0003277 A1, 2004.01.01, 全文.

US 2002/0184046 A1, 2002.12.05, 说明书摘要, 说明书第 0010 段, 说明书第 0068 段, 说明书第 0082 段, 说明书第 0083 段, 说明书第 0086 段, 说

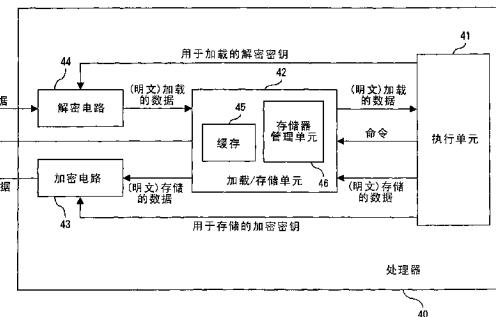
权利要求书 3 页 说明书 24 页 附图 74 页

(54) 发明名称

安全处理器

(57) 摘要

本发明公开了一种安全处理器。指令代码包括存储在下述区域中的指令代码，在所述区域中加密指令代码以不可重写的格式存储，使用执行指令代码的核心所专用的专用密钥或者通过所述专用密钥验证的密钥来验证所述指令代码，以对核心和外界之间的输入和输出数据执行加密处理。



1. 一种安全处理器，包括：

指令执行单元，其执行指令；

加载 / 存储控制单元，其响应于来自所述指令执行单元的命令，控制数据对于外部存储器的加载 / 存储；

加密处理单元，其对所述加载 / 存储控制单元与所述外部存储器之间的数据执行数据加密 / 解密；以及

密钥存储单元，其存储多个密钥，其中

所述指令执行单元响应于被执行的指令，指定用于所述加密处理单元所进行的数据加密 / 解密的密钥，对于多个指令或被执行的指令的多个间隔指定不同的密钥，并且所述指令执行单元将用于指定密钥的密钥号输出到密钥存储单元，并且

所述密钥存储单元响应于所述密钥号，将要被用于数据加密 / 解密的密钥给予所述加密处理单元。

2. 如权利要求 1 所述的安全处理器，其中，所述密钥存储单元存储用于对从外部加载的指令取得数据进行解密的密钥，并且

当所述指令执行单元处于指令取得状态时，所述密钥存储单元将所述用于解密的密钥给予所述加密处理单元。

3. 如权利要求 1 所述的安全处理器，还包括：

密钥号存储单元，其存储用于指定密钥的密钥号，所述密钥号是由所述指令执行单元输出的，其中

所述密钥存储单元响应于从所述密钥号存储单元给出的密钥号，将要被用于数据加密 / 解密的密钥给予所述加密处理单元。

4. 如权利要求 1 所述的安全处理器，还包括：

密钥号存储单元，其存储要被用于对从外部加载的指令取得数据进行解密的密钥的密钥号，其中

当所述指令执行单元处于指令取得状态时，所述密钥存储单元响应于从所述密钥号存储单元输出的密钥号，将要被用于对指令取得数据进行解密的密钥给予所述加密处理单元。

5. 如权利要求 1 所述的安全处理器，其中所述指令执行单元除了输出作为指定所述密钥的信号的密钥号以外，还响应于所述指令，输出管理员 / 用户切换信号。

6. 如权利要求 1 所述的安全处理器，其中所述指令执行单元除了输出作为指定所述密钥的信号的密钥号以外，还输出进程标识符，所述进程标识符包括被执行的指令。

7. 如权利要求 1 所述的安全处理器，其中所述加载 / 存储控制单元还包括：

直写式缓存；以及

读取修改写入单元，其将合并的经由所述加密处理单元从所述外部存储器中加载的数据和要被存储在所述外部存储器中的数据给予所述加密处理单元。

8. 如权利要求 1 所述的安全处理器，还包括数据旁路单元，所述数据旁路单元通过旁路所述加载 / 存储控制单元与所述外部存储器之间的加密处理单元，传输明文数据而不执行加密 / 解密。

9. 一种安全处理器，包括：

指令执行单元,其执行指令;

加载 / 存储控制单元,其响应于来自所述指令执行单元的命令,控制数据对于外部存储器的加载 / 存储;

加密处理单元,其对所述加载 / 存储控制单元与所述外部存储器之间的数据执行加密 / 解密;以及

密钥存储单元,其存储多个密钥,其中

响应于基于被执行的指令的数据 / 指令取得的访问地址,所述指令执行单元将用于指定要被用于数据加密 / 解密的密钥的信号给予所述加密处理单元,对于多个数据 / 指令取得的访问地址指定不同的密钥,并且所述指令执行单元将逻辑地址作为所述访问地址输出到所述密钥存储单元,并且

所述密钥存储单元响应于所述逻辑地址,将用于数据加密 / 解密的密钥给予所述加密处理单元。

10. 如权利要求 9 所述的安全处理器,其中

所述加载 / 存储控制单元响应于从所述指令执行单元给出的命令,将物理地址作为所述访问地址输出到所述密钥存储单元,并且所述密钥存储单元响应于作为所述访问地址的所述物理地址,将用于数据加密 / 解密的密钥给予所述加密处理单元。

11. 如权利要求 9 所述的安全处理器,其中,所述密钥存储单元存储分别对应于作为所述访问地址的逻辑地址和物理地址的多个密钥,并且

所述密钥存储单元向所述加密处理单元给予用于加密 / 解密的密钥,所述密钥与基于来自所述指令执行单元的指令而选择的地址相对应,所述指令指示出应当选择由所述加载 / 存储控制单元给出的物理地址还是从所述指令执行单元给出的逻辑地址作为所述访问地址。

12. 如权利要求 9 所述的安全处理器,还包括:

地址选择指令存储单元,其存储用于地址选择指令的数据,所述地址选择指令由所述指令执行单元输出,并且所述地址选择指令指示出响应于所述逻辑地址和物理地址之一应被给予到所述加密处理单元的密钥,其中

所述密钥存储单元基于存储在所述地址选择指令存储单元中的内容,将响应于所述逻辑地址或物理地址的密钥作为用于数据加密 / 解密的密钥给予所述加密处理单元。

13. 如权利要求 9 所述的安全处理器,其中

所述加载 / 存储控制单元响应于在所述指令被执行时从所述指令执行单元给出的访问地址,选择存储在密钥存储单元中的密钥,并将所述密钥作为用于数据加密 / 解密的密钥给予所述加密处理单元。

14. 如权利要求 13 所述的安全处理器,其中所述指令执行单元将指示密钥存储单元的开 / 关的信号,以及在所述密钥存储单元指示“关”状态时,将给出要被用于数据加密 / 解密的密钥的信号输出到所述加密处理单元,并且响应于所述开 / 关信号,所述加密处理单元在所述密钥存储单元指示“开”时使用从所述密钥存储单元给出的密钥,而在所述信号为“关”时使用从所述指令执行单元给出的密钥,作为用于数据加密和解密的密钥。

15. 如权利要求 9 所述的安全处理器,其中所述指令执行单元除了输出作为指定所述密钥的信号的访问地址之外,还响应于所述指令而输出管理员 / 用户切换信号。

16. 如权利要求 9 所述的安全处理器, 其中所述指令执行单元除了输出作为指定所述密钥的信号的访问地址以外, 还输出包括被执行指令在内的进程标识符。
17. 如权利要求 9 所述的安全处理器, 其中所述加载 / 存储控制单元还包括直写式缓存和读取修改写入单元, 所述读取修改写入单元将合并的经由所述加密处理单元从所述外部存储器中加载的数据和要被存储在所述外部存储器中的数据给予所述加密处理单元。
18. 如权利要求 9 所述的安全处理器, 还包括数据旁路单元, 所述数据旁路单元通过旁路所述加载 / 存储控制单元与所述外部存储器之间的加密处理单元, 传输明文数据而不执行加密 / 解密。

## 安全处理器

### [0001] 分案申请

[0002] 本申请是申请日为 2005 年 6 月 30 日,发明名称为“安全处理器和用于安全处理器的程序”的中国发明专利申请第 200510080503.5 号的分案申请。

### 技术领域

[0003] 本发明涉及用于确保诸如计算机之类的信息处理系统的安全性的系统。更准确地说,本发明涉及安全处理器和用于安全处理器的程序,其能够在如计算机和各种具有内置处理器的设备中防止恶意执行代码的操作。

### 背景技术

[0004] 在使用处理器的系统中,操作可以由程序来描述,因此与完全由硬件配置的系统相比,这些操作显示出高的灵活性,并且可以容易地安装多种功能。由于这些特征,处理器已被用在多种系统中,所述系统例如有个人计算机、PDA、蜂窝电话和信息家庭应用设备,并且随着这些系统越来越普及,已广泛地执行了像在电子商务中这样要求较高安全性级别的处理。虽然已经采取了多种基于系统的措施,例如线路数据加密和用户验证,以便加强安全性,但近年来,在应对计算机病毒的传播和非法访问,以及处理系统级安全性时,软件级或处理器级的安全性成为了问题。

[0005] 例如,当包括蜂窝电话和信息家庭应用设备在内的多种具有内置处理器的设备连接到网络时,该设备很有可能面临与个人计算机类似的外部威胁。当对诸如非法访问指令的问题进行准确分析时,其主要原因在于以下事实,即恶意执行代码在终端中操作。防止恶意代码或不需要的代码在处理器上执行是重要的,但传统上在处理器侧所采取的用于防止恶意代码操作的措施根本不够。结果,未能提供安全的软件执行环境这样的问题仍然存在。

[0006] 接着,传统上如下执行处理:当在主存储器设备或辅助存储器设备中存储数据和用于指令的执行代码时,执行加密以确保安全性,然后加密数据在指令实际执行之前被解密并存储在处理器内的缓存中,并执行处理。在此情况下,用于执行加密处理的硬件被从外部加载在与处理器芯片不同的另一芯片上。因此存在以下问题,即诸如处理速度之类的加密处理性能易于变劣。

[0007] 而且,在这种加密处理中,用于加密数据的加密密钥是在外部芯片上的加密处理侧确定的,因此与要在处理器侧执行的指令的种类、管理员 / 用户模式或用于取数据或指令的访问地址无关。另外,仍然存在以下问题,即无法响应于要执行的指令而选择适当的加密密钥,这是因为处理器侧的执行单元无法指定用于加密和解密的密钥。

[0008] 以下文献可以作为关于此软件执行环境安全性的现有技术。

[0009] 专利文献 1:日本专利申请公开 2002-353960,“code executionapparatus and code distributing method”。

[0010] 该文献公开了一种指令执行装置,其中对加密执行代码进行验证,以确认加密代码的有效性,并且安全处理器取得与加密代码相应的指令,以将其作为安全任务来执行。

[0011] 然而,在这种代码执行装置中,在对应于执行代码的进程与用于验证的密钥之间没有关系。例如,如果在操作系统(OS)中执行了恶意操作,并且然后在程序中分配了另一验证密钥,则恶意代码一定会操作,这样的问题是无法解决的。

## 发明内容

[0012] 本发明的第一目的在于提供一种安全处理器,其基于在以不可重写的格式存储加密指令代码的存储器中作为基础的存储器内容,连续地对例如存储在辅助存储单元中的程序的执行代码进行验证,逐步地扩展安全可靠应用的范围,从而可以仅执行可靠的操作。

[0013] 本发明的第二目的在于,使得可以选择要使用的密钥,以用于被执行的指令的数据和执行代码的加密/解密,以及通过在与处理器相同的芯片上安装加密处理模块,来提高加密处理性能。

[0014] 本发明的第三目的在于,通过在将进程执行代码存储到主存储器中时使用与所述进程相对应的验证密钥对执行代码进行验证,从而仅使已被成功验证的执行代码成为可执行的,进而提高处理器所执行的信息处理的安全性。

[0015] 本发明的一种安全处理器包括指令执行设备、加载/存储控制设备和加密处理设备。所述指令执行设备执行指令。所述加载/存储控制设备响应于来自指令执行设备的命令,控制数据对于外部存储器的加载/存储。所述加密处理设备对加载/存储控制设备与外部存储器之间的数据执行数据加密/解密。另外,所述指令执行设备响应于指令被执行,指定用于所述加密处理设备的数据加密/解密的密钥。

[0016] 本发明的另一种安全处理器包括指令执行设备、加载/存储控制设备和加密处理设备。所述指令执行设备执行指令。所述加载/存储控制设备响应于来自指令执行设备的命令,控制数据对于外部存储器的加载/存储。所述加密处理设备对加载/存储控制设备与外部存储器之间的数据执行数据加密/解密。另外,所述指令执行设备响应于被执行的指令的数据/指令取得的访问地址,为加密处理设备给出指定要被用于数据加密/解密的密钥的信号。

## 附图说明

- [0017] 图1是示出本发明的安全处理器的主要配置的框图;
- [0018] 图2是示出第一实施例中的处理器的基本配置的框图;
- [0019] 图3是第一实施例中的处理器的基本处理流程图;
- [0020] 图4是示出包括代码验证处理模块和加密处理模块的处理的流程图;
- [0021] 图5是在指令区域和数据区域指定了不同的密钥时,加密处理模块的流程图;
- [0022] 图6是通过使用公钥加密的加密密钥的存储系统的说明图;
- [0023] 图7是通过使用公钥加密的加密密钥的存储处理的流程图;
- [0024] 图8是用于具有所添加的认证权威机构的签名的加密密钥的存储系统的说明图;
- [0025] 图9是用于具有所添加的认证权威机构的签名的加密密钥的存储处理的流程图;
- [0026] 图10是在检测到无效指令时的处理流程图;
- [0027] 图11是用于存储在数据区域中的指令的密钥替换处理的流程图;
- [0028] 图12是示出第二实施例中的处理器的基本配置的框图;

- [0029] 图 13 是第二实施例中的处理器的基本处理流程图；
- [0030] 图 14 是示出包括安全核心和通常核心的处理器的基本配置的框图；
- [0031] 图 15 是图 14 所示的处理器的处理的基本流程图；
- [0032] 图 16 是示出在图 14 所示的处理器中,由安全核心对通常核心的中止控制系统的说明图；
- [0033] 图 17 是示出在图 14 所示的处理器中,由安全核心对通常核心的中止控制处理的流程图；
- [0034] 图 18 是具有对应于安全核心的密钥生成机制的处理器的配置框图；
- [0035] 图 19 是在图 18 所示的处理器中的密钥处理系统的一个具体示例的实际说明图；
- [0036] 图 20 是示出第三实施例中的处理器的基本配置的框图；
- [0037] 图 21 是在第三实施例中,具有密钥表存储器的处理器的配置框图；
- [0038] 图 22 是示出在第三实施例中,处于命令访问状态的处理器的配置的框图；
- [0039] 图 23 是示出具有对应于密钥表存储器的密钥选择寄存器的处理器的配置的框图；
- [0040] 图 24 是具有对应于处于命令访问状态的密钥表存储器的密钥选择寄存器的处理器的配置框图；
- [0041] 图 25 是示出密钥表存储器的配置示例的图；
- [0042] 图 26 是示出加密电路和解密电路的配置示例的框图；
- [0043] 图 27 是示出具有数据忽略功能的加密电路和解密电路的配置示例的图；
- [0044] 图 28 是对应于缓存直写系统中的加载存储单元的读取修改写入系统的说明图；
- [0045] 图 29 是示出第四实施例中所示的处理器的基本配置的框图；
- [0046] 图 30 是具有给出了逻辑地址的密钥表存储器的处理器的配置框图；
- [0047] 图 31 是具有给出了物理地址的密钥表存储器的处理器的配置框图；
- [0048] 图 32 是示出第四实施例中的密钥表存储器的配置示例 (No. 1) 的图；
- [0049] 图 33 是另一个示出第四实施例中的密钥表存储器的配置示例 (No. 2) 的图；
- [0050] 图 34 是另一个示出第四实施例中的密钥表存储器的配置示例 (No. 3) 的图；
- [0051] 图 35 是示出包括给定了逻辑地址和物理地址的密钥表存储器的处理器的配置的框图；
- [0052] 图 36 是包括密钥选择寄存器的处理器的配置框图,所述密钥选择寄存器地址选择指令给出到图 35 所示的密钥表存储器；
- [0053] 图 37 是示出图 35 和图 36 中所示的密钥表存储器的配置示例的图；
- [0054] 图 38 是示出在存储器管理单元内具有密钥表的处理器的配置的框图；
- [0055] 图 39 是图 38 所示的数据访问系统的说明图；
- [0056] 图 40 是在密钥表安装在地址映射寄存器中的情况下的数据访问系统的说明图；
- [0057] 图 41 是示出用于基于存储器管理单元的开 / 关状态来切换密钥的处理器的配置的框图；
- [0058] 图 42 是基于存储器管理单元的开 / 关状态来切换密钥的加密 / 解密系统的说明图；
- [0059] 图 43 是第三和第四实施例中的执行单元的 I/O 信号的说明图；

- [0060] 图 44 是第五实施例的处理器系统的详细配置框图；
- [0061] 图 45 是产生安全上下文标识符的系统的说明图；
- [0062] 图 46 是产生安全上下文标识符的方法的说明图；
- [0063] 图 47 是消除安全上下文标识符的系统的说明图；
- [0064] 图 48 是添加到执行代码上的验证信息的说明图；
- [0065] 图 49 是将公钥存储到验证密钥寄存器的存储系统的说明图；
- [0066] 图 50 是将公钥存储到验证密钥寄存器的存储处理的流程图；
- [0067] 图 51 是将加密共享密钥存储到验证密钥寄存器的存储系统的说明图；
- [0068] 图 52 是将加密共享密钥存储到验证密钥寄存器的存储处理的流程图；
- [0069] 图 53 是在页面调入 (page-in) 到物理存储器时处理系统的说明图；
- [0070] 图 54 是在页面调入到物理存储器时的处理流程图；
- [0071] 图 55 是示出验证电路的配置的框图；
- [0072] 图 56 是验证单元的操作流程图；
- [0073] 图 57 是在使用第五实施例中的页面时，存储器访问控制单元的访问检查系统的说明图；
- [0074] 图 58 是说明存储器访问控制单元处的操作示例的图；
- [0075] 图 59 是在取得指令时存储器访问控制单元的处理流程图；
- [0076] 图 60 是说明在使用来自安全核心和通常核心的页面时的访问控制系统的图；
- [0077] 图 61 是具有用于在安全模式和通常模式之间切换的模式寄存器的处理器的配置图；
- [0078] 图 62 是示出安全 DMA 的配置的框图；
- [0079] 图 63 是使用安全 DMA 的数据传输处理的流程图；
- [0080] 图 64 是在由 OS 进行页面调入时的处理的流程图；
- [0081] 图 65 是在第七实施例中的上下文信息加密系统的说明图；
- [0082] 图 66 是用于上下文信息的解密系统的说明图；
- [0083] 图 67 是用于上下文信息的篡改检测信息添加系统的说明图；
- [0084] 图 68 是上下文信息的篡改检测系统的说明图；
- [0085] 图 69 是用于安全操作的上下文信息的加密系统的说明图；
- [0086] 图 70 是用于安全操作的上下文信息的篡改检测信息添加系统的说明图；
- [0087] 图 71 是页面表条目的加密系统的说明图；
- [0088] 图 72 是页面表条目的解密系统的说明图；
- [0089] 图 73 是对页面表条目的篡改检测信息添加系统的说明图；
- [0090] 图 74 是用于页面表条目的篡改检测系统的说明图；并且
- [0091] 图 75 是在将程序加载到计算机以实现本发明时的说明图。

## 具体实施方式

- [0092] 下面参照附图，详细描述本发明的实施例。
- [0093] 图 1 是示出本发明的安全处理器的主要配置的框图。在该图中，本发明的安全处理器 1 包括专用密钥存储设备 2、指令代码存储设备 3、验证处理设备 4 和加密处理设备 5。

[0094] 专用密钥存储设备 2 存储专用密钥,所述专用密钥例如是在安全处理器中执行指令代码的核心中的 CPU 专用密钥。指令代码存储设备 3 例如是加密 ROM 代码区域,其以不可重写的格式存储加密指令代码。验证处理设备 4 使用专用密钥来验证包含存储在指令代码存储设备 3 中的指令代码在内的指令代码,加密处理设备 5 对所述核心和外部存储器之间的 I/O 数据进行加密。

[0095] 在本发明所实现的模式中,加密处理设备 5 将经验证的指令代码加密,并可将其存储在例如是主存储器的存储器设备中,所述存储器设备连接到页面单元上的安全处理器 1,并且验证处理设备 4 可以将验证信息添加到要被验证的指令代码上。

[0096] 接着,图 1 所示的安全处理器 1 可以包括用于执行指令代码的核心:用于仅执行由验证处理设备 4 验证了的指令代码的安全核心,以及用于执行未经验证的常规指令代码的通常核心。

[0097] 在此情况下,使用存储在指令代码存储设备中的加密指令代码来引导(激活)安全核心,并且安全核心可以具有通常核心引导设备,该设备在对安全核心的引导完成之后对通常核心进行引导。另外,安全核心可以包括通常核心监控设备,该设备在引导通常核心之后监控通常核心的操作,并在检测到异常状态时,中止通常核心的操作或分支执行到专用处理。

[0098] 接着,本发明的用于安全处理器的程序使计算机执行以下过程:使用存储器中的程序来执行激活处理的过程,其中加密指令代码以不可重写的格式存储在所述存储器中;验证处理模块,用于执行对包括存储在所述存储器中的指令代码在内的指令代码的验证处理;密钥管理处理,用于管理处理器专用密钥;为密钥表建立运算处理的过程,其中密钥用于经验证处理模块验证的指令代码的加密/解密处理;使用验证处理模块对辅助存储器执行程序验证处理的过程;以及用于执行作为密钥处理监控器的操作的过程,所述密钥处理是在执行包括所激活的操作系统在内的经验证程序时所需的。

[0099] 本发明的安全处理器包括:执行命令的指令执行设备,例如执行单元;响应于来自指令执行设备的命令,控制对外部存储器的数据的加载和存储的加载/存储控制设备,例如加载/存储单元;以及执行加载/存储控制设备与外部存储器之间的数据的加密/解密的加密处理设备,例如加密电路和解密电路。所述指令执行设备响应于命令被执行,指定用于加密处理设备的数据加密/解密的密钥。

[0100] 在本发明的实施例中,安全处理器还可包括存储多个密钥的密钥存储设备,例如密钥表存储器,并且指令执行设备将由密钥指定的密钥号输出到密钥存储设备,并且密钥存储设备可以响应于密钥号,给出要被用于加密处理设备的数据加密/解密的密钥。

[0101] 此外,此安全处理器还可包括存储要被用于对从外部加载的命令取得数据的解密的密钥的密钥存储单元,当指令执行设备处于命令取得状态时,所述密钥存储单元可向加密处理设备提供将所取得的命令解密的密钥。

[0102] 此外,本发明的安全处理器可以包括:执行命令的指令执行设备;响应于来自指令执行设备的命令,控制对外部存储器的数据的加载和存储的加载/存储控制设备;以及执行加载/存储控制设备与外部存储器之间的数据的加密/解密的加密处理设备。所述指令执行设备响应于基于被执行的命令的数据/指令取得访问地址,将指定要被用于数据加密/解密的密钥的信号给出到加密处理设备。

[0103] 在本发明的实施例中,安全处理器还可以包含存储多个密钥的密钥存储设备,并且指令执行设备输出逻辑地址作为所述访问地址,并且密钥存储设备可以响应于逻辑地址,将要被用于数据加密 / 解密的密钥给出到加密处理设备。

[0104] 或者,此安全处理器还可包含存储多个密钥的密钥存储设备,加载 / 存储控制设备响应于从指令执行设备给出的命令,将物理地址作为访问地址而输出,并且密钥存储单元可以响应于物理地址,将要被用于数据加密 / 解密的密钥给出到加密处理设备。

[0105] 本发明的安全处理器还包括:安全进程(上下文)标识符生成设备,在发出进程生成命令的时刻,其生成安全进程标识符,用于与对应于页面的安全进程标识符相比较,在所述页面上建立了安全页面标志,以指示执行代码在执行对应于所述执行代码的所述进程之前被正确地验证;以及进程信息保持设备,例如上下文信息存储单元,其保持作为与该进程相关的信息而生成的安全进程标识符。

[0106] 在本发明的实施例中,与对相应于所述进程的执行代码的验证信息的添加一起,进程信息保持设备可以保持验证密钥,用于在所生成的进程的存续期间执行的执行代码验证。

[0107] 此外,此安全处理器还可包括验证设备,其在对应于所述进程的执行代码被存储在存储器的空闲页面中之后,使用每个页面的验证密钥执行对执行代码的验证,并且安全进程标识符被与该页的地址相对应地存储在处理器内的缓冲器中,并且所述验证设备在验证成功时在该缓冲器中设定安全页面标志。

[0108] 或者,此安全处理器还可包括存储器访问控制设备,其在执行代码的实际执行之前,将存储在所述缓冲器中的安全进程标识符与保持在进程信息保持单元中且对应于要执行的指令代码的安全进程标识符相比较,其中对所述缓冲器设定了相应的安全页面标志,并且所述存储器访问控制设备在两个标识符彼此一致时,允许执行命令的命令执行单元访问存储了执行代码的存储器上的页面。

[0109] 另外,此安全处理器还可包括仅执行经验证执行代码的安全核心和执行未被验证的常规执行代码的通常核心,并且每个核心具有各自的命令执行单元和缓存。

[0110] 而且,此安全处理器还可以包括直接存储器访问设备,其在将执行代码存储在存储器中的同时执行验证执行代码所必需的算术计算,并保持计算的结果以将其给出到验证设备。

[0111] 接着,用于本发明的安全处理器的程序是由计算机使用的程序,其将包含执行代码的页面进行页面调入到存储器中,其使得所述计算机执行以下过程:用于请求计算机中的直接存储器访问机构将所述页面传输到存储器的过程;用于在传输成功完成之后,在执行进程之前在翻译后备缓冲器中的页面表条目中设定关于页面的数据的过程,在所述页面上设定了安全页面标志以指示存储执行代码的页面被正确验证,所述数据包括要与对应于所述页面的安全进程标识符相比较的安全进程标识符,并且是在发出进程产生命令时生成的;以及请求硬件验证页面和在页面表条目中设定指示成功验证的安全页面标志的过程。

[0112] 而且,用于本发明的安全处理器的程序是由计算机使用的程序,其执行对包含执行代码的页面的验证,其使得所述计算机执行以下过程:用于对在存储器中读取的页面执行哈希计算的过程;用于将添加到所述页面上的验证信息解密的过程;用于比较哈希运算结果和解密结果的过程;以及当检测出比较结果一致时,在计算机的翻译后备缓冲器中的

页面表条目中设定指示对所述页面的验证成功的安全页面标志的过程。

[0113] 根据本发明，使用处理器中所保持的以不可重写格式加密的指令代码被用作基本完整点，并且执行对包括操作系统在内的程序的验证，以使得可以通过扩展可靠程序的范围来显著提高系统的安全性级别。

[0114] 此外，根据本发明，在加密处理模块与处理器安装于同一芯片内以提高加密处理性能的同时，可以对与被执行的命令相对应的数据和执行代码进行加密。因此，可以响应于被执行的指令而变更加密等级，以使得可以提高作为系统的安全性级别。

[0115] 而且，根据本发明，可以在执行指令代码之前验证指令代码，然后可以执行进程。并且在检测到与设定了安全页面标志的进程相对应的进程标识符和被执行的进程的进程标识符一致时，执行进程。因此，可以防止处理器上恶意变更的执行代码的操作，结果可以提供安全的软件执行环境。

[0116] 下面，对本发明的安全处理器的整体配置及其处理的概要进行说明，作为第一实施例。

[0117] 图 2 是作为第一实施例而示出安全处理器的基本配置的框图。在该图中，处理器 10 包括以下部件：核心 11，其包含执行单元和缓存；加密处理模块 12，其执行与外部接口的命令处理，并执行对总线数据（程序代码或数据）的加密 / 解密；代码验证处理模块 13，其执行对指令代码的验证；加密 ROM 代码区域 14，在其中对在激活处理器时使用的最基本的程序进行加密并存储；以及 CPU 专用密钥 15，用于对存储在此代码区域 14 中的程序执行解密。加密处理模块 12 的操作将在后面的第三实施例中详细说明，而代码验证处理模块 13 的操作也将在后面的第五实施例中详细说明。

[0118] 虽然命令和数据在核心 11 和加密处理模块 12 之间传输，但用于加密的密钥是受控的。验证接口安装在核心 11 和代码验证处理模块 13 之间。而且，加密处理模块 12 和代码验证处理模块 13 执行对主存储器 17 的访问，并且代码验证处理模块 13 执行对辅助存储器 18 的访问。

[0119] 图 3 是第一实施例中的安全处理器的整体处理流程图。

[0120] 当在图 1 中接通电源时，在步骤 S1，图 2 所示的核心 11 使用 CPU 专用密钥 15 将存储在加密 ROM 代码区域 14 中的程序解密，并执行引导（激活）处理。由于内置 ROM 的缘故，对该程序的篡改是非常困难的。然而，即使通过某种方法而执行了篡改，所述程序已经是加密的，难以执行有意义的篡改。因此，如果程序被正确地引导，则可以确定的是，在程序中没有进行篡改。结果，可以假定存储在加密 ROM 代码区域 14 中的程序是绝对可靠的程序。然后，此状态可被定义为程序的基本可靠点。

[0121] 关于加密 ROM 代码区域 14，如果使用了 AES (Advanced Encryption Standard, 高级加密标准) 方法，所述 AES 方法具有比以 64 比特单位执行加密的 DES (Data Encryption Standard, 数据加密标准) 方法更高的保密性，则其可以安装在处理器的外部而非内部。在此情况下，为了避免响应于诸如指令代码的 NOP 和数据模式的全 0/ 全 1 之类的频繁模式而进行的加密密钥估计，可以使用除了 ECB (electric code book, 电代码簿) 模式之外的其他模式，在所述 ECB 模式中，对于同一明文句子总是输出同一加密句子。

[0122] 随后，在步骤 2，执行用于密钥表（存储器）的操作处理、密钥管理处理和对代码验证处理模块 13 的建立，并将处理的内容定义为相同的完整点 (integrity point)，所述

密钥表安装在加密处理模块 12 中,后面将会提及;所述密钥管理处理使用 CPU 专用密钥 15 来执行对公钥和私钥的生成。

[0123] 随后,在步骤 S3,对存储在辅助存储器 18 中的程序执行验证处理。在第一实施例中,包括操作系统 (OS) 在内的一般程序经由硬盘和网络存储在辅助存储器 18 上,并对这些程序执行验证处理。后面将对所述验证处理进行说明。

[0124] 用于执行所述密钥表操作处理的程序组形成了一个库,这个库称为密钥处理监控器。在步骤 S4,将对安全硬件 20 的访问限制在密钥处理监控器所操作的区段,所述安全硬件 20 包括加密处理模块 12、代码验证处理模块 13 和 CPU 专用密钥 15。密钥处理监控器被操作,并且允许访问安全硬件 20 的状态称为访问级别 1。访问级别 1 由下述硬件来实现,所述硬件监控程序计数器是否指示固定区域中步骤 S4 的密钥处理监控器的地址。

[0125] 当与访问级别 1 相比时,使用包括所述 OS 在内的一般程序而进行的操作被分类成访问级别 2 或访问级别 3。在第一实施例中,OS 被分类成访问级别 2,当在步骤 S5 激活 OS 时,在步骤 S6 执行经验证的程序。访问级别 2 的经验证程序可以请求密钥处理监控器(即用于密钥处理的访问级别 1 的步骤 4),并经由所述密钥处理监控器间接执行诸如对自身空间的加密或数据的加密和解密之类的操作。即使将来自 CPU 之外的程序分配作为访问级别 2,如果它们经过了验证,则也可执行密钥处理。然而,禁止对除了公钥或安全硬件 20 之外的所有密钥进行直接访问,因此即使级别 2 的程序存在问题,除了公钥之外的密钥信息也不会对外暴露。

[0126] 在步骤 S5 的 OS 激活之后,在步骤 S7 执行访问级别 3 的未经验证程序。访问级别 3 的程序无法访问除了公钥之外的所有密钥,也无法请求密钥处理监控器的密钥处理。使用每个访问级处的程序之间的进程间通信来执行从步骤 S4 到步骤 S7 的处理。

[0127] 如上所述,在第一实施例中,如果在处理器激活时,使用存储在加密 ROM 代码区域 14 中的程序而执行的引导处理成功,则建立了程序的完整基本点,并且随着使用所述完整基本点来执行对包括 OS 在内的各种程序的验证,从而扩大可靠程序范围,就可以实现由处理器自身逐步提升系统安全性级别的目的。在启动操作之后,可以在每个验证单元处执行对代码和数据的加密,因此对于保持程序之间的保密性来说,可以保持足够的完整性。在第一实施例中,所说明的方法将访问级别 1 处的处理实现为由处理器核心执行的软件,但级别 1 处的处理的一部分或整个处理也可实现为微代码或接线逻辑。

[0128] 图 4 是示出由图 2 中的代码验证处理模块 13 和加密处理模块 12 执行的处理的概要的流程图。在该图中,在步骤 S10 处的代码验证处理模块的初始处理之后,在步骤 S11 执行加密处理模块中的处理。

[0129] 在图 4 中,起初在步骤 S12 对例如存储在主存储器 17 或辅助存储器 18 中的程序执行代码验证处理。后面将对此处理的细节进行描述。随后,在步骤 S13 确定验证是成功还是失败。如果验证失败,则在步骤 S14 执行对代码执行的终止处理。

[0130] 如果验证成功,则开始在加密处理模块中的处理,并在步骤 S16 确定是否在每个页面单元处都指定了用于加密的密钥。如果未指定,则在步骤 S17 使用随机数生成器等生成随机密钥,但如果指定了,则在步骤 S18 取得所指定的密钥。未指定密钥的情况包括页面是新产生的等等情况,但指定了密钥的情况包括:在所产生的页面被页面调出 (page-out) 一次后重复页面调入 (page-in) 的情况,或者存储了来自外部的加密页面的情况。在步骤

S19，在定义了密钥之后，生成加密页面条目，即后面将会提到的翻译后备缓冲器 (TLB) 中的页面表条目 (PTE)，并分配加密页面以执行对代码或数据的加密。

[0131] 图 5 是在同一处理命令区域和数据区域中分配了不同的加密密钥以执行对代码的加密时，代码验证及其加密处理的总体流程图。在该图中，步骤 S10，即由代码验证处理模块进行的处理与图 4 的情况下相同。

[0132] 在图 5 中，如果代码验证成功，则在步骤 S21，确定是否命令密钥被指定为用于命令区域的密钥。如果未指定，则在步骤 S22 生成随机密钥，但如果指定了，则在步骤 S23 取得所指定的密钥，并在步骤 S24 使用所述随机密钥或指定密钥来生成加密命令页面表条目，即 PTE，并在命令区域中分配加密页面，以执行对命令区域的加密。

[0133] 随后，在步骤 S26，确定是否数据密钥被指定为数据区域中的加密密钥。如果未指定，则在步骤 S27 生成随机密钥，但如果指定了，则在步骤 S28 取得所指定的密钥，以在步骤 S29 生成用于数据的页面表条目，并分配加密页面和对数据区域执行加密。

[0134] 随后，参照图 6 到图 9 来说明在第一实施例中用于获取加密密钥的操作。图 6 和图 7 分别是加密密钥获取操作示例 (No. 1) 中的处理器内部的配置示例，以及其处理的流程图。在该示例中，假定在安全系统中预先在处理器内部保持了处理器专用的 RSA 私钥，通过某种方法将相应的 RSA 公钥输出到处理器外部，并且从外部给出的加密密钥已经被利用此公钥而加密。即，例如，用于在页面单元处进行加密和解密的加密密钥是共用密钥，并且利用公钥进行的再加密对于保持保密性是关键的。

[0135] 图 6 示出了用于将加密密钥设定处理执行到处理器中的处理器 10 的配置。处理器内部包括以下关键模块：加密密钥设定单元 21；解密单元 22；处理器专用 RSA 私钥 23；以及翻译后备缓冲器 (TLB) 24。TLB 内部包括逻辑地址表 25，物理地址表 26 和密钥表 27，这些表对应于所述的页面表条目 (PTE)。从外部向加密密钥设定单元 21 给出用于对包括利用处理器专用 RSA 公钥加密的加密密钥在内的加密密钥进行设定的请求。

[0136] 图 7 是加密密钥获取处理的流程图。当在该图中处理开始时，起初在步骤 S31 由加密密钥设定单元 21 接收加密密钥设定请求，在步骤 S32，使用处理器专用 RSA 私钥 23，对解密单元 22 所接收的经加密的加密密钥进行解密。在步骤 S33 由加密密钥设定单元 21 解密的加密密钥被存储在步骤 S32 处的 TLB24 内部的密钥表 27 中，并且处理终止。

[0137] 图 8 是加密密钥获取操作示例 (No. 2) 中的处理器的配置示例。在该图中，以下是处理器 10 与图 6 的示例 No. 1 的示例相比的不同之处：设有签名核实单元 28，代替了解密单元 22；并且存储了认证权威机构 29 的证书作为认证权威机构的公钥，代替了处理器专用 RSA 私钥 23。假定所述的认证权威机构 29 的证书记录在处理器内部，以避免对此证书的非法替换，并将包括具有认证权威机构的签名的加密密钥在内的加密密钥设定请求提供到加密密钥设定单元 21。

[0138] 图 9 是加密密钥获取操作示例 (No. 2) 的处理的流程图。当在该图中处理开始时，起初在步骤 S36 由加密密钥设定单元 21 与签名一起接收加密密钥，在步骤 S37，由签名核实单元 28 使用认证权威机构的签名和公钥来核实接收到的加密密钥，在步骤 S38 确定核实是否成功。如果核实成功，则将加密密钥设定单元 21 所接收的加密密钥存储在 TLB24 内部的密钥表 27 中，然后处理终止。或者，如果核实失败，则处理立即结束。为了提高加密密钥的可靠性，可以既执行操作示例 No. 1，即保持加密密钥的保密性，又结合执行操作示例 No. 2，

即识别加密密钥。

[0139] 图 10 是在第一实施例中,当在加密命令区域中执行命令期间检测到非法命令时的非法命令对待处理。在该图中,如果在步骤 S41 检测到非法命令,则在步骤 S42 确定该非法命令是不是加密页面内的命令,如果是非加密页面内的命令,则在步骤 S43 执行常规的非法命令对待处理。如果确定其是加密页面内的命令,则在步骤 S44 确定出进行了命令篡改,并执行进程锁定 (process lock down) 作为用于篡改的命令篡改对待处理,或执行对挂起处理的取消,以停止对指令代码的执行。

[0140] 图 11 是在图 5 所说明的对同一进程在命令区域和数据区域中分配了不同加密密钥的情况下,防止在执行存储在数据区域中的指令代码之前将命令检测为非法命令的密钥替换处理流程图。这种指令代码在数据区域中的存储发生在执行编程 I0(PIO), 即由程序对命令进行拷贝的时候。

[0141] 在图 11 中,起初,当在步骤 S46 通过 PIO 将指令代码拷贝到数据区域中时,在步骤 S47 引导密钥替换处理。在该处理中,在步骤 S48 读取与存储了命令的数据页面相对应的数据 PTE, 在步骤 S49, 在删除 PTE 之后, 取得存储在该条目中的加密密钥。此外, 使用数据 PTE 的内容, 即加密密钥, 生成密钥存储在密钥表 27 中的命令 PTE。并且, 在步骤 S51 将命令 PTE 写入 TLB 中, 然后在步骤 S52, 分支执行到拷贝了命令的拷贝区域, 最后执行存储在拷贝区域中的命令。

[0142] 在第一实施例中,如图 2 所说明的那样,在处理器 10 中仅安装了一个包括执行单元和缓存的核心 11, 并且核心 11 作为安全核心, 在作为安全处理器的操作中扮演着中心角色。相反, 在被称为多处理器系统或多核心系统的系统中, 处理可以被分割, 例如, 可将多个核心分类成执行安全操作的安全核心和执行通常操作的通常核心。下面将参照第二实施例来说明这种处理器系统。

[0143] 图 12 是第二实施例中的处理器的基本配置框图。与示出第一实施例的图 2 相比, 有以下几点不同: 在该图中代替核心 11 安装了安全核心 31 和通常核心 32; 在这两个核心 31 和 32 与加密处理模块 12 或代码验证处理模块 13 之间安装了总线接口 33; 在安全核心 31 与加密处理模块 12 之间执行密钥控制; 在安全核心 31 与代码验证处理模块 13 之间执行验证控制; 另外 CPU 专用密钥 15 仅连接到安全核心 31。基本上, 第二实施例的特征在于, 作为图 3 所说明的安全硬件的加密处理模块 12、代码验证处理模块 13 和 CPU 专用密钥 15 仅由安全核心 31 控制。

[0144] 在第二实施例中, 对安全硬件 20 的访问仅限于安全核心 31。在第一实施例中, 在作为安全操作的图 2 的步骤 S4 的密钥处理监控器的操作中可能涉及用户软件, 并且如前所述, 通过对程序计数器的硬件监控来限制访问。在第二实施例中, 不涉及软件, 在软件错误 (bug) 方面没有问题。

[0145] 在第一实施例中, 例如, 在共享系统使用同一核心之前, 必须共享这种访问级别。然而, 由于在第二实施例中使用了不同的核心, 因此在切换访问级别时所要求的诸如计数器清零之类的软件处理的量变少了。

[0146] 图 13 是第二实施例中所使用的处理器的基本处理流程图。与示出第一实施例的图 3 相比, 处理中有以下几点不同。如果假定图 12 中的安全核心 31 和通常核心 32 具有基本平等的关系, 则在通电时, 各个核心使用存储在加密 ROM 代码区域 14 中的程序来执行引

导进程。即,在步骤 S1,在安全核心的引导进程中使用 CPU 专用密钥 15 来将加密程序解密。如果引导成功,则该状态被定义为程序的基本可靠点,随后,安全核心继续例如主要作为密钥处理监控器而操作。

[0147] 相反,通常核心 32 主要负责与访问级别 2 等同的处理,例如 OS。响应于图 13 中步骤 S3 的安全核心侧辅助存储器上程序的验证处理,在步骤 S55,在通常核心 32 侧通电,并利用加密 ROM 代码区域 14 中的程序来执行引导进程。假定由安全核心 31 确认了加密 ROM 代码区域 14 中的程序绝对可靠,则通常核心侧的引导过程基本上没有任何问题地结束,并在步骤 S5 连续执行诸如 OS 之类的其他引导进程。

[0148] 图 14 是在第二实施例中,当安全核心和通常核心不具有平等关系,并且通常核心由安全核心控制以便严格实施安全性时,处理器的配置框图。与安全核心 31 和通常核心 32 具有基本平等关系的图 12 相比,处理器的配置组件是相同的,但区别在于核心控制信号是从安全核心 31 给出到通常核心 32 的。核心控制信号的实际示例包括复位信号和中断信号。

[0149] 图 15 是图 14 所示的处理器的全部处理的流程图。在该图中的安全核心 31 侧,在步骤 S1 的引导进程之后,代替步骤 S2,在步骤 S57 建立密钥表操作处理、密钥管理处理和验证处理模块时,还执行系统审计 (systemauditing)。在该系统审计中,核实系统配置改变的存在以及辅助存储器上程序中改变的存在,以便确认系统和系统配置的安全性功能没有问题。

[0150] 随后,在步骤 S58 从安全核心 31 侧引导通常核心,并且作为响应,在步骤 S59,在通常核心 32 侧对存储在加密 ROM 代码区域 14 中的程序进行引导。随后的处理与图 3 所示情况下的处理相同。

[0151] 图 16 是在图 14 所示的处理器中,作为安全核心 31 对通常核心 32 进行的控制处理之一的通常核心停止控制处理的说明图。在该图中,例如,当在步骤 S6 在通常核心侧执行经验证的程序时,向安全核心 31 侧请求了用于验证数据的密钥处理。当检测到在步骤 S4 的密钥处理监控器的操作中的验证失败以及违反了安全性标准时,通过来自安全核心 31 侧的指令,由通常核心 32 执行的诸如步骤 S6 处对经验证程序的执行和步骤 S7 处对未经验证程序的执行之类的处理被中止。

[0152] 图 17 是图 14 所示的安全核心 31 对通常核心 32 的控制处理的流程图。在步骤 S61,在安全核心 31 侧执行引导。当在步骤 S62 完成处理时,在通常核心 32 侧执行对通常核心 32 的引导控制。然后,在步骤 S63 引导通常核心,并在步骤 S64,在通常核心侧执行不需要密钥和验证处理的常规处理。在安全核心 31 侧,在步骤 S65 始终执行使用发送自通常核心 32 侧的监控信息而进行的验证 / 监控处理。在步骤 S66 确定是否产生错误。如果没有错误,则继续步骤 S65 之后的后续处理,但如果有错误,则通过向通常核心 32 侧请求中止或中断,来中止通常核心 32 侧的处理。为了由安全核心控制通常核心,例如,可以使用上述的复位信号,但作为另一种方法,可以使用 CPU 的 NMI (不可屏蔽中断)。

[0153] 图 18 是在第二实施例中,具有密钥生成机制的处理器的配置框图。除了图 12 所示的配置以外,图 18 的处理器还包括密钥生成机制 34。

[0154] 图 19 是在第二实施例中,由安全核心生成密钥以及使用所生成的密钥的加密处理的说明图。在该图中,处理器中的安全核心 31 使用 CPU 专用密钥 15 和密钥生成机制 34,生成公钥 Ke、N 和私钥 Kd35。例如,其经由通常核心 32 将公钥 Ke 和 N 通知到处理器外部。

在此情况下,不将私钥 Kd 递送到通常核心 32 侧,如前所述,通常核心 32 无法执行除了公钥之外的任何密钥处理。

[0155] 如果已经使用公钥和原文 P 进行了加密的加密语句 C 被从外部输入到通常核心 32,则通常核心 32 向安全核心 31 请求解密处理,因为通常核心 32 并不持有私钥 Kd。随后,安全核心 31 使用私钥 Kd 将文本 P 解密。

[0156] 接下来说明本发明的第三实施例。图 20 是第三实施例中的处理器的基本配置框图。该图中的处理器 40 包括执行单元 41、加载 / 存储单元 42、加密电路 43 和解密电路 44。另外,加载 / 存储单元 42 包括缓存 45 和存储器管理单元 46。

[0157] 和第一及第二实施例一样,第三实施例是基本上执行安全操作的处理器。和第一实施例中的加密处理模块一样,其基本特征在于,用于存储的加密密钥和用于加载的解密密钥在处理器 40 中从执行单元 41 指定,用于执行对所存储的数据的加密操作的加密电路 43,以及用于对包括所取得的命令在内的所加载的数据进行解密的解密电路 44。

[0158] 在图 20 所示的第三实施例中,将明文作为命令和所存储的数据,从执行单元 41 给出到加载 / 存储单元 42,并将加载数据作为明文从加载 / 存储单元 42 给出到执行单元 41。如图 2 所说明的那样,经由加载 / 存储单元 42 将命令给出到主存储器或辅助存储器,并将所存储的数据作为明文给出到加密电路 43,然后作为加密的存储数据输出到主存储器。或者,从主存储器输入的加密加载数据被解密电路 44 所解密,以作为明文的加载数据给出到加载 / 存储单元 42。

[0159] 图 21 是在第三实施例中,具有存储加密密钥和解密密钥的密钥表存储器的处理器的配置框图。在该图中,密钥表存储器 47 存储用于对所存储的数据进行加密的加密密钥,并且密钥表存储器 48 存储用于对所加载的数据进行解密的解密密钥。从执行单元 41,用于存储的密钥号指令和加密密钥的更新 (renewal) 指令被给出到密钥表存储器 47,用于加载的密钥号指令和解密密钥的更新指令被给出到密钥表存储器 48。后面将对密钥表存储器的配置进行详细描述。

[0160] 图 22 是在第三实施例中,具有密钥表存储器的处理器的配置框图,所述密钥表存储器存储用于取命令的解密密钥,以便对要取得的指令执行解密。在该图中,执行单元 41 在命令访问状态下执行处理,以取得例如存储在主存储器中的命令。例如,命令取得数据作为来自主存储器的加载数据被给出到解密电路 44,并且在此情况下,执行单元 41 将命令访问状态标志给出到密钥表存储器 48。解密电路 44 使用解密密钥执行对命令取得数据的解密,以用于来自密钥表存储器 48 的命令取得输出,并经由加载 / 存储单元 42 将所述命令取得数据作为明文给出到执行单元 41。如果需要,则从执行单元 41,将用于命令取得的解密密钥的更新指令给出到密钥表存储器 48。

[0161] 图 23 是在第三实施例中,具有密钥选择寄存器的处理器的配置框图,所述密钥选择寄存器给出用于密钥表存储器的密钥号指令。在该图中,安装了密钥选择寄存器 51 以在存储用于存储的加密密钥的密钥表存储器 47 与执行单元 41 之间,将用于存储的密钥号指令给出到密钥表存储器 47,并且安装了密钥选择寄存器 52 以在存储用于加载的加密密钥的密钥表存储器 48 与执行单元 41 之间,将用于加载的密钥号指令给出到密钥表存储器 48。用于存储的密钥选择寄存器的更新指令从执行单元 41 给出到密钥选择寄存器 51,用于加载的密钥选择寄存器的更新指令从执行单元 41 给出到密钥选择寄存器 52。

[0162] 与在图 21 中,响应于来自执行单元 41 的各个执行命令而输出密钥号指令相反,在图 23 中在命令的某个间隔处给出寄存器更新指令,并使用相同的密钥执行加密 / 解密直到给出下一更新指令为止。另外,可以既安装用于将密钥号指令从执行单元给出到密钥表存储器的直接路线,又安装经由密钥选择寄存器的间接路线,使得将用于给出关于执行单元 41 响应于执行命令而应使用哪条路线的指令的信号给出到密钥表存储器。

[0163] 图 24 是在第三实施例中,具有与执行单元的命令访问整体相对应的密钥选择寄存器的处理器的配置框图。在该图中,和图 22 中一样,执行单元 41 处于例如从主存储器中取得命令的命令访问状态,并且将命令访问状态标志从执行单元 41 给出到密钥选择寄存器 52,密钥选择寄存器 52 给出与存储用于命令取得的解密密钥的密钥表存储器 48 相对应的用于命令取得的密钥号指令。

[0164] 图 25 是在第三实施例中的密钥表存储器的配置示例的说明图。在该图中,相应的加密密钥及其属性被存储在密钥表存储器中。执行单元 41 将密钥号指令直接或经由密钥选择寄存器给出到密钥表存储器,所述密钥号被用作读取地址。并且加密密钥或解密密钥被与用于加密电路 43 或解密电路 44 的加密方法的规范信息、或者是指示出加密必要性的属性数据一起给出。从执行单元 41 给出的密钥更新号指令被用作写入地址,并写入密钥更新数据。

[0165] 取决于加密方法,每个条目的属性数据指示出条目的有效 / 无效、加密的开 / 关,和加密方法及加密模式,以及加密密钥。如后文所述,指示加密开 / 关的数据对应于在加载和存储明文数据而不执行加密 / 解密时的指令。

[0166] 图 26 是在第三实施例中的加密电路或解密电路的配置示例的说明图。例如,图 20 中的解密电路 44 基本上配置有解密流水线 (pipe) 55 和总线仲裁器 57。解密流水线 55 响应于从执行单元 51 经由命令缓冲器 59 对命令信息的输入而进行操作。解密流水线 55 是 N 状态流水线,用于将从例如主存储器经由总线输入的加密数据解密成明文数据。此 N 级流水线是通过将 N 级的处理 56 连接而形成的,所述处理 56 是一步共用密钥加密处理的示意性示例。然后,从解密流水线 55 输出的明文数据被经由总线仲裁器 57,存储在例如图 20 所示的缓存 45 中。

[0167] 加密电路 43 基本上包括加密流水线 60 和总线仲裁器 61。例如,从缓存 45 将 32 比特明文数据给出到加密流水线 60,并且由 N 级流水线使用从执行单元 41 指定的加密密钥而加密的加密数据被输出到总线上,该总线例如经由总线仲裁器 61 连接到主存储器。和解密流水线 55 的情况一样,加密流水线 60 的操作由执行单元 41 经由命令缓冲器 59 给出的命令信息来控制。另外,加密流水线 60 每级的处理的基本结构与解密流水线 55 的相同。而且,多种加密方法包括 AES128,DES 和 SC2000 都可作为加密方法。对于 AES 方法,已经调整了 AES192 和 AES256 规范。

[0168] 例如,在本发明中,总线仲裁器 61 对连接到主存储器或辅助存储器的总线执行仲裁,并且基本上与安全处理器的操作无关。

[0169] 图 27 是示出加密电路和解密电路的配置的框图,其中留下数据的一部分作为明文数据而非对全部数据执行加密,例如,添加数据传递功能以使得数据在主存储器之间输入和输出。在该图中,加密电路和解密电路的基本配置与图 26 中的相同。然而,在加密电路侧,从缓存 45 给出的明文数据中无需加密的数据被直接给出到旁路选择器 63 而不经过

加密流水线 60，并与来自加密流水线 60 的加密数据输出一起存储在多个旁路缓冲器 64 的任意之一中，然后经由总线仲裁器 61 给出到连接到主存储器的总线。

[0170] 旁路选择器 63 对明文数据或加密数据的选择也由来自执行单元 41 的命令信息经由命令缓冲器 59 而控制。由于由加密流水线 60 执行处理要花费时间，因此可以通过使用旁路选择器 63 而进行的控制，使无需加密的明文数据超过要被给出到主存储器侧的加密数据。图 27 中加密所必需的密钥被经由密钥寄存器 69 给出到加密流水线 60。

[0171] 例如，在从连接到主存储器的总线承载的数据当中，未被加密的明文数据被直接给出到旁路选择器 66 而不经过解密流水线 55，使用旁路选择器 66 与已被解密流水线 55 解密的明文数据一起存储在多个旁路缓冲器 67 之一中，然后经由总线仲裁器 57 输出到缓存 45。

[0172] 图 28 是在第三实施例中，用于直写 (write-through) 缓存系统的读取修改写入系统的说明图。如果缓存 45 使用直写系统，则若在存储数据期间发生缓存错误则数据不被存储在缓存 45 中，并且数据被直接存储在主存储器中。如果要存储的数据的大小小于 1 字节，则在主存储器中存储 1 字节数据。然而，在第三实施例中，所存储的数据基本上是起初由加密电路 43 加密，然后存储在主存储器中。在加密进程中，需要一定量的数据作为所存储的数据，使得即使加密 1 字节数据并将之存储在主存储器中，也难以执行正确的解密。

[0173] 例如，在图 28 的加载 / 存储单元加载 / 存储单元 42 中，如果必须在主存储器中存储 1 字节数据，则从主存储器中加载加密进程所必需的长度的数据，并与要存储的 1 字节数据合并。于是，合并后的数据经历读取修改写入操作，以使得合并后的数据被加密并存储在主存储器中。

[0174] 即，例如，如果 (1) 处的缓存存储命令，即 1 字节数据应被存储在缓存中这样一个命令在 (2) 处被确定为缓存错误，则在 (3) 从缓存 45 向主存储器发出作为命令的加载，在 (4) 经由解密电路 44 而来的明文加载数据被存储在读取修改写入 (RMW) 缓冲器 71 中，在 (5) 将要在 (5) 存储的数据给出到 RMW 缓冲器 71，在 (6) 将要存储的数据和所加载的数据合并，并将合并后的数据给出到加密电路 43，并在 (7) 将存储作为命令发出以用于主存储器。

[0175] 接下来将说明本发明的第四实施例。第四实施例和第三实施例之间的区别如下。在第三实施例中，加密电路中使用的加密密钥和解密电路中使用的解密密钥的密钥号，例如密钥号是由执行单元 41 指定的，与之相反的是，在由执行单元 41 执行命令时，由执行单元 41 指定要存储或加载的数据的访问地址，并基于所述地址来选择加密密钥或解密密钥。

[0176] 图 29 是第四实施例中的处理器的基本配置框图。图中所示的处理器 40 除了执行单元 41、加密电路 43 和解密电路 44 之外，还包括密钥表存储器 73，该密钥表存储器 73 响应于执行单元 41 所给出的地址，将用于存储的加密密钥给出到加密电路 43，并将用于加载的解密密钥给出到解密电路 44。

[0177] 图 30 是处理器的配置框图，其中响应于由执行单元指定的存储数据或加载数据的逻辑地址来选择密钥。与图 29 所示的处理器不同，本图中的处理器 40 包括存储用于存储的加密密钥的密钥表存储器 74 和存储用于加载的解密密钥的密钥表存储器 75，以及配备了缓存和存储器管理单元 46 的加载 / 存储单元 42，这与图 20 的处理器中相同。从执行单元 41 给出到加载 / 存储单元 42 的地址，即存储数据或加载数据的地址是逻辑地址，然后

将这些逻辑地址给出到密钥表存储器 74 或 75, 以选择用于存储的加密密钥或用于加载的解密密钥。然后将所选择的密钥给出到加密电路 43 或解密电路 44。从执行单元 41, 将用于存储的加密密钥的更新指令给出到密钥表存储器 74, 并将用于加载的解密密钥的更新指令给出到密钥表存储器 75。

[0178] 图 31 是在第四实施例中, 响应于数据的物理地址而选择了密钥的处理器的配置框图。与图 30 所示的处理器相比, 从加载 / 存储单元 42 分别给出存储数据的物理地址或加载数据的物理地址以用于密钥表存储器 74 或 75, 将用于存储的加密密钥给出到加密电路 43, 并将用于加载的解密密钥给出到解密电路 44。

[0179] 图 32 是第四实施例中的密钥表存储器的配置图。与第三实施例的图 25 相比, 如果从执行单元侧给出从第 0 比特到第 31 比特的 32 个比特作为数据的访问地址, 则用这些地址作为读取地址来选择所存储的加密密钥, 并将所述加密密钥与加密属性一起给出到加密电路 43 或解密电路 44。如果对作为存储器读取地址的每 4k 字节使用不同的密钥, 则从第 12 比特到第 31 比特的地址被用来选择加密密钥。在此情况下, 后面将会提到, 4k 字节对应于主存储器中 1 页的大小。如果将 4k 字节称为用于加密的地址单元, 则密钥表存储器的条目数据包括排除了总条目数与地址单元之积的字节之后的地址标签。例如, 如果条目的总数是 32(5 比特), 则从第 17 比特到第 31 比特的地址成为地址标签。

[0180] 图 33 是在第三实施例中, 具有多路配置的密钥表存储器的说明图。该图中的密钥表存储器由从密钥表 1 到密钥表 4 的多个表组成, 并选择存储在与从执行单元侧给出的访问地址相对应的四个表中之一的密钥和加密属性, 以给出到加密电路 43 或解密电路 44。

[0181] 图 34 是使用联合存储器系统的密钥表存储器的配置示例的说明图。在该图中, 在与比较选择器 77 所存储的加密密钥相对应的目标地址的范围内, 将访问地址 32 比特分类到任意之一中, 选择对应于所分类范围的加密密钥, 以与加密属性一起给出到加密电路 43 和解密电路 44。在图 34 中, 通过排除地址单元量而不管条目总数而得到的地址标签被包括在条目中。如果地址单元是 4k 字节, 则从第 12 比特到第 31 比特的地址成为地址标签。

[0182] 图 35 是在第四实施例中, 响应于数据的逻辑地址或者物理地址而选择密钥的处理器的配置框图。在该图中, 数据的逻辑地址从执行单元 41, 或者物理地址从加载 / 存储单元 42 分别给出到密钥表存储器 74 或 75。或者, 从执行单元 41 将用于所存储数据的逻辑地址和物理地址的选择指令给出到密钥表存储器 74。并且, 从执行单元 41 将用于所加载数据的逻辑地址和物理地址的选择指令给出到密钥表存储器 75。响应于这些选择指令, 选择与这些逻辑地址或物理地址中的任意地址相对应的密钥, 并将之分别给出到加密电路 43 和解密电路 44。

[0183] 图 36 是包括密钥选择寄存器的处理器的配置示例, 所述密钥选择寄存器将对逻辑地址和物理地址的选择指令给出到密钥表存储器。与图 35 相比, 在执行单元 41 与密钥表存储器 74 或 75 之间安装了各自的密钥选择寄存器 78 和 79, 并且用于存储数据的逻辑地址和物理地址的选择指令、以及用于加载数据的逻辑地址和物理地址的选择指令被输出到密钥表存储器 74 和 75。从执行单元 41 到密钥选择寄存器 78 和 79, 给出各自的密钥选择寄存器的更新指令。

[0184] 图 37 是图 35 和图 36 中的密钥表存储器的配置示例。在该图中, 密钥表存储器具有物理地址密钥表和逻辑地址密钥表, 并分别响应于物理地址和逻辑地址而输出物理密钥

和逻辑密钥。并且，响应于来自执行单元 41 侧的密钥选择指令，或者来自密钥选择寄存器的选择指令，物理密钥或逻辑密钥被逻辑和物理密钥选择单元 81 与加密属性一起输出到加密电路 43 或解密电路 44。

[0185] 图 38 是在第四实施例中，在加载 / 存储单元 42 内的存储器管理单元 (MMU) 46 中具有作为密钥表的密钥表存储器内容的处理器的配置示例。

[0186] 图 39 和图 40 分别是在所述存储器管理单元和缓存访问系统中的密钥信息的存储格式的说明图。一般地，响应于 MMU46 内部的翻译后备缓冲器 (TLB) 中的物理存储器的每一页，逻辑地址与物理地址之间的对应关系被存储在每个条目中。在图 39 中，对于页的密钥信息被存储在 TLB 的每个条目中。例如，如果数据访问地址是逻辑地址，则选择与所述逻辑地址相对应的条目，并由属性检查 83 检查响应于该条目的数据属性和访问属性，然后将由缓存命令生成 84 生成的命令发送到缓存 45。

[0187] 在缓存 45 侧，响应于接收到的命令的内容来检索标签。在缓存命中的情况下，立即将数据相应返回到执行单元 41 侧，而在缓存缺失的情况下，将对应于所述标签的命令发出到包含加密电路 43 和解密电路 44 的加密 / 解密总线接口。在此情况下，使用从条目中读取的密钥信息和物理地址，例如在将来自主存储器的响应数据解密之后，将所述解密后的数据存储在缓存中，然后将数据响应返回到执行单元 41。

[0188] 图 40 是当在存储器管理单元中安装地址映射寄存器 (AMR) 来代替 TLB 时的密钥信息存储格式的说明图。在该图中，与 TLB 的存储内容相对应的信息被存储在寄存器而非存储器中。例如，页面大小可以是可变的，使得单个条目可以覆盖很大的数据区域。

[0189] 图 41 是处理器的配置示例，其中在加载 / 存储单元中的存储器管理单元 (MMU) 的操作被暂停，即处于关 (OFF) 状态时，将加密密钥从执行单元 41 给出到加密电路 43，并将解密密钥给出到解密电路 44。在该图中，将 MMU 的开 / 关信号给出到加密电路 43 和解密电路 44。如果所述信号关，则加密电路 43 或解密电路 44 使用从执行单元 41 给出的密钥，或者如果所述信号开，则其使用从存储器管理单元 46 内部的 TLB87 或 AMR88 给出的密钥，以便执行加密或解密处理。

[0190] 图 42 是图 41 的加密电路或解密电路中的密钥切换系统的说明图。在该图中，加密电路或解密电路的配置基本上与第三实施例的图 26 中相同，但添加了密钥选择器 90。根据从执行单元给出的 MMU 开 / 关信号的值，在关的时候由密钥选择器 90 选择从执行单元给出的密钥，而在开的时候由密钥选择器 90 选择从 TLB 或 AMR 给出的密钥，然后将所选择的密钥给出到加密流水线 60 或解密流水线 55。

[0191] 图 43 是第三或第四实施例中的执行单元的 I/O 信号的说明图。在第三实施例的图 20 中，必需的信号包括作为输出信号的加载加密密钥、存储加密密钥、存储数据和命令，以及作为输入信号的加载数据 (圆形记号)，并且访问地址和加载 / 存储状态信号是根据结构而存在的信号 (三角形记号)。

[0192] 在图 21 中，加载密钥号指令而非加载解密密钥的输出信号，以及存储密钥号指令的输出信号是必需的。另外，由于对于执行单元，对密钥表存储器的更新所具有的值与寄存器访问相等，因此寄存器相关的 I/O 信号也变得必要。

[0193] 在图 22 中，与命令访问状态相对应的 I/O 信号是必要的，并且作为输出信号的执行状态信号以及作为输入信号的命令取得数据被认为是必需的。

[0194] 在图 23 和图 24 中,除了图 20 和图 21 中所使用的以外,还添加了密钥选择寄存器,寄存器相关的 I/O 信号也是必需的。

[0195] 对以下说明进行了简化,仅说明特征性的部分。将图 21、图 22 和图 23 合计起来,除了在合并这三种情况时的 I/O 信号以外,还添加了指示要执行的进程对应于哪些管理员和用户的管理员 / 用户状态信号,以及上下文,即进程 ID(标识符)的数据。在第三实施例中,除了从执行单元输出的密钥号指令信号以外,这些管理员 / 用户状态信号和上下文 ID 数据也用于选择加密密钥和解密密钥。

[0196] 图 29 以后的对应于第四实施例。对数据的访问地址成为必需的输出信号。此外,用于选择逻辑地址或物理地址的密钥选择指令信号也在图 35 和图 36 中输出。

[0197] 在图 38 中,因为在存储器管理单元的 TLB 中添加了密钥表,所以从配置方面来看,存在寄存器相关的信号。另外,图中示出了添加了管理员 / 用户状态信号和上下文 ID 数据时的情况。在第三实施例中,这些添加的信号用于选择加密密钥和解密密钥,以及访问地址。

[0198] 图 41 包括响应于指示存储器管理单元 (MMU) 的开 / 关的状态信号的值而使用来自执行单元的密钥输出的情况,以及使用来自 TLB 的密钥输出的情况。结果,所有的 I/O 信号以及管理员 / 用户状态信号和上下文 ID 数据变为必需的。

[0199] 如上所述,在第三和第四实施例中,从执行单元指定密钥以用于数据和指令代码的加密 / 解密,使得加密处理可以在对应于要执行的命令的级别上执行。而且,由于加密 / 解密密钥由密钥选择寄存器或访问地址来指定,因此可以对每个程序单元或每次访问执行加密处理,从而,可以响应于多种条件而选择要执行的处理。

[0200] 接下来将说明本发明的第五实施例。第五实施例示范了一种更精确的配置,以便实现作为第一实施例而提出的安全处理器的安全操作。下面将说明与其配置相对应的验证密钥的设定和诸如进程验证之类的操作,以便进一步提高进程 (程序) 的可靠性。

[0201] 图 44 是用于说明第五实施例的处理器中所必需的功能性配置图。在该图中,处理器 100 连接到物理存储器 101 (例如图 2 中的主存储器 17)、I/O 设备 102 (例如辅助存储器 18)。

[0202] 处理器 100 包括:存储器访问控制单元 105,其控制对物理存储器 101 和 I/O 设备 102 的访问;命令解释单元 106,其对要执行的命令进行解释;验证单元 107,其对存储执行代码的页面执行验证;加密 / 解密和签名生成 / 核实单元 108,其例如执行对已被验证的页面的加密 / 解密;安全上下文标识符生成单元 109,其在生成进程时生成与进程 (即上下文) 相对应的安全上下文标识符;安全上下文标识符消除单元 110,其在消除进程时消除相应的标识符;处理器专用密钥 111,其用于加密;经验证信息主要存储单元 112,其存储与例如存储在物理存储器 101 中的物理页面相对应的经验证信息;以及安全 DMA113,其用于在访问存储器时进行直接存储器访问。

[0203] 处理器 100 的内部包括图 39 中所说明的翻译后备缓冲器 (TLB) 114,以及上下文信息存储单元 115。TLB114 存储页面表条目 (PTE) 122,所述 PTE122 例如参照物理页面,指示出逻辑地址和物理地址之间的对应关系。上下文信息存储单元 115 包括程序计数器 (用于保持程序计数器的值的计数器) 117、存储安全上下文标识符的安全上下文标识符寄存器安全上下文标识符寄存器 118、存储验证所需的密钥的验证密钥寄存器验证密钥寄存器 119,

以及寄存器组 120。

[0204] 而且,物理存储器 101 例如存储每个物理页面单元 124 的执行代码,I/O 设备 102 存储执行代码和数据,其以为每个页面单元 125 添加了验证信息 126 的格式存储。在第七实施例中,使用存储在验证密钥寄存器 119 中的验证密钥,执行对其安全上下文标识符存储在安全上下文标识符寄存器 118 中的上下文(进程)的执行代码的验证。

[0205] 图 45 是一个方法的说明图,所述方法用于在例如用户所使用的程序之类的在处理器上运行的程序被激活并且上下文生成命令被发出时,生成对应于上下文的安全上下文标识符。上下文生成命令被给予到处理器中的命令解释单元 106,由安全上下文标识符生成单元 109 响应于解释的结果而生成安全上下文标识符,并且其值被设定在安全上下文标识符寄存器 118 中。在第五实施例中,系统被配置以使得仅可通过该方法将所述值设定在安全上下文标识符寄存器 118 中。因此,不可能篡改安全上下文标识符以伪装另一上下文。所述上下文是基本上基于面向对象编程的概念。更一般而言,其对应于进程的执行状态,即程序的执行状态。在第五实施例中,使用术语“上下文”来代替进程。

[0206] 图 46 是生成安全上下文标识符的实际方法的说明图。如图中所示的随机数生成器 127 或单调递增计数器单调递增计数器 128 可用于其生成。相同的值被作为随机数而生成的概率不是零,在初始的一个周期之后,计数器值呈现相同的值。因此,无法严格保证标识符的唯一性。然而,可以通过使用具有足够长的比特长度的安全上下文标识符来基本避免这种问题的发生。

[0207] 或者如图所示,可以在接合单元(joint unit)129 处将处理器现有的上下文 ID 与随机数生成器 127 的输出或单调递增计数器 128 的输出合并,以生成安全上下文标识符。现有的上下文 ID 是由 OS 建立的任意值,其唯一性一般是不受保证的。

[0208] 图 47 是消除安全上下文标识符的方法的说明图。在该图中,当在处理器上允许的程序发出上下文消除命令时,处理器使安全上下文标识符寄存器 118 的内容无效。可以通过以下方式实现无效化的方法:通过清 0,或者在寄存器中为表示有效/无效的标志给出一个存储区域,然后可将该标志设定为无效。

[0209] 图 48 是图 44 中的验证的说明图,例如对每个页面单元,要将信息 126 添加到执行代码 125 上。在处理器内的上下文信息存储单元 115 中的验证密钥寄存器 119 中,存储了用于使用验证信息 126 而进行的执行代码 125 的验证处理的密钥。例如,如果利用 RSA 的电子签名被用作验证信息,RSA 公钥被用作验证密钥,但如果 SHA(安全哈希算法)-1HMAC(基于哈希的消息验证代码)被用作共享密钥系统,则验证密钥是 20 字节的值。

[0210] 当由 OS 生成上下文时,即在对上下文信息进行初始化的阶段,将密钥存储在验证密钥寄存器 119 中,同时检查验证密钥的合法性。如果验证密钥本身是由敌对者生成的,并且使用此密钥生成了对应于恶意执行代码的验证信息,则验证处理本身会毫无问题地成功,并且处理器的验证功能被禁止。因此,如何保证验证密钥的合法性是重要的关心内容。

[0211] 图 49 是在验证密钥是公钥的情况下,验证密钥寄存器中的密钥设定系统的说明图。在该图中,假定所使用的验证密钥是 RSA 公钥,并且作为认证权威机构 134 的公钥的认证权威机构的证书例如是在从工厂发货时就嵌入在处理器中的,因此随后的替换和修改被禁止。要设定在验证密钥寄存器 119 中的验证密钥由认证权威机构私钥以签名的形式给出。例如,当生成上下文时,将验证密钥设定命令给出到处理器,其命令由命令解释单元 106

解释，并在验证密钥由签名校核实单元 108 核实之后，验证密钥被存储在验证密钥寄存器 119 中。结果，在验证密钥寄存器中仅建立了由认证权威机构签名的公钥。

[0212] 图 50 是图 49 中的验证密钥建立处理的流程图。在该图中，起初在步骤 S71 由命令解释单元 106 取得验证密钥建立命令，在步骤 S72，由签名校核实单元 108 使用签认证权威机构的签名和公钥来核实时所取得的公钥。在步骤 S73 确定核实是否成功。如果成功，则将包括在命令解释单元 106 所取得的建立命令中的公钥存储在验证密钥寄存器 119 中，并且处理终止。而如果核实失败，则处理立即终止。

[0213] 图 51 示出了在验证密钥是共享密钥的情况下密钥建立系统，图 52 是密钥建立处理的流程图。如果共享密钥被用作验证密钥，则必须从将验证信息添加到执行代码上的一侧通过安全方法接收验证密钥。在此情况下，假定已使用 RSA 公钥加密的 HMAC 密钥被与验证密钥建立命令一起接收，然后在处理器侧被使用解密单元 108 用处理器专用 RSA 私钥 137 解密，并存储在验证密钥寄存器 119 中。

[0214] 在图 52 所示的流程图中，在步骤 S76 由命令解释单元 106 取得验证密钥设定命令，在步骤 S77，由解密单元 108 使用处理器专用 RSA 私钥 137 将包括在命令中的加密 HMAC 密钥解密，在步骤 S78，由命令解释单元 106 将解密后的 HMAC 密钥存储在验证密钥寄存器 119 中，然后处理终止。

[0215] 图 53 是一个页面调入系统的说明图，其中其安全上下文标识符已被生成的上下文的执行代码被存储在主存储器中，即存储在物理存储器 101 的物理页面中，并且物理页面被验证以能够开始执行处理。图 54 是页面调入处理的流程图。

[0216] 在此页面调入处理中，起初由 OS 将执行代码存储在物理页面中，并设定页面表条目 (PTE) 中的各种数据，所述 OS 请求验证单元 107 建立安全页面标志字段，作为对上下文，即物理页面的验证请求。物理页面由验证单元 107 响应于请求而验证，并建立安全页面标志字段的标志，以使得允许使用 PTE。

[0217] 在图 54 的处理流程图中一旦处理开始，则在步骤 S80，由 OS 将执行代码存储在空闲的物理页面中，在步骤 S81，由 OS 将物理页面的顶地址和相应的逻辑页面的顶地址设定为 TLB 中的 PTE 的物理地址和逻辑地址，并且在步骤 S82，在 PTE 中设定安全上下文标识符的值。例如，如图 45 和图 46 中所说明的那样，在发出上下文生成命令时所生成的安全上下文标识符和存储在安全上下文标识符寄存器 118 中的安全上下文标识符可由 OS 读取的假定之下，OS 将所读取的安全上下文标识符建立在 PTE 中。

[0218] 随后，如果需要，在步骤 S83 由 OS 在 PTE 中设定用于该页面的读取 / 写入属性，并且在步骤 S84，OS 请求作为硬件的验证单元 107 建立安全页面标志字段。主要假定 OS 本身已被验证，并且设定安全页面标志字段基本上是 OS 任务，但这里，经验证的 OS 请求硬件建立标志。

[0219] 在步骤 S85，在验证单元 107 处执行验证处理。后面将会描述该处理的细节。在该处理中，使用对应于安全上下文标识符的验证密钥，以及存储在经验证信息主要存储单元 112 中的验证信息，来验证物理页面。然后，在步骤 S86 确定验证是否成功。如果成功，则安全页面标志字段的标志被建立，并且其 PTE 变为可用。相反，如果失败，则安全页面标志字段的标志被复位，并且其 PTE 被禁止使用，并在步骤 S89 由 OS 执行恢复或错误处理。

[0220] 图 54 所示处理的主要目标是可以直接为 TLB 中的 PTE 设定值的处理器。然而，例

如,在为主存储器上的 PTE 设定值,并且 TLP 充当缓存的处理器中,从步骤 S80 到步骤 S83 的处理被执行用于主存储器上的 PTE,并且在内容被存储在 TLB 中的缓存中之后,执行步骤 S84 之后的操作。

[0221] 图 55 是图 53 所示的验证单元 107 的配置示例,图 56 是图 54 所示的步骤 S85 处的验证处理的流程图。在此情况下,为整个页面计算 SHA-1 值,并将其与电子签名的解密结果相比较。如图 56 所示,自然可以将验证单元 107 的操作实现为由软件而非硬件进行的处理。

[0222] 在图 55 中,以被给予 SHA-1 哈希运算器 140 的 64 字节的部分来划分物理页面 125,其中计算整个页面上的哈希值,并将其给予到比较器 142。相反,存储在经验证信息主要存储单元 112 中的 RSA 电子签名被与存储在验证密钥寄存器 119 中的 RSA 公钥一起给予到 RSA 解密设备 141,并且作为其输出的解密哈希值被比较器 142 与 SHA-1 哈希运算器 140 的输出相比较。如果它们彼此匹配则确定验证成功,而如果它们彼此不匹配则确定失败。

[0223] 在图 56 所示的验证处理中,在步骤 S90 以 64 字节为单位读取物理页面,在步骤 S91 执行哈希运算,并在步骤 S92 确定是否到达页面末端。如果未到达页面末端,则重复步骤 S90 及其以后的处理。

[0224] 如果达到末端,则在步骤 S93,电子签名使用 RSA 公钥继续解密处理,并且在步骤 S94,将解密结果与哈希运算的结果相比较。如果它们彼此匹配,则在步骤 S95 建立安全页面标志字段,而如果它们彼此不匹配,则在步骤 S96 将安全页面标志字段复位,以结束处理。

[0225] 如上所述,在第五实施例中,在将执行代码页面调入到物理存储器(主存储器)中时验证执行代码,并建立指示验证成功的安全页面标志。

[0226] 接着,将参照第六实施例来说明在本发明中,执行物理页面上的命令时的存储器访问控制。图 57 是在执行物理页面上的命令时的存储器访问控制系统的说明图。在该图中,如果在安全上下文标识符寄存器 118 中存在有意义的值,并且如果建立了 PTE122 的安全页面标志字段,并且如果存储在安全上下文标识符寄存器 118 中的标识符的值与 PTE122 上的上下文标识符的值彼此匹配,则允许执行物理页面 124 上的命令。此控制由存储器访问控制单元 105 来执行。在此情况下,对数据读取 / 写入属性的检查和与物理页面相对应的管理员属性与本发明的内容并不直接相关,是单独执行的。

[0227] 图 58 是存储器访问控制单元 105 的操作示例的说明图。在该图中,粗点划线所包围的内容对应于存储器访问控制单元 105。另外,在 TLB114 中包括了安全页面标志字段和安全上下文标识符,作为 PTE 的属性数据。

[0228] 和图 39 中一样,如果用逻辑地址作为访问地址来进行访问,则通过此地址选择的 PTE 的属性数据被读取,并由属性检查 146 与访问属性相比较。如果检查结果是 OK,则使用对应于逻辑地址而读取的物理地址和属性检查结果来生成缓存命令 147,并且例如检索缓存 45 中的标签 148,在缓存命中的情况下直接返回数据响应。而例如在缓存缺失的情况下,从主存储器加载的数据经由队列和总线接口 149 存储在缓存中,并将数据响应返回到执行单元。

[0229] 图 59 是在取得命令时存储器访问控制单元 105 的处理流程图。如果由图 57 所示的命令执行单元 144 输出用于命令取得的逻辑地址,则在步骤 S98 选择对应于所指定的逻

辑地址的 PTE 的属性数据。并且在步骤 S99, 检查当前上下文, 即当前要执行的上下文是不是安全上下文, 即是否存在具有有效的安全上下文标识符的上下文。如果当前上下文是安全上下文, 则在步骤 S100, 检查是否建立了对应于所述上下文的 PTE 的安全页面标志字段 (SPF)。如果建立了, 则确定当前上下文的安全上下文标识符, 即存储在安全上下文标识符寄存器 118 中的标识符是否与存储在 PTE 中的安全上下文匹配。

[0230] 如果匹配, 则在步骤 S102, 检查作为对应于上下文的页面属性的读取 / 写入属性和管理员属性。如果 OK, 则在步骤 S103 生成缓存命令, 用于将物理地址输出到缓存以用于命令取得, 以结束处理。

[0231] 如果在步骤 S99, 当前上下文不具有有效的安全上下文标识符, 则在步骤 S104, 确定是否建立了相应 PTE 中的安全页面标志字段 (SPF); 如果未建立, 则没有建立安全上下文标识符, 应该处理与未经验证的通常情况相同的执行代码, 并转到步骤 S102 处的处理。如果在步骤 S100 和 S102 处的判断结果是“否”, 则将会在步骤 S105 被视为错误处理, 以终止进程。必须进行以下两个进程: 将逻辑地址分离成逻辑页面的顶地址和页面中的偏移值的进程, 以及将物理页面的顶地址和其偏移值相加的进程, 但这些进程与本发明并不直接相关, 将省略其说明。

[0232] 图 60 是在具有安全核心和通常核心的处理器中的存储器访问控制系统的说明图。在该图中, 通常核心 152 仅执行传统的进程, 所述传统进程与如图 12 所示的加密处理模块 12 和代码验证处理模块 13 所进行的进程无关, 而安全核心 151 可以执行包括代码验证处理模块的验证控制在内的安全操作, 所述代码验证处理模块的验证控制包括图 44 中未说明的加密处理模块所进行的操作的控制。

[0233] 在图 60 中, 利用存储器访问控制单元 105 的控制, 安全核心 151 被允许使用对应于其中已建立了安全页面标志字段的 PTE 的物理页面, 但受到控制以使通常核心无法使用所述页面。

[0234] 在图 44 中, 安全核心所控制的代码验证处理模块包括存储器访问控制单元 105, 所述存储器访问控制单元 105 对应于验证单元 107、签名生成 / 核实单元 108、安全上下文标识符生成单元 109、安全上下文标识符消除单元 110、处理器专用密钥 111、经验证信息主要存储单元 112、安全 DMA113、安全上下文标识符寄存器 118、验证密钥寄存器 119, 以及 PTE122 中的安全页面标志字段和安全上下文标识符。

[0235] 在图 60 中, 基本上安全核心 151 仅执行在下述物理页面中的执行代码, 在所述物理页面中, 验证处理已完成并且在 PTE 中建立了安全页面标志字段, 并且通常核心 152 仅执行未经验证的常规代码。然而, 也可以是下述情况, 即通常核心可被配置以使得通常核心除了常规代码以外还可执行经验证代码。

[0236] 图 61 是具有在安全模式和通常模式之间切换的核心的处理器的配置框图。在该图中, 模式寄存器 155 安装在核心 154 中, 并且其中建立了安全页面标志字段的页面仅在安全模式情况下才可使用。安全模式和通常模式之间的切换可以通过使用中断作为触发的方法来执行, 例如在通常用户模式和内核模式之间切换的情况, 但也可使用其他方法。

[0237] 图 62 是使用图 44 中的安全 DMA113 对物理存储器 101 的页面数据传输系统的说明图, 所述系统作为存储器访问控制系统。图 63 是使用安全 DMA113 的数据传输处理的流程图。例如, 在图 53 和图 54 的页面调入处理中, 作为执行代码的页面数据起初存储在物理

页面中，然后对这些执行代码进行验证。然而，由于验证处理需要诸如哈希值计算之类的处理，因此在该实施例中，在页面数据的每个传输单位处计算哈希值，并保持计算的结果作为哈希运算的中间结果。重复这些处理，并且哈希运算在传输末尾完成，使得结果可用在随后的验证处理中。

[0238] 图 62 中的安全 DMA113 包括来自核心 154 的数据的传输源地址、传输目的地地址、接收传输大小的传输管理单元 157、从 I/O 设备 102 读取数据的数据读取器 158、用于执行哈希运算的哈希运算器 159、在物理存储器 101 中写入页面数据的数据写入器 160，以及物理页面顶地址保持单元 161，其保持物理页面的顶地址和用于页面的哈希值。

[0239] 当在图 63 中处理开始时，进行以下过程。使用在核心 154 上运行的程序，一般是由从 OS 接收诸如传输源地址之类指令的传输管理单元 157 来进行。数据读取器 158 被指示从 I/O 设备 102 中读取接下来的 64 字节数据，在步骤 S111，这 64 字节数据被数据读取器 158 读取。在步骤 S112，哈希运算器 159 被传输管理单元 157 指示执行哈希运算，在步骤 S113 由哈希运算器 159 执行哈希运算。保持中间结果，并且在步骤 S114，数据写入器 160 被传输管理单元 157 指示将 64 字节数据写入物理存储器 101，在步骤 S115 由数据写入器 160 将这 64 字节数据写入物理存储器 101 中。在步骤 S116，确定一个页面的数据传输是否完成。如果未完成，则重复从步骤 S110 开始的处理，而如果完成了，则由传输管理单元 157 将作为传输目的地地址的由哈希值和物理页面顶地址所构成的对给出到保持单元 161，以结束处理。

[0240] 图 64 是在执行代码，包括存储器访问控制时的处理流程图。该图是在一般由 OS 进行的页面调入期间的处理流程图，并且本发明的特征点在于粗线所包围的处理。当处理开始时，起初在步骤 S120，传输源 / 目的地地址、传输大小被指示到安全 DMA113，并在步骤 S121 确定传输是否成功。如果成功，则在步骤 S122，在 TLB 内的 PTE 中建立多种信息，像图 54 所示的步骤 S81 到 S83 那样，并在步骤 S123，像在步骤 S84 那样地作出请求以建立安全页面标志字段。在验证单元处执行验证处理之后，在步骤 S124 确定是否成功设定了标志。如果成功，则处理结束。如果在步骤 S121 传输失败，或在步骤 S124 设定失败，则处理立即终止。

[0241] 如上所述，根据第六实施例，即使对已被成功验证的执行代码的访问也仅在检查安全上下文标识符和安全页面标志字段之后才被允许。

[0242] 最后，将参照图 65 到图 74 来说明本发明的第七实施例。在第七实施例中，当与上下文开关 (switch) 相对应的上下文信息和 PTE 例如被撤出到主存储器中时，执行用于保护数据的加密或添加篡改检测信息。例如，在第一实施例中，经验证的执行代码被加密然后存储在物理存储器中。相反，在第七实施例中，上下文信息在被存储在物理存储器中之前被加密。

[0243] 图 65 是用于上下文信息的加密方法的说明图。在该图中，由加密设备 165 使用处理器专用密钥 111，对如图 44 中所说明的存储在上下文信息存储单元 115 中的全部上下文信息进行加密，然后将加密上下文信息 166 存储在物理存储器 101 中。

[0244] 图 66 是用于图 65 所示的相应上下文信息的解密方法的说明图。由解密设备 168 在通过上下文开关而变得必要时，使用处理器专用密钥 111 将存储在物理存储器 101 中的加密上下文信息 166 解密，并将解密的上下文信息存储在上下文信息存储单元 115 中。

[0245] 图 67 是将篡改检测信息添加到上下文信息上的系统的说明图。对于该图中存储

在上下文信息存储单元 115 中的上下文信息,由篡改检测信息生成器篡改检测信息生成器 169 使用处理器专用密钥 111 来生成篡改检测信息 170,并将其与上下文信息一起存储在物理存储器 101 中。

[0246] 图 68 是使用图 67 所示的相应的变更检测信息,用于上下文信息的篡改检测系统的说明图。在该图中,使用添加到上下文信息上的篡改检测信息 170,由篡改检测器 172 使用处理器专用密钥 111 来检测篡改。

[0247] 图 69 是在安全操作所需的上下文信息被与常规上下文信息区分开时,在存储于上下文信息存储单元 115 中的上下文信息当中仅将用于安全操作的上下文信息 175 加密的上下文信息加密系统的说明图。根据此方法,尽可能在处理器核心的部分中如前一样地对待常规上下文信息 176,即诸如现有的上下文 ID 之类的上下文信息,而不进行加密,使得仅对存储在验证密钥寄存器 119 和安全上下文标识符寄存器 118 中的内容被加密,作为用于安全操作的上下文信息 175。

[0248] 对于现有的上下文 ID,可以存储相同的值作为安全上下文标识符,作为 OS 的操作。例如,如果 OS 被恶意代码重写,则不能保证两个标识符的值变得一样。如果处理器被配置以使得处理器可以仅在得到相同值时操作,则保持了只有得到相同值处理器才会工作这样的安全性,不再存在问题。

[0249] 在图 69 中,由加密器 / 解密器 174 使用处理器专用密钥 111,仅将用于安全操作的上下文信息 175 加密,使得加密上下文信息 177 存储在物理存储器 101 中,而常规上下文信息 176 仍作为用于直接存储在物理存储器 101 中的明文上下文信息 176。

[0250] 图 70 是将添加到用于安全操作的上下文信息 175 上的篡改检测信息存储在物理存储器 101 中的上下文信息存储系统的说明图。在该图中,由篡改检测信息生成器 / 篡改检测器 179 使用处理器专用密钥 111,来为用于安全操作的上下文信息 175 生成篡改检测信息 180,并将其与用于安全操作的上下文信息 175 和常规上下文信息,即明文上下文信息 176 一起存储在物理存储器 101 中。在此情况下,并未对作为常规上下文信息的程序计数器的值和寄存器组的值应用加密,但为了进一步提高可靠性,自然可以对常规上下文信息应用加密,或者添加篡改检测信息。

[0251] 图 71 到图 74 是用于存储在页面表条目 (PTE) 122 中的内容的保护性系统的说明图。图 71 示出了 PTE 的加密系统。存储在 PTE122 中的内容,即安全页面标志字段、安全上下文标识符、逻辑地址和物理地址的值被加密器 165 使用处理器专用密钥 111 而加密,并将加密 PTE183 存储在物理存储器中的页面表 182 中。

[0252] 图 72 是用于图 71 所示的相应的加密 PTE 的解密系统的说明图。在该图中,存储在物理存储器 101 中的加密 PTE183 被用使用处理器专用密钥 111 的解密器 168 所解密,并作为 PTE 存储在 TLB114 中。

[0253] 图 73 是对 PTE 的篡改检测信息添加系统的说明图,图 74 是用于 PTE 的篡改检测系统的说明图。在图 73 中,由篡改检测信息生成器 169 使用处理器专用密钥 111 来生成用于 PTE122 的篡改检测信息 185,并将其与 PTE122 一起存储在页面表 182 中。

[0254] 在图 74 中,由篡改检测器 172 使用变更检测信息 185 和处理器专用密钥 111,在存储于页面表 182 中的 PTE122 中检测篡改。

[0255] 如上所述,在第七实施例中,为安全处理器所使用的上下文信息和 PTE 执行加密

和篡改检测，因此进一步提高了信息处理的安全性。

[0256] 以上详细描述了本发明的安全处理器和用于安全处理器的程序。可以将此安全处理器用作通用计算机系统的基本组件。图 75 是这种计算机系统，即硬件环境的配置框图。

[0257] 图 75 所示的计算机系统包括中央处理单元 (CPU) 200、只读存储器 (ROM) 201、随机访问存储器 (RAM) 202、通信接口 203、存储器设备 204、I/O 设备 205、移动存储介质读取设备 206，以及连接所有组件的总线 207。

[0258] 作为存储器设备 204，可以使用各种存储器设备，例如硬盘和磁盘。在本发明范围内，程序在图 3 到图 5、图 7、图 9 到图 11 以及其他流程图中示出，并且在权利要求 7、19 和 20 中描述了程序，所述程序被存储在这种存储器设备 204 或 ROM201 中，并且当这些程序被 CPU200 执行时，可以执行本发明的实施例中的安全处理器的操作。加密密钥的设定、代码验证处理和加密处理。

[0259] 这些程序可以由程序提供器 208 经由网络 209 和通信接口 203 存储在存储器设备 204 中，或者它们可以被存储在商业上发行的移动存储器介质中，设定在读取设备 206 中并由 CPU200 执行。作为移动存储器介质 210，可以使用多种类型的存储器介质，例如 CD-ROM、软盘、光盘、光磁盘和 DVD。本发明模式中的安全处理器可以在存储于这种存储器介质中的程序被读取设备 206 所读取时操作。

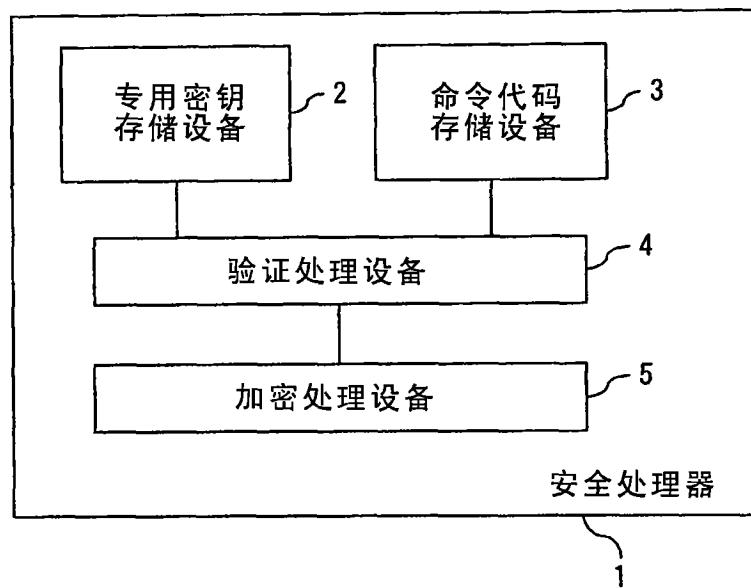


图 1

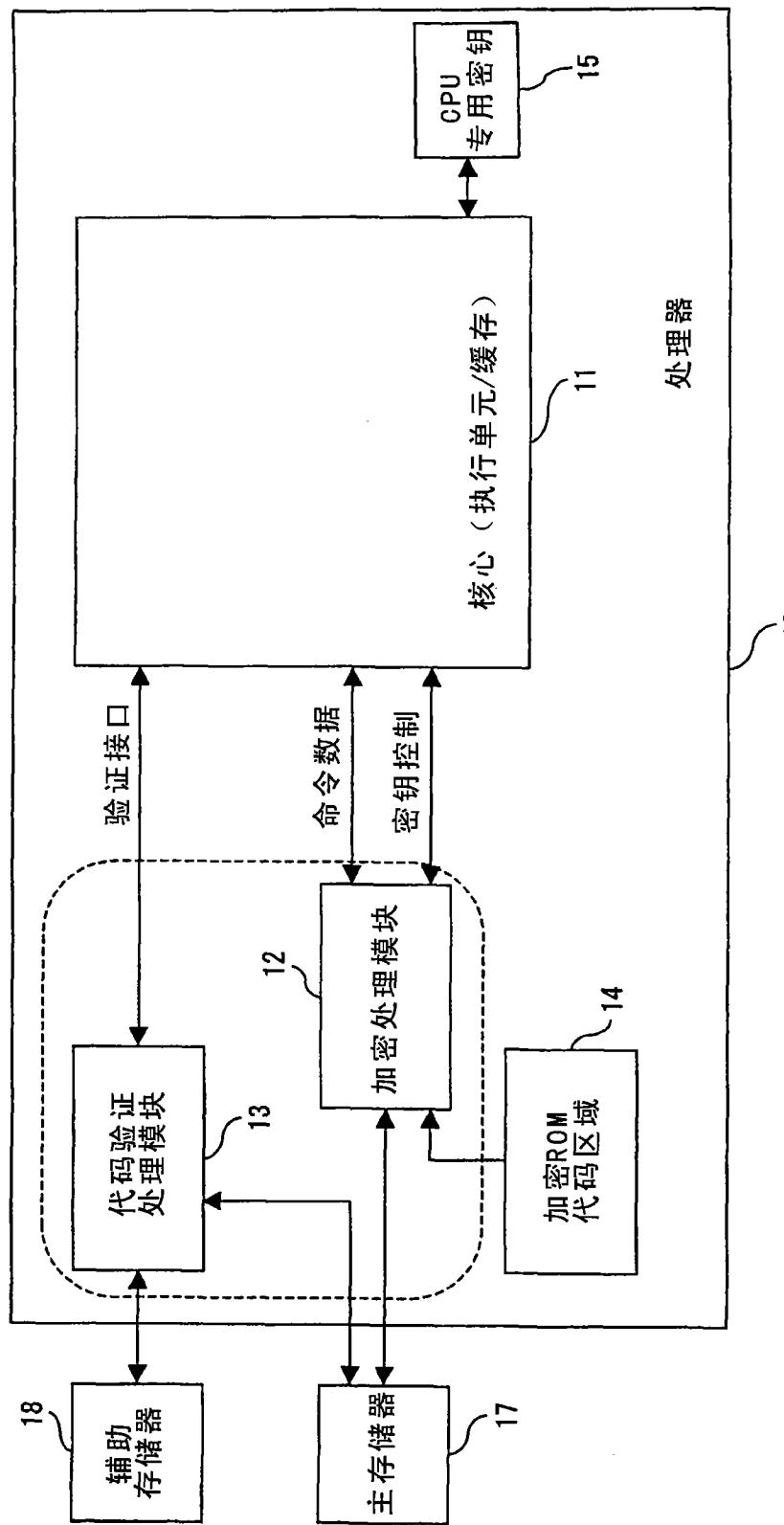
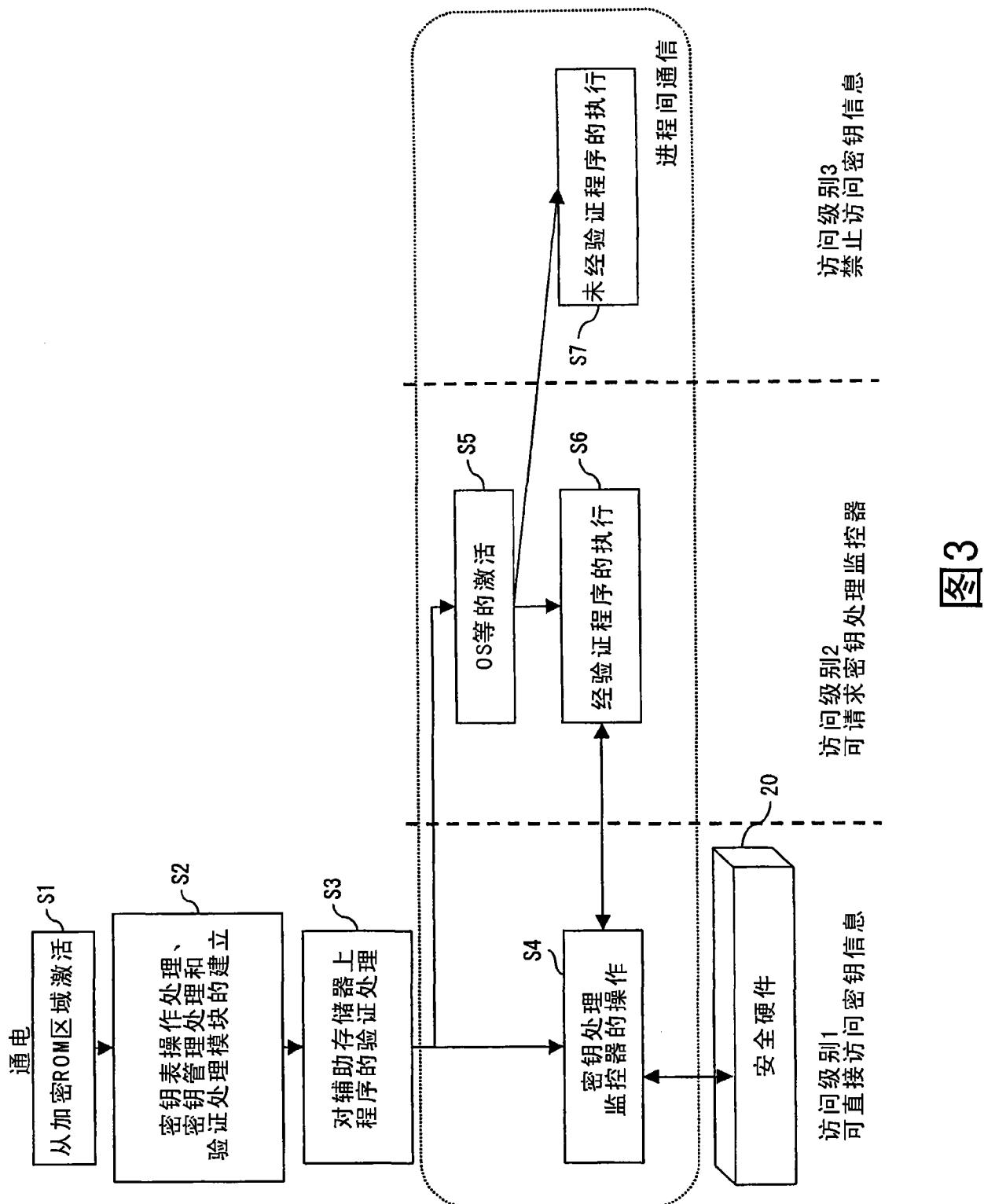


图2



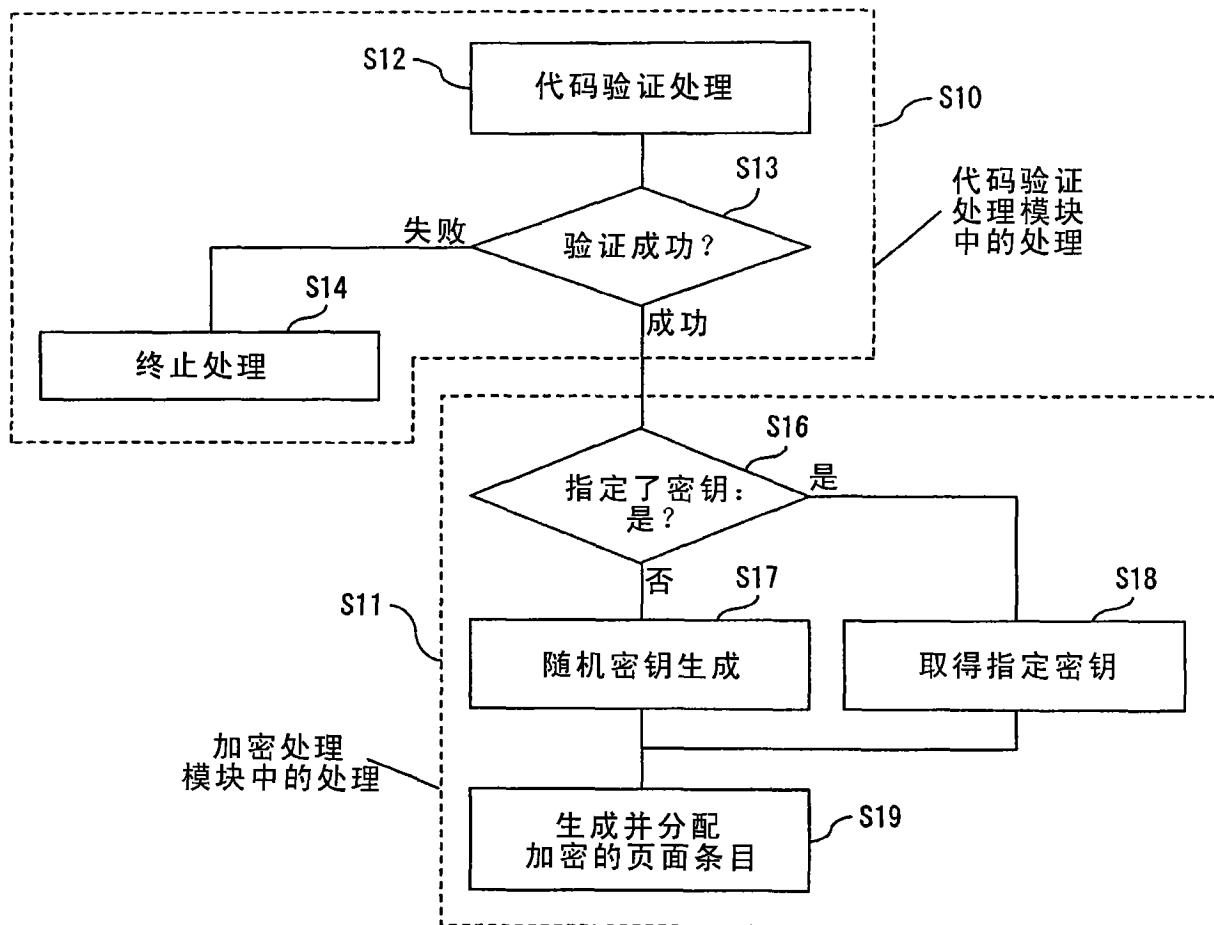


图 4

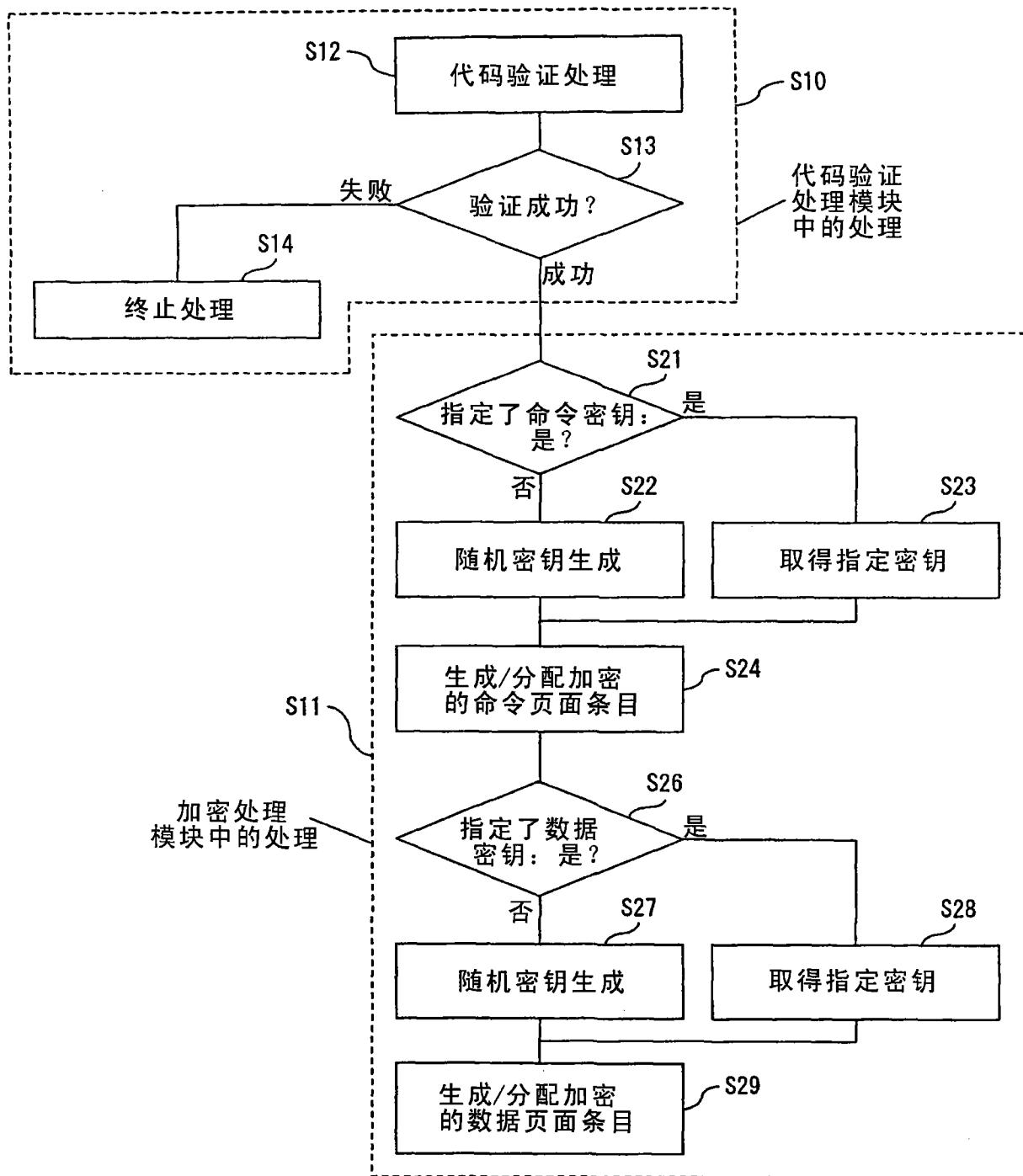


图 5

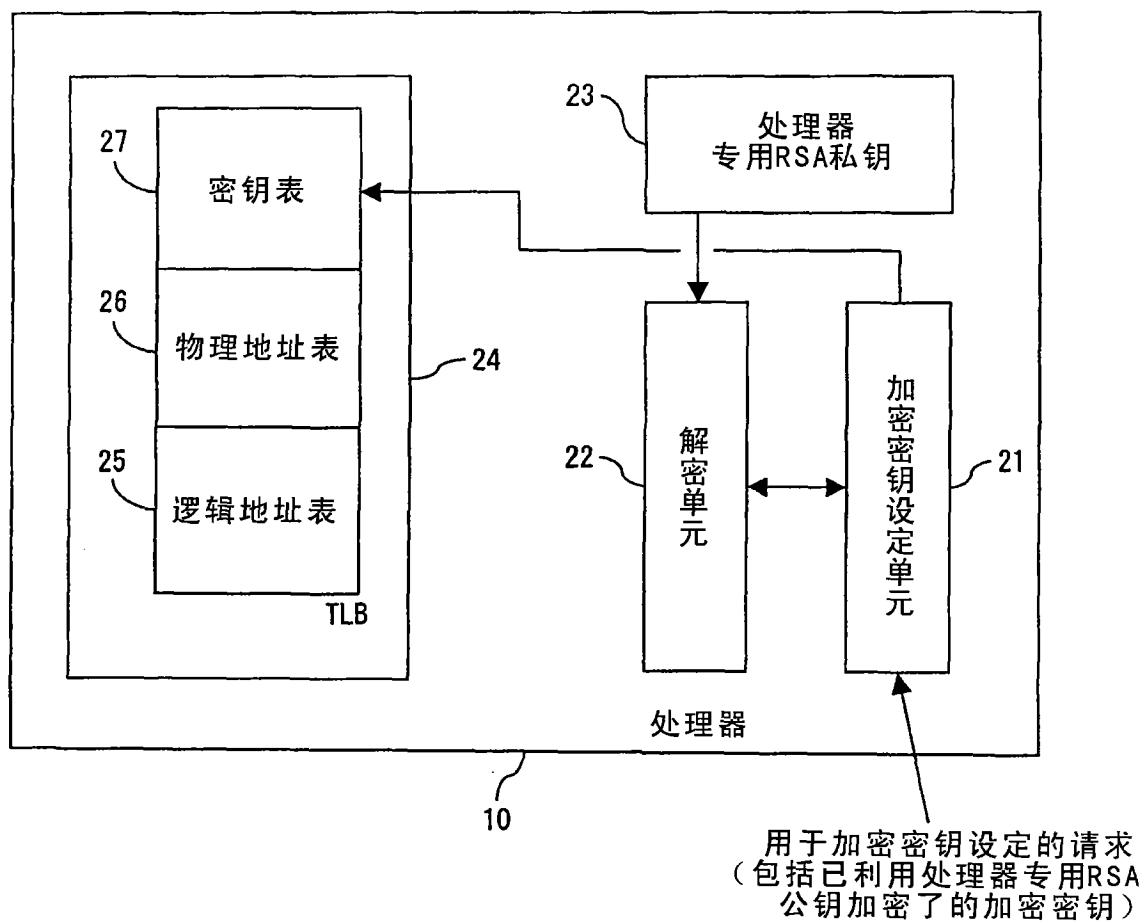


图 6

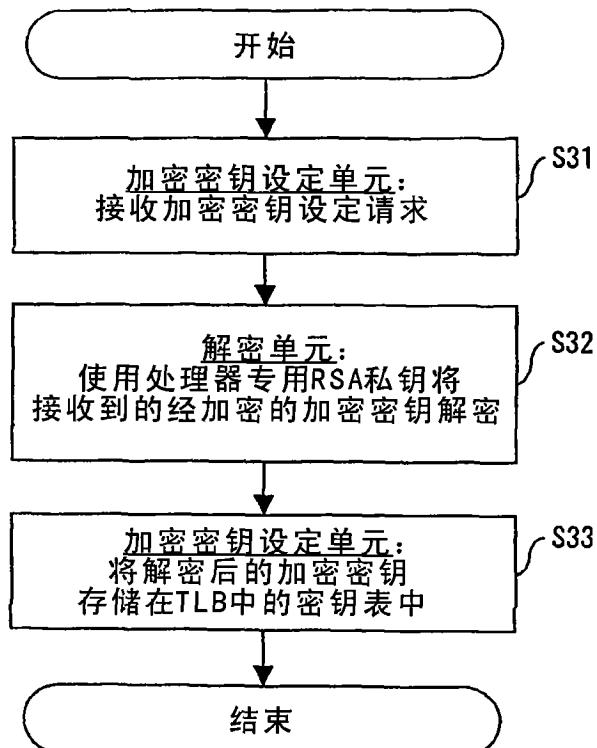


图 7

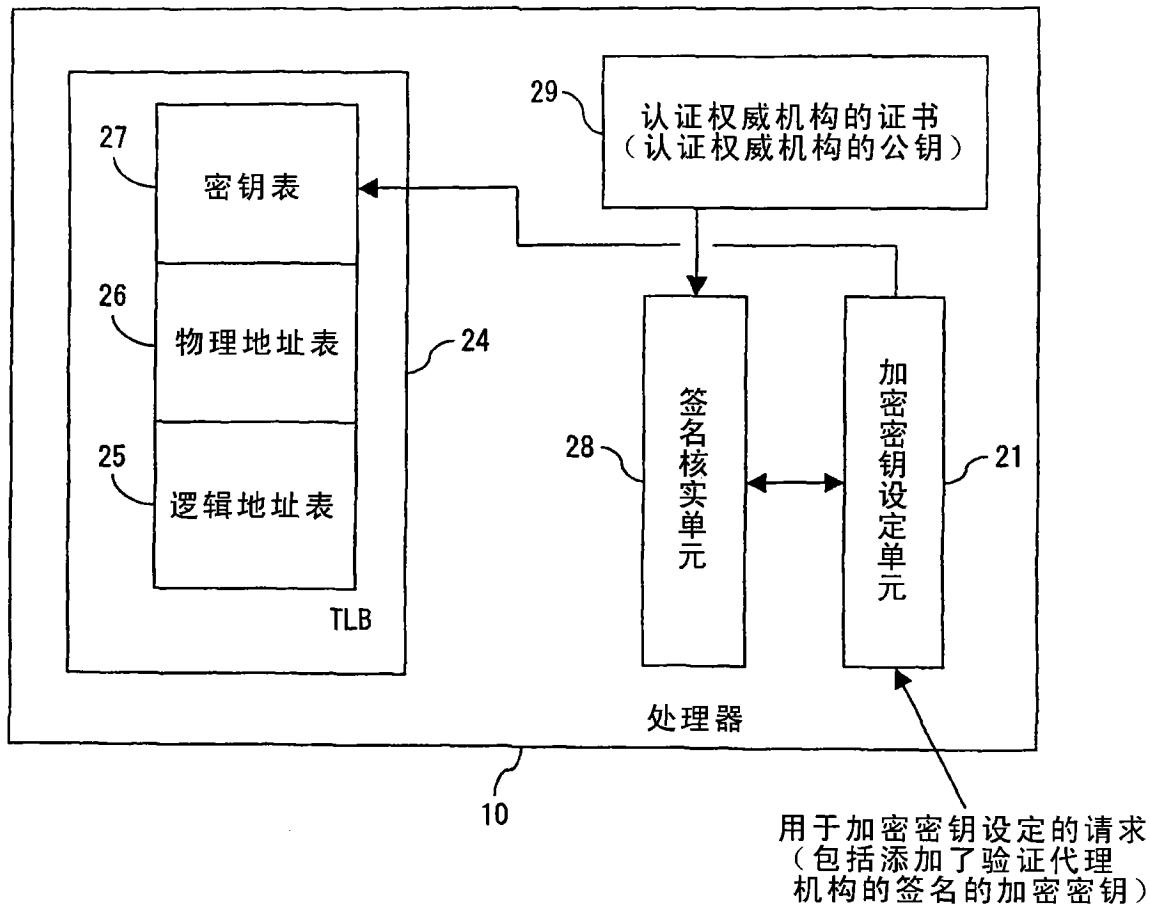


图 8

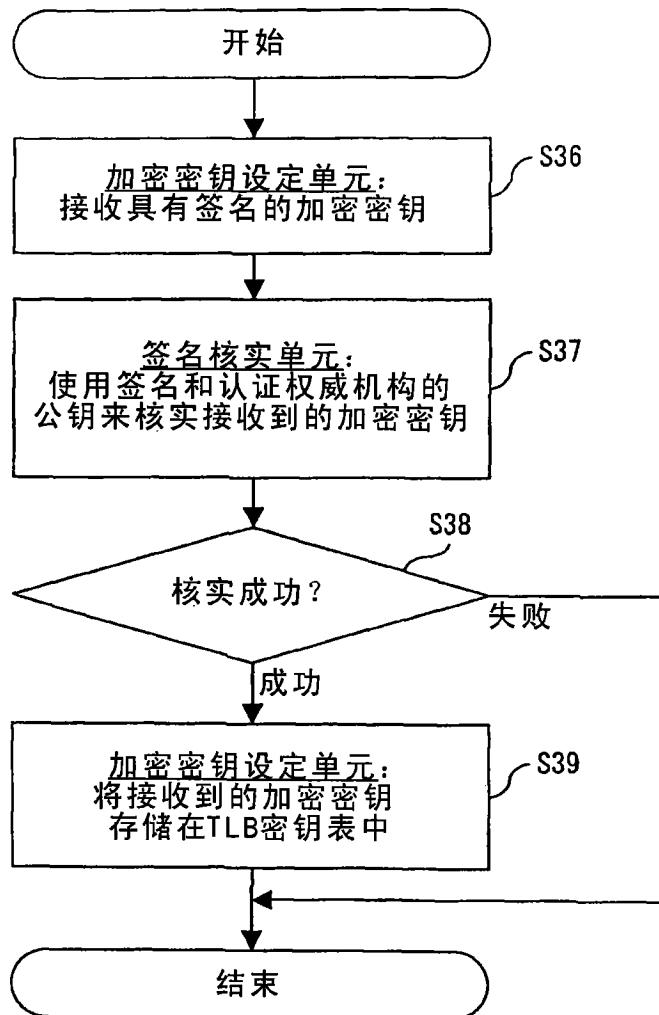


图 9

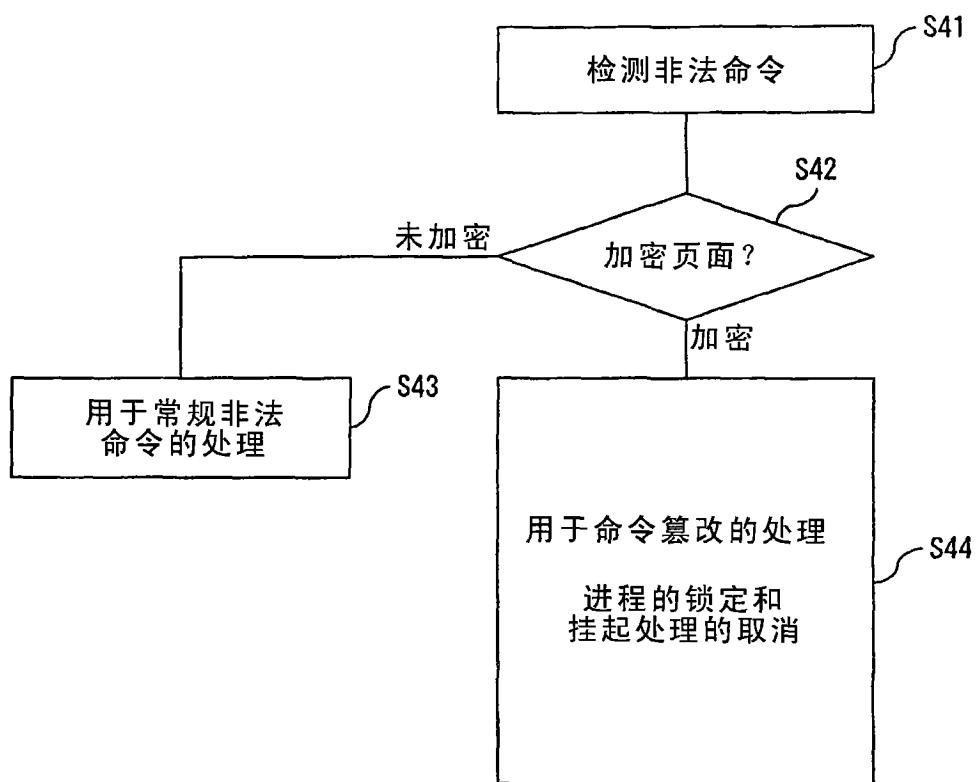


图 10

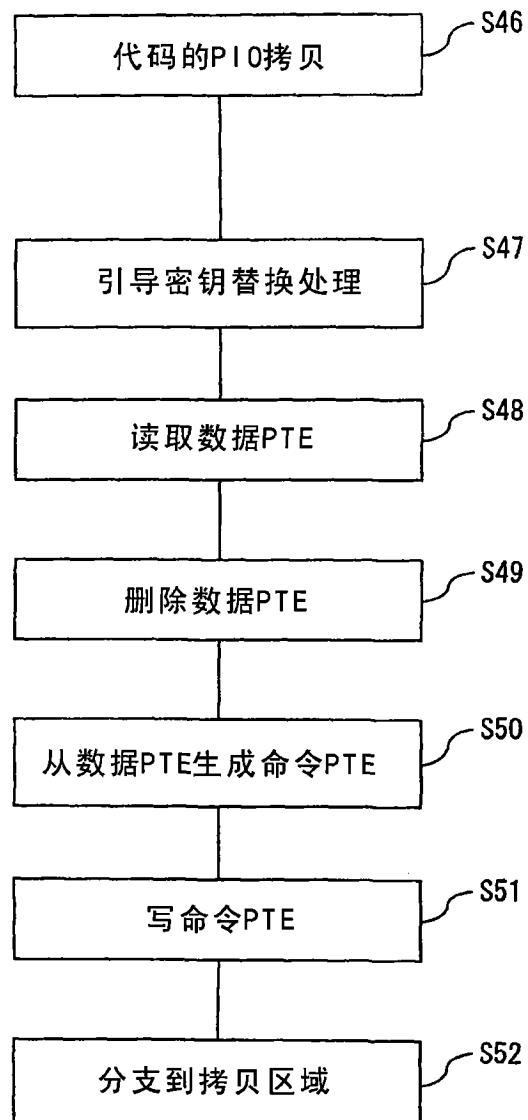


图 11

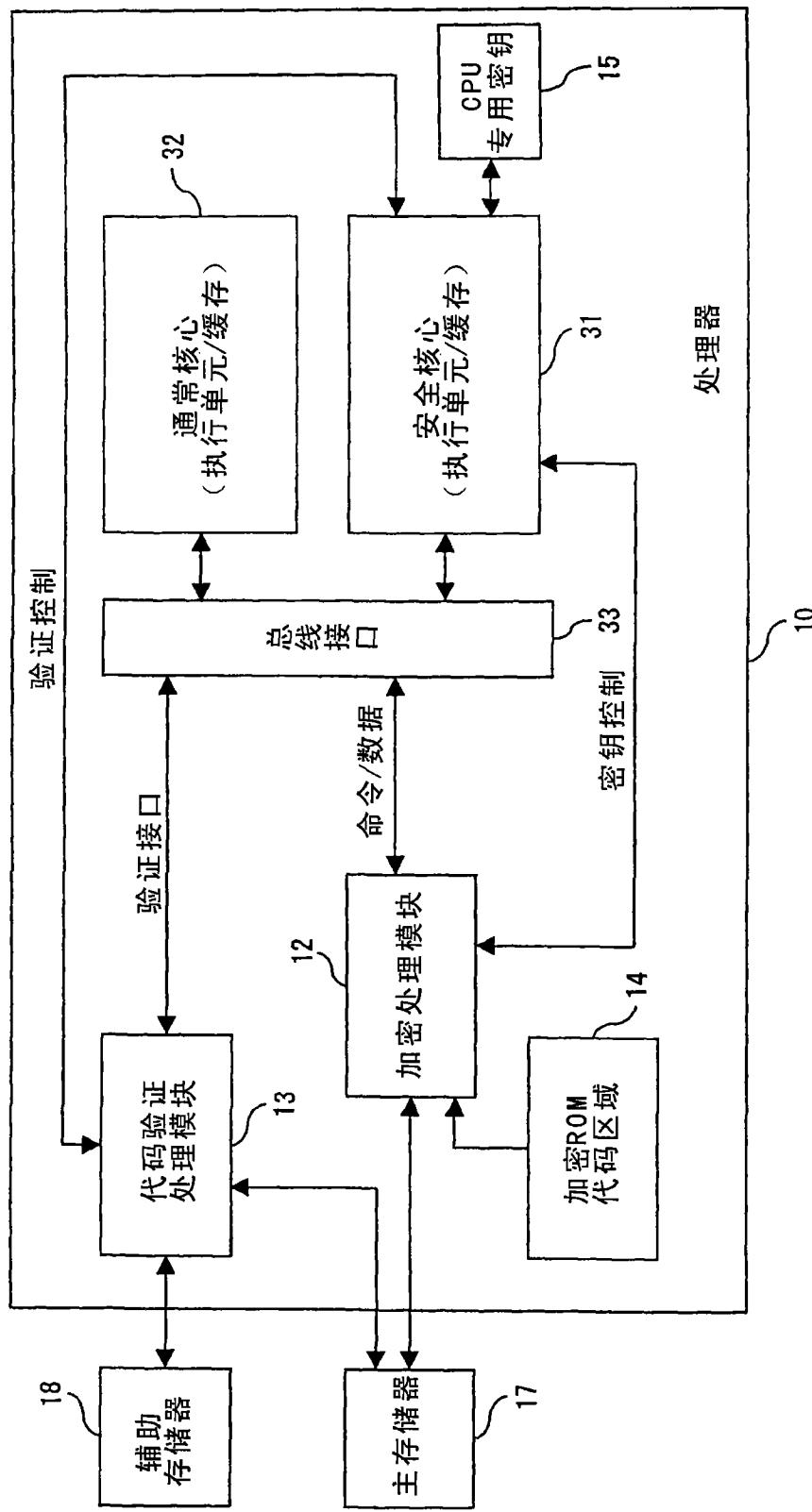


图 12

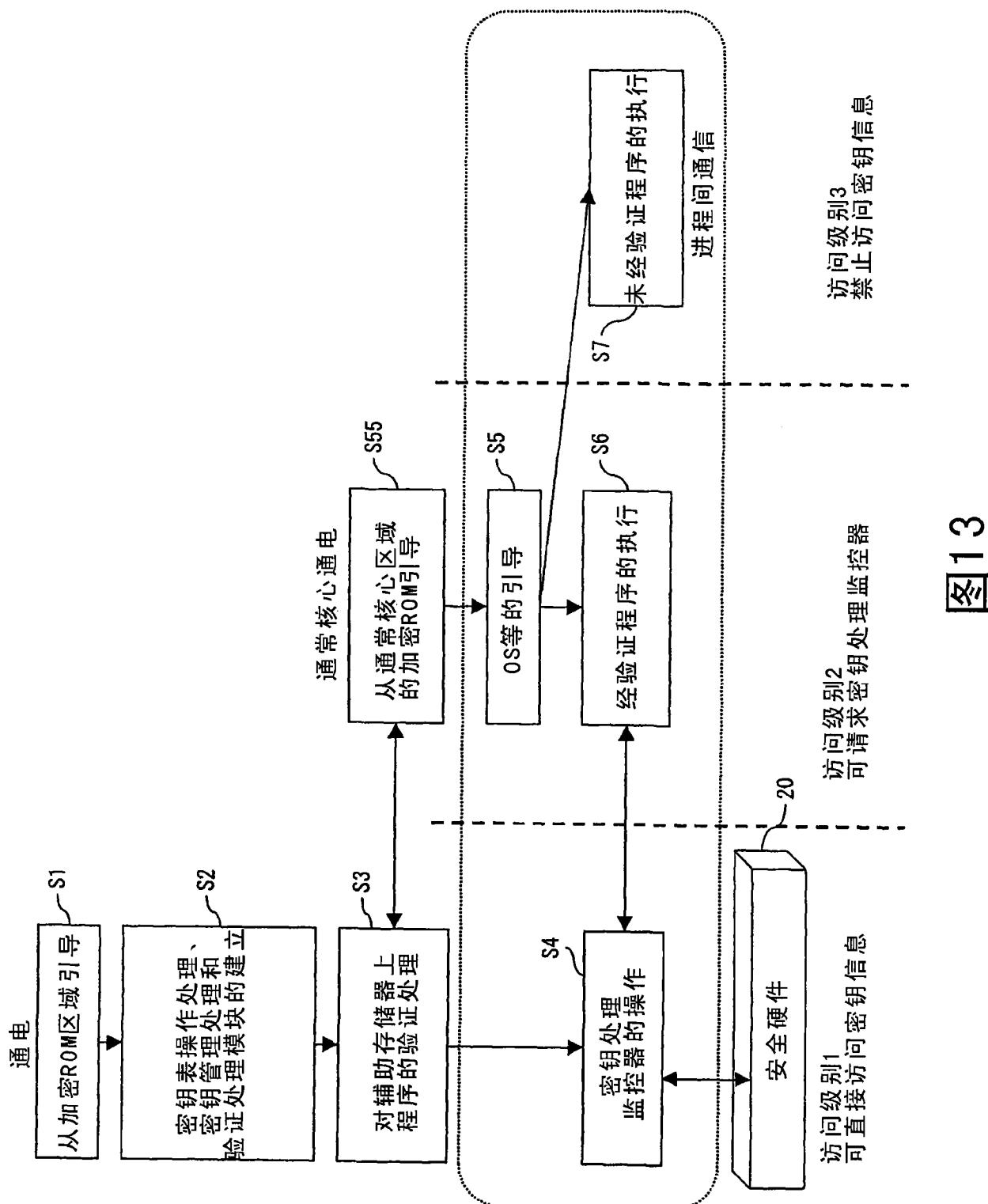


图13

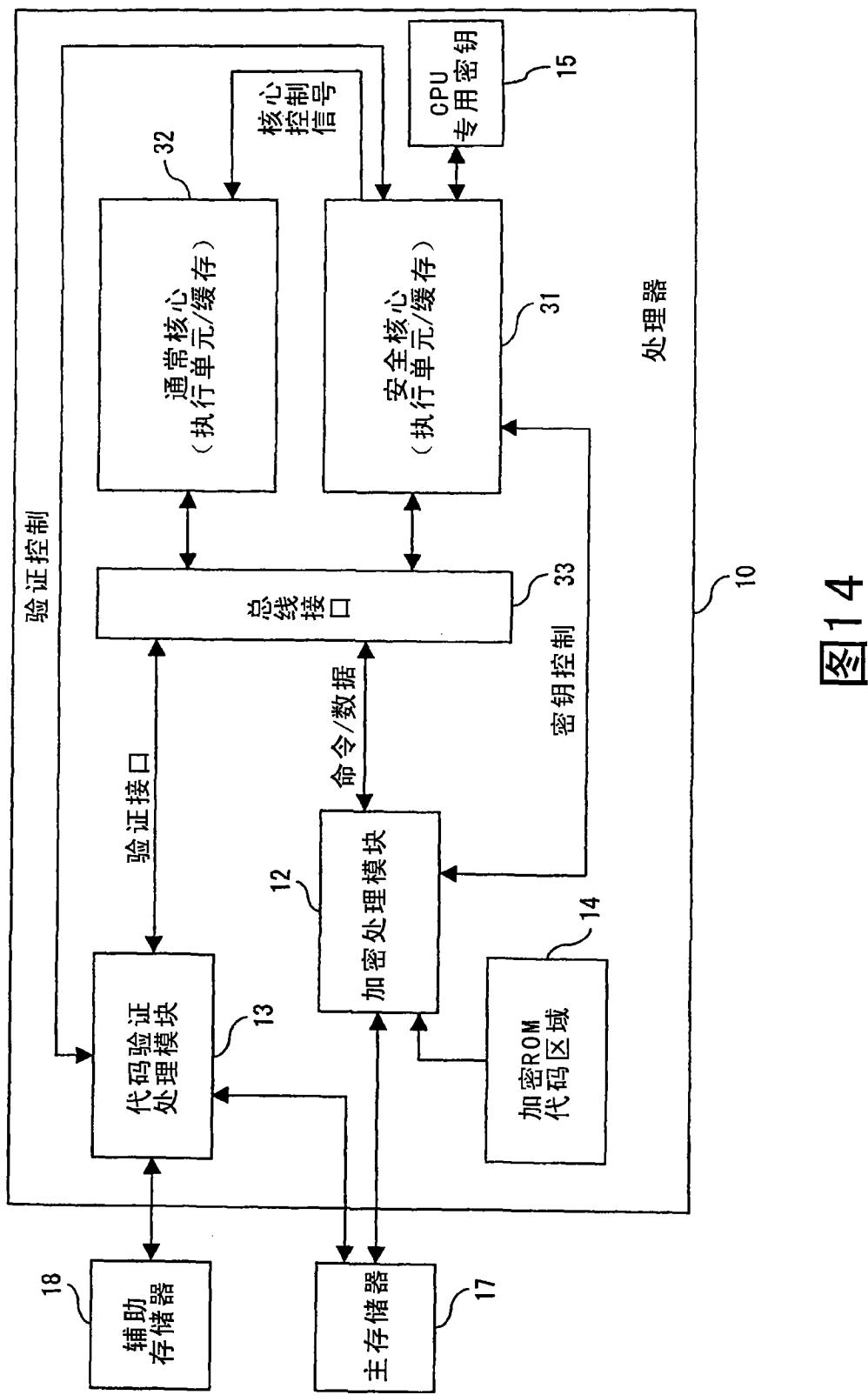
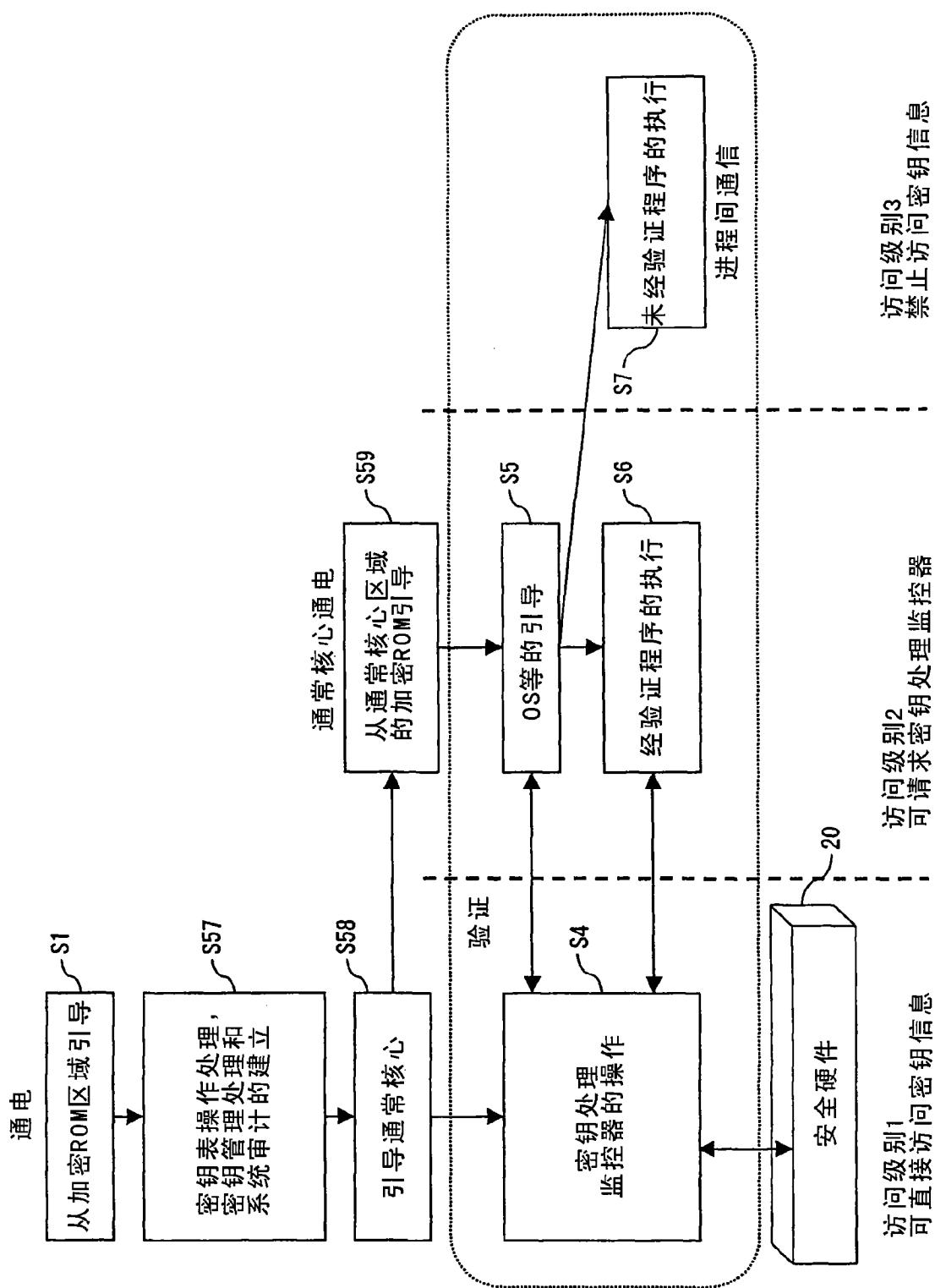


图 14



15

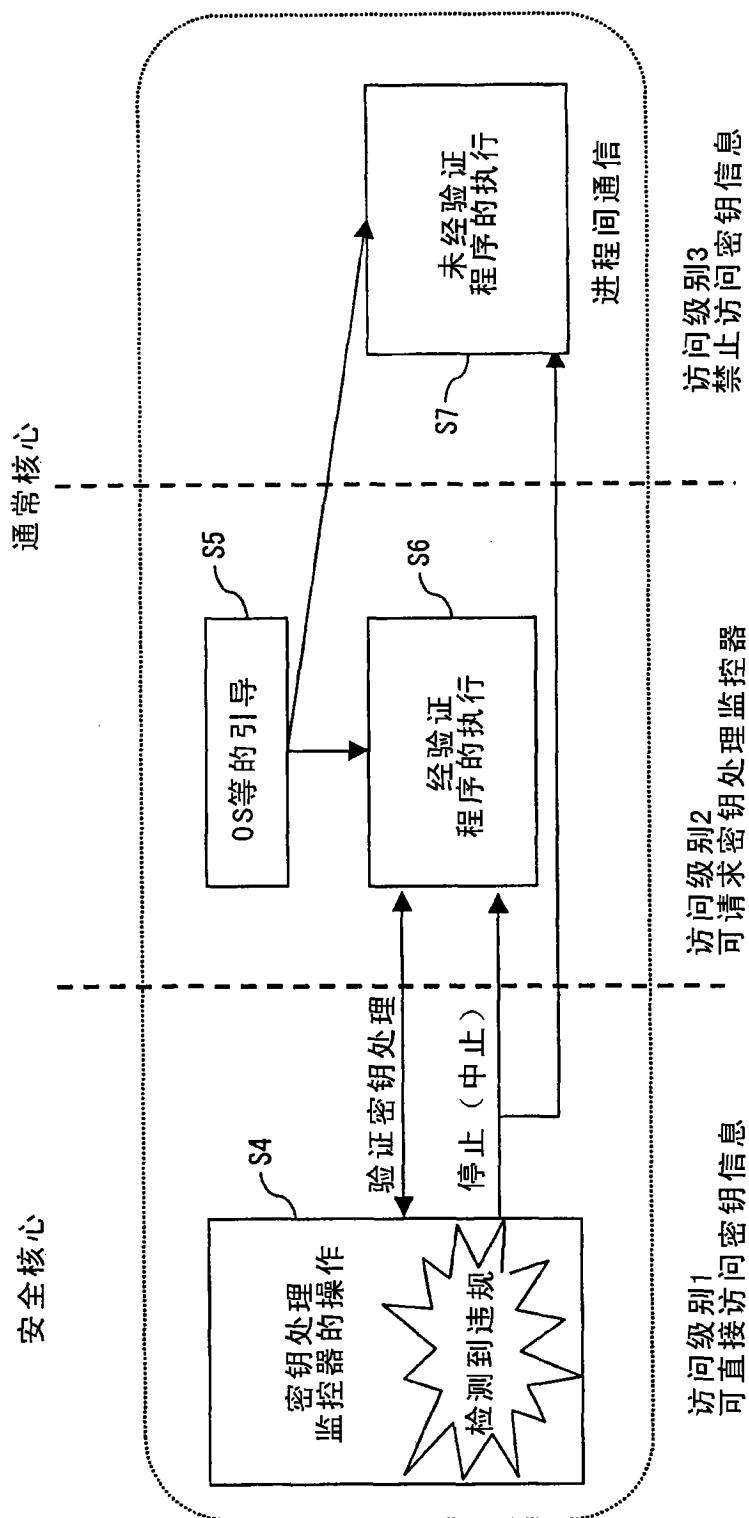


图 16

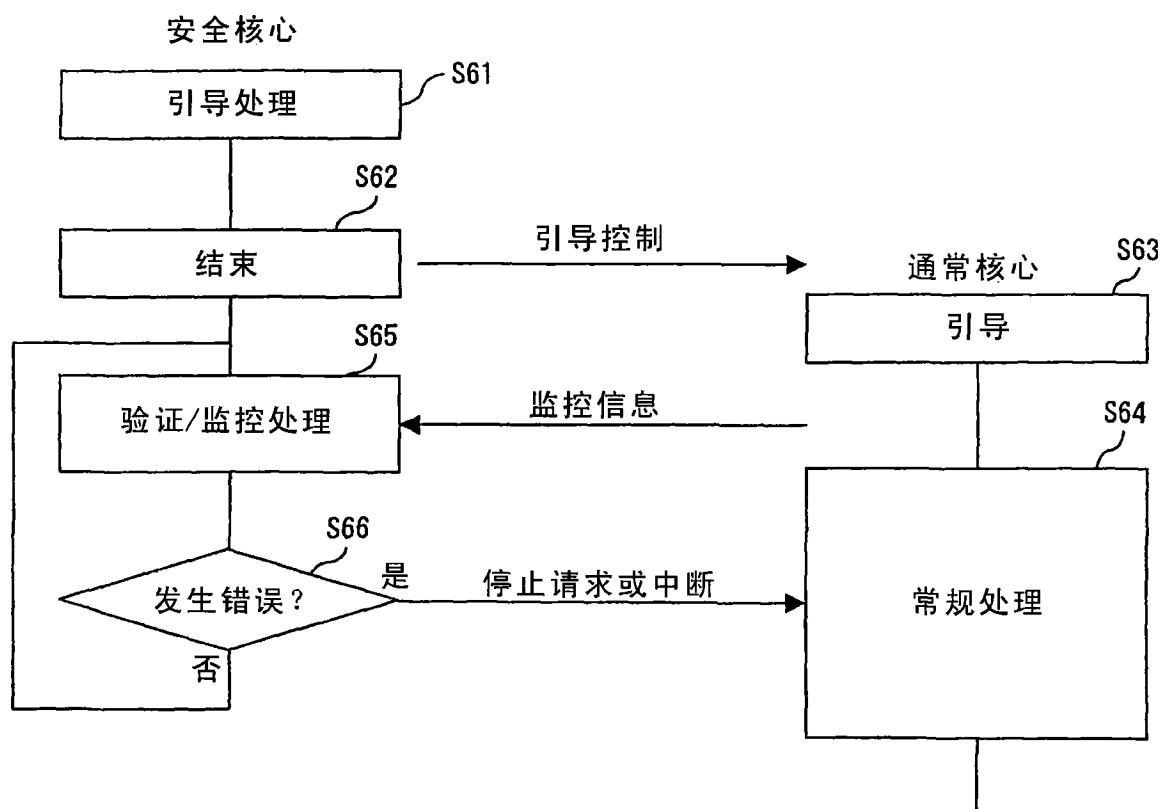


图 17

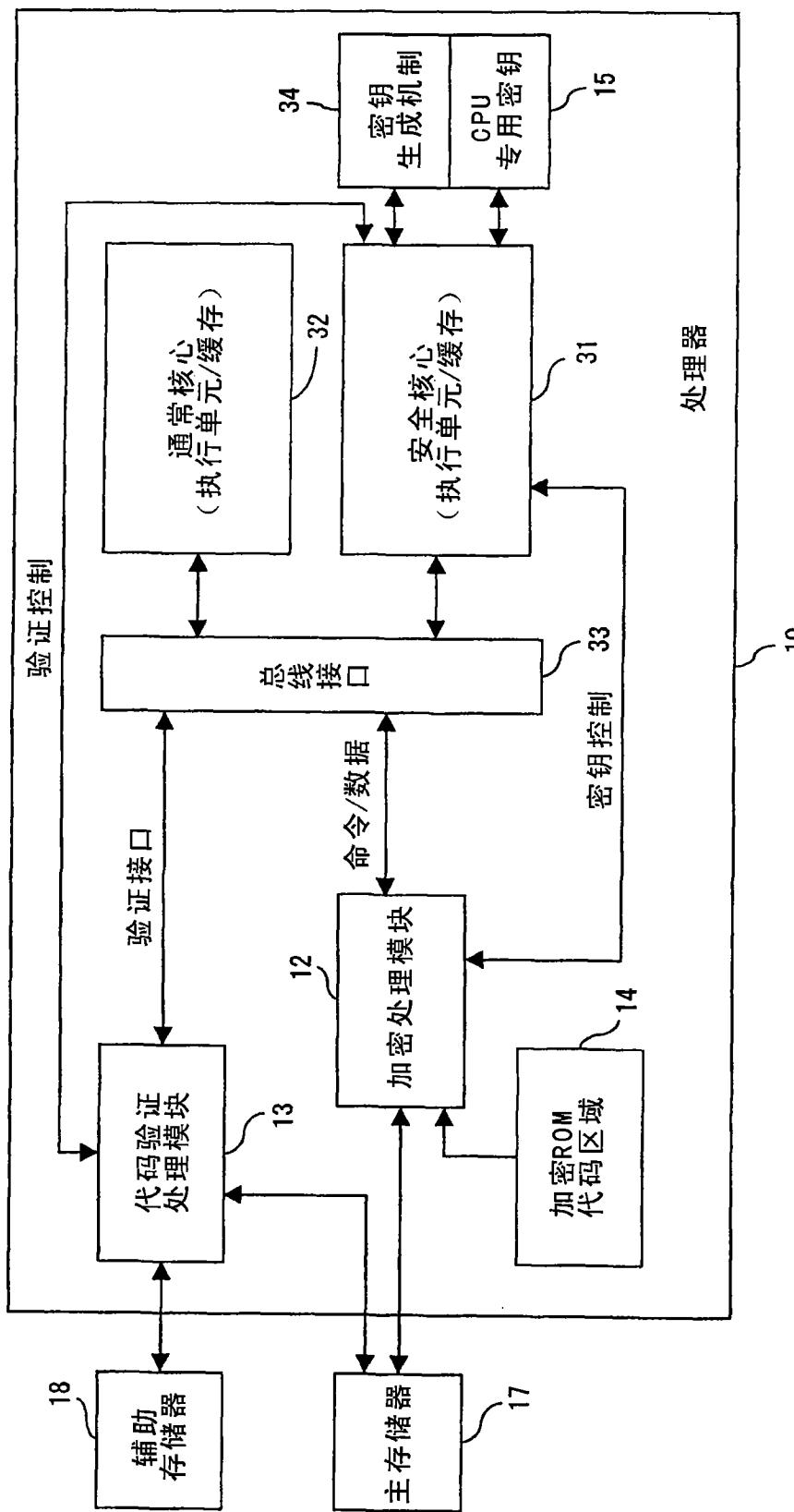


图 18

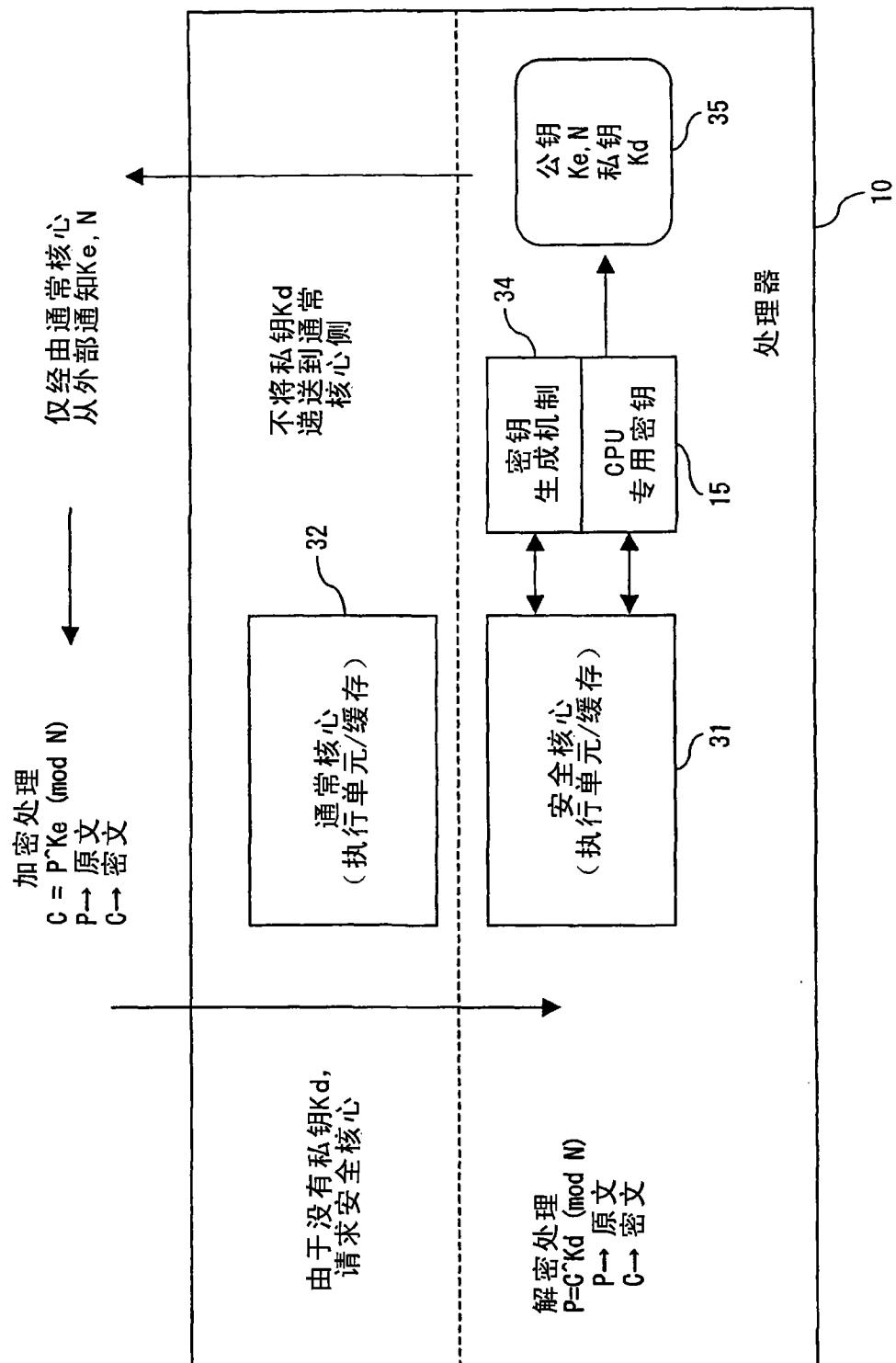


图19

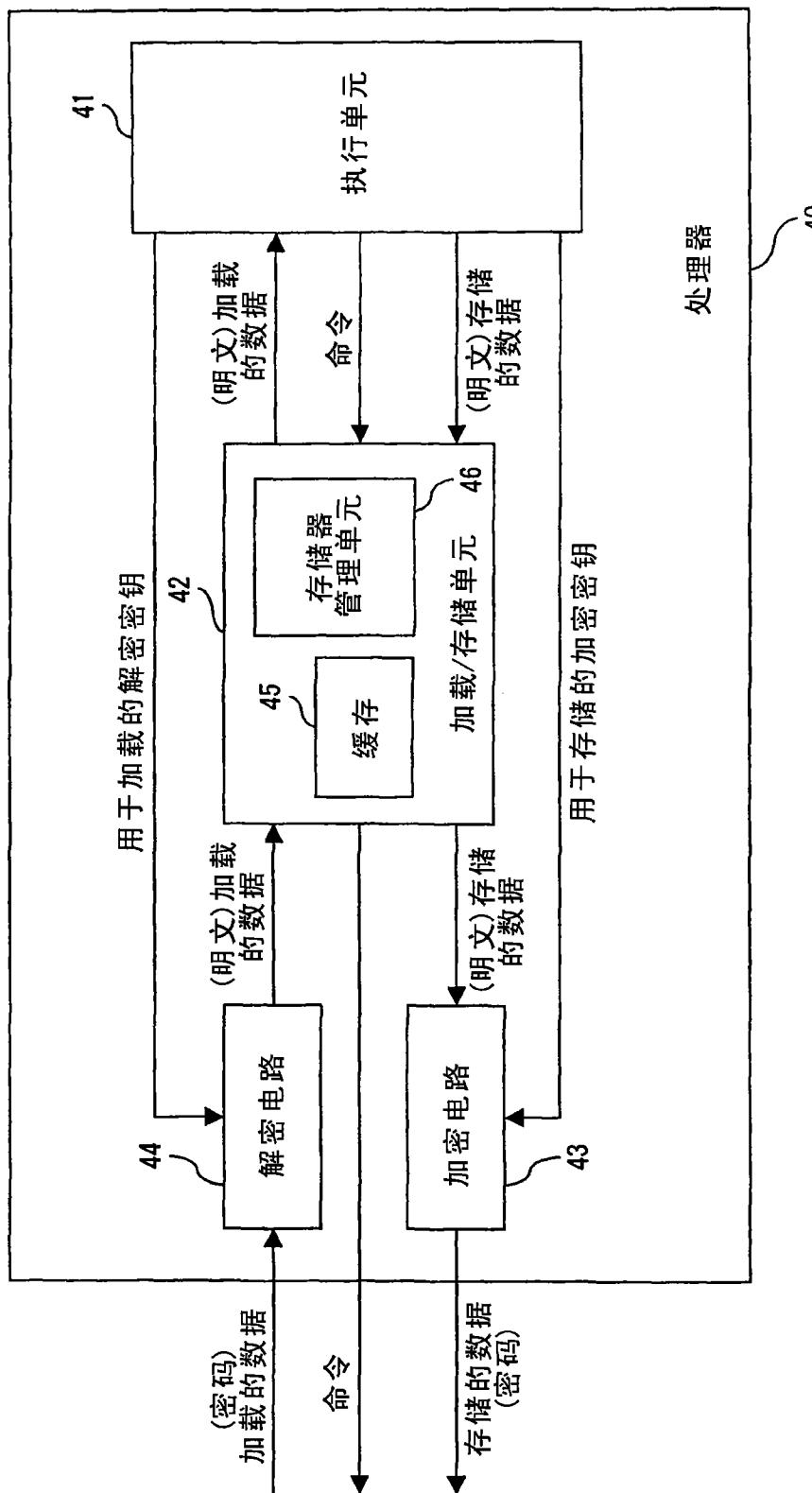
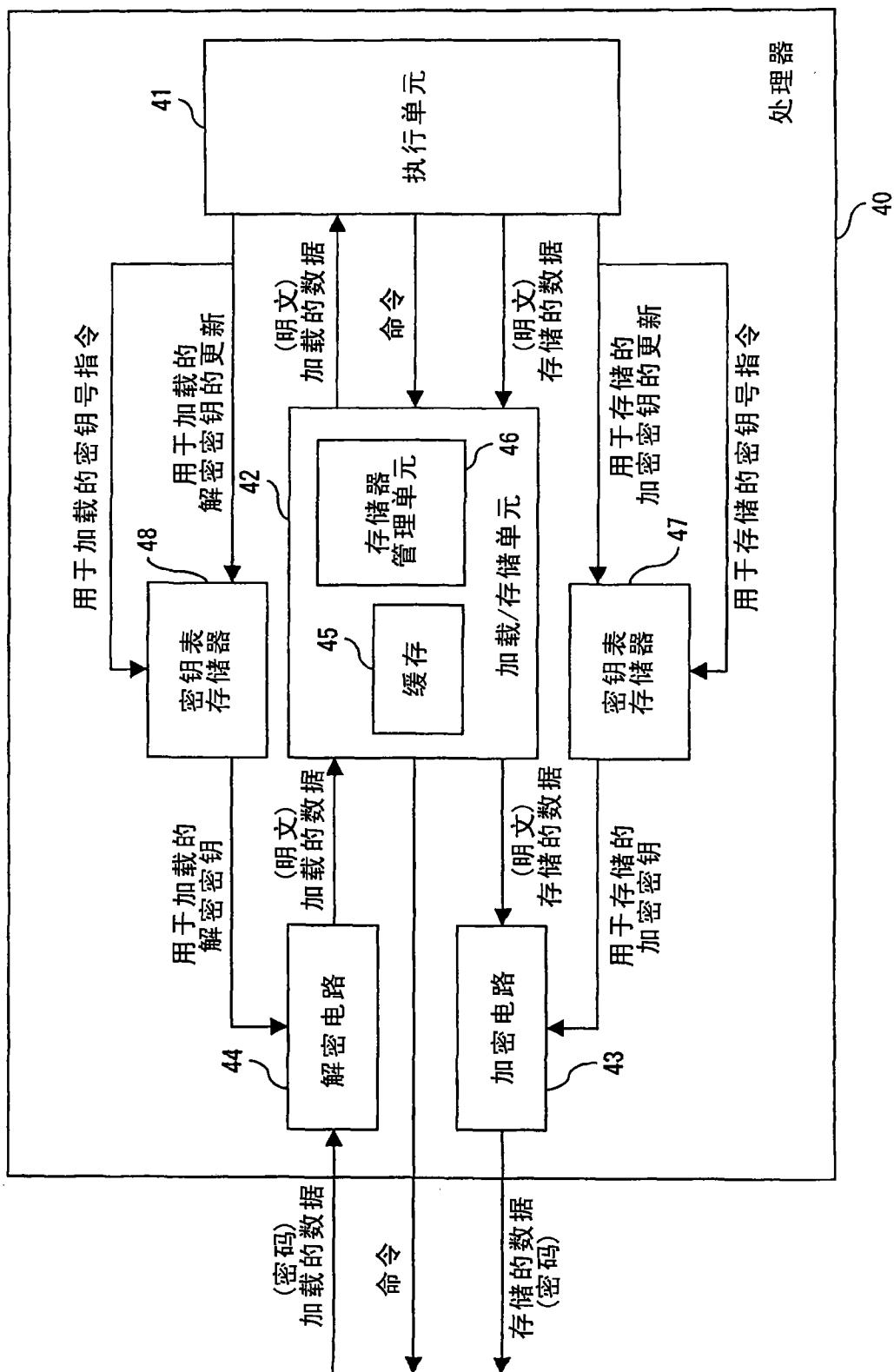


图 20



21

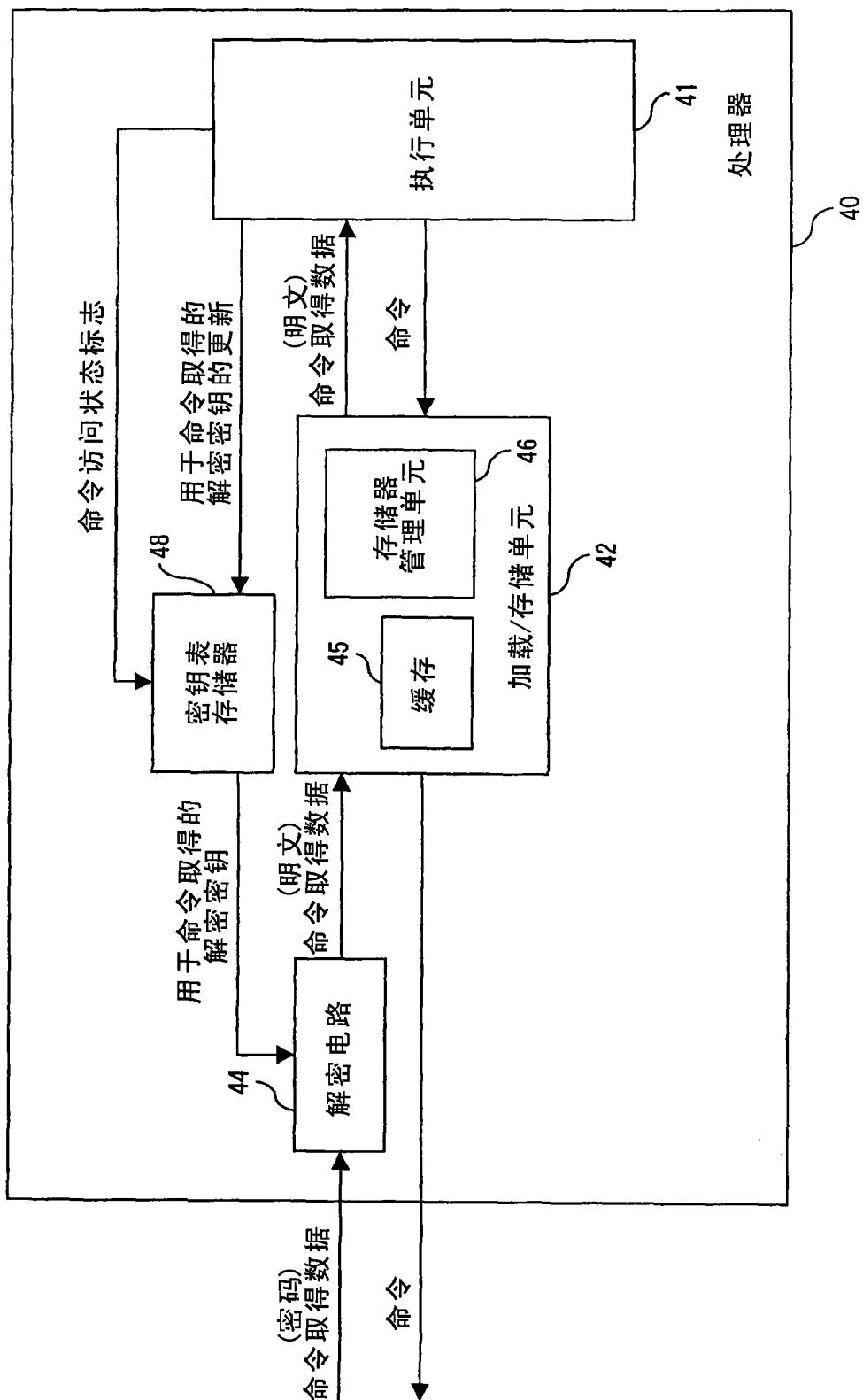


图22

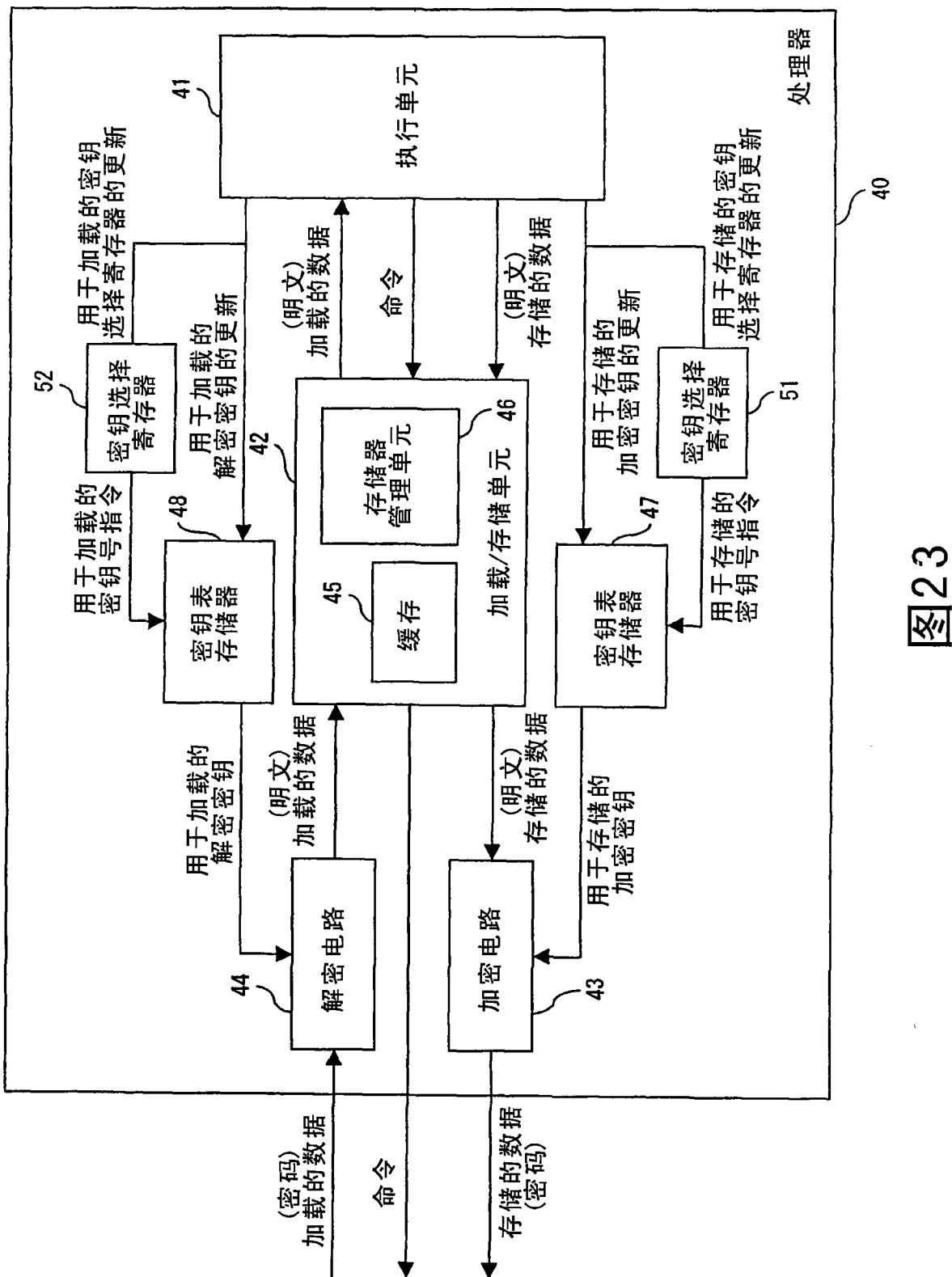


图 23

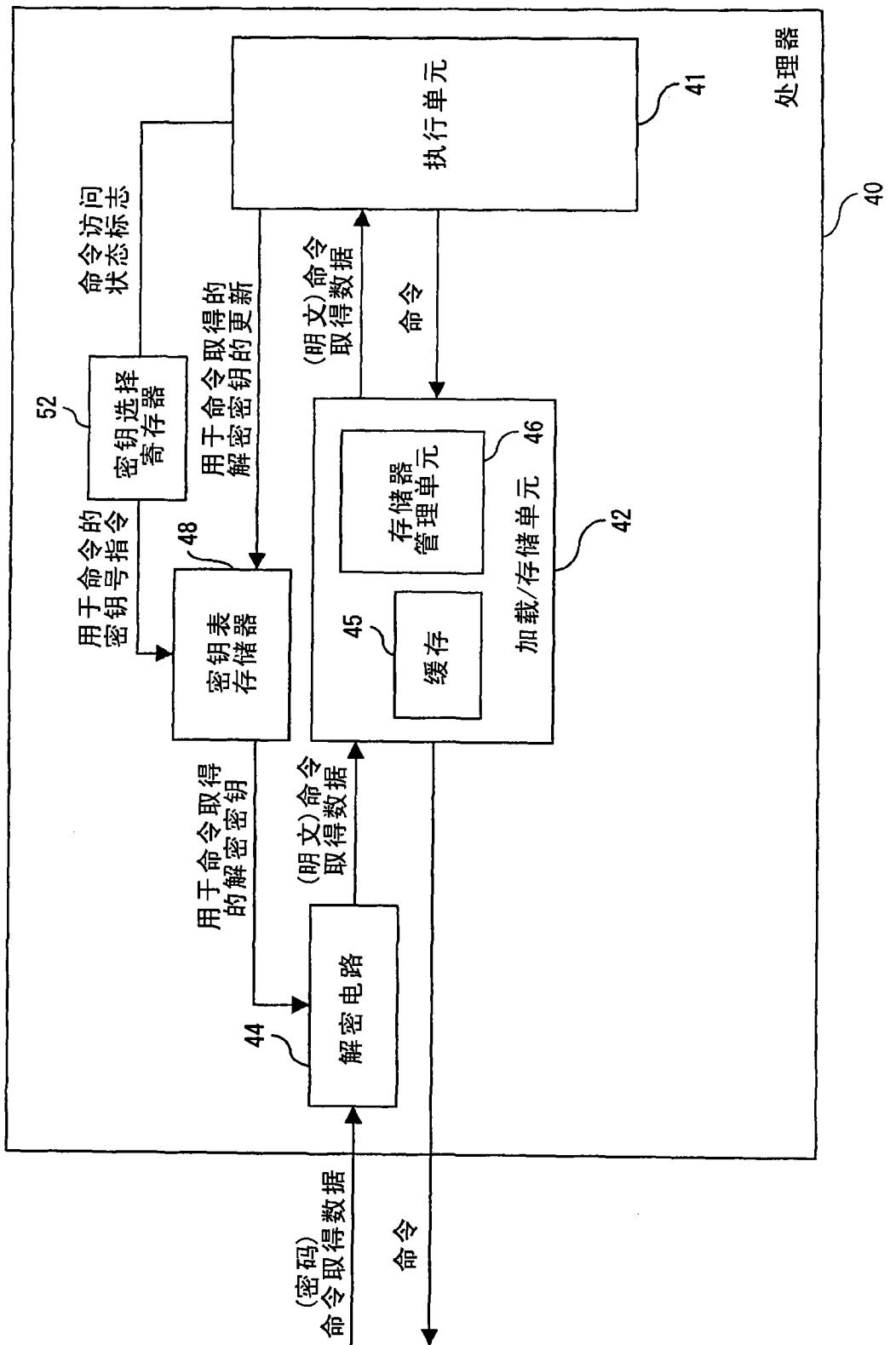


图24

图23和图24中的配置

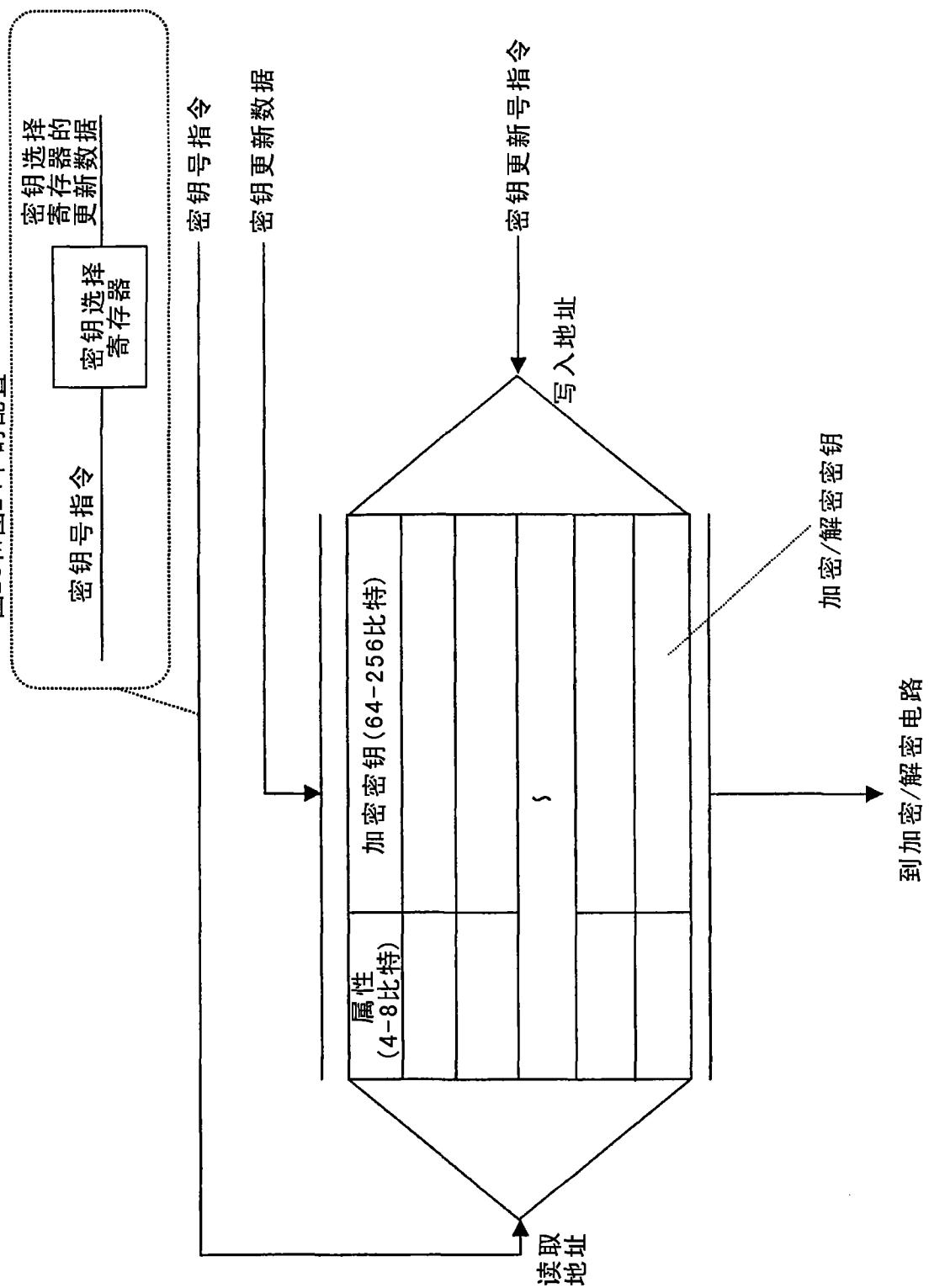
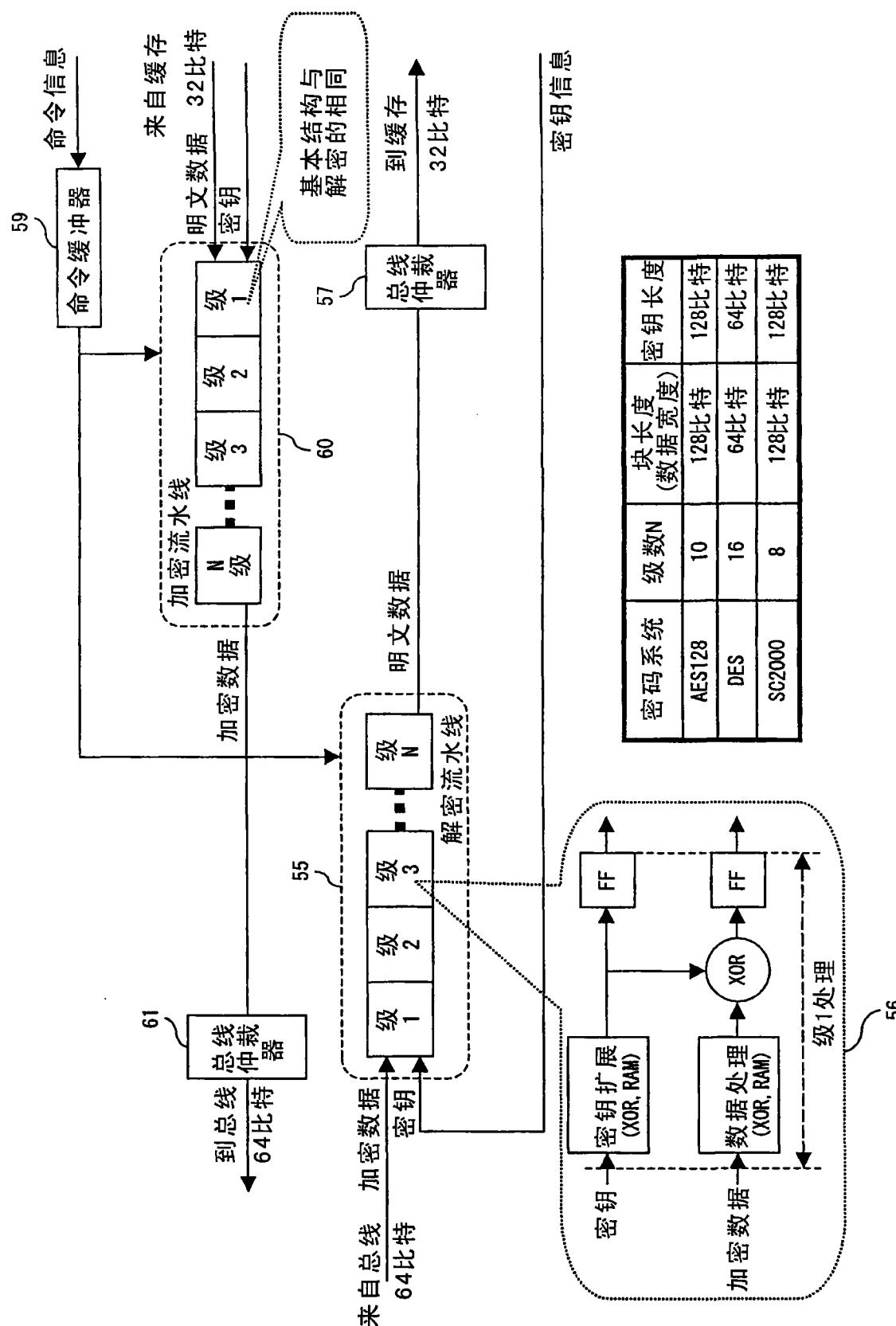
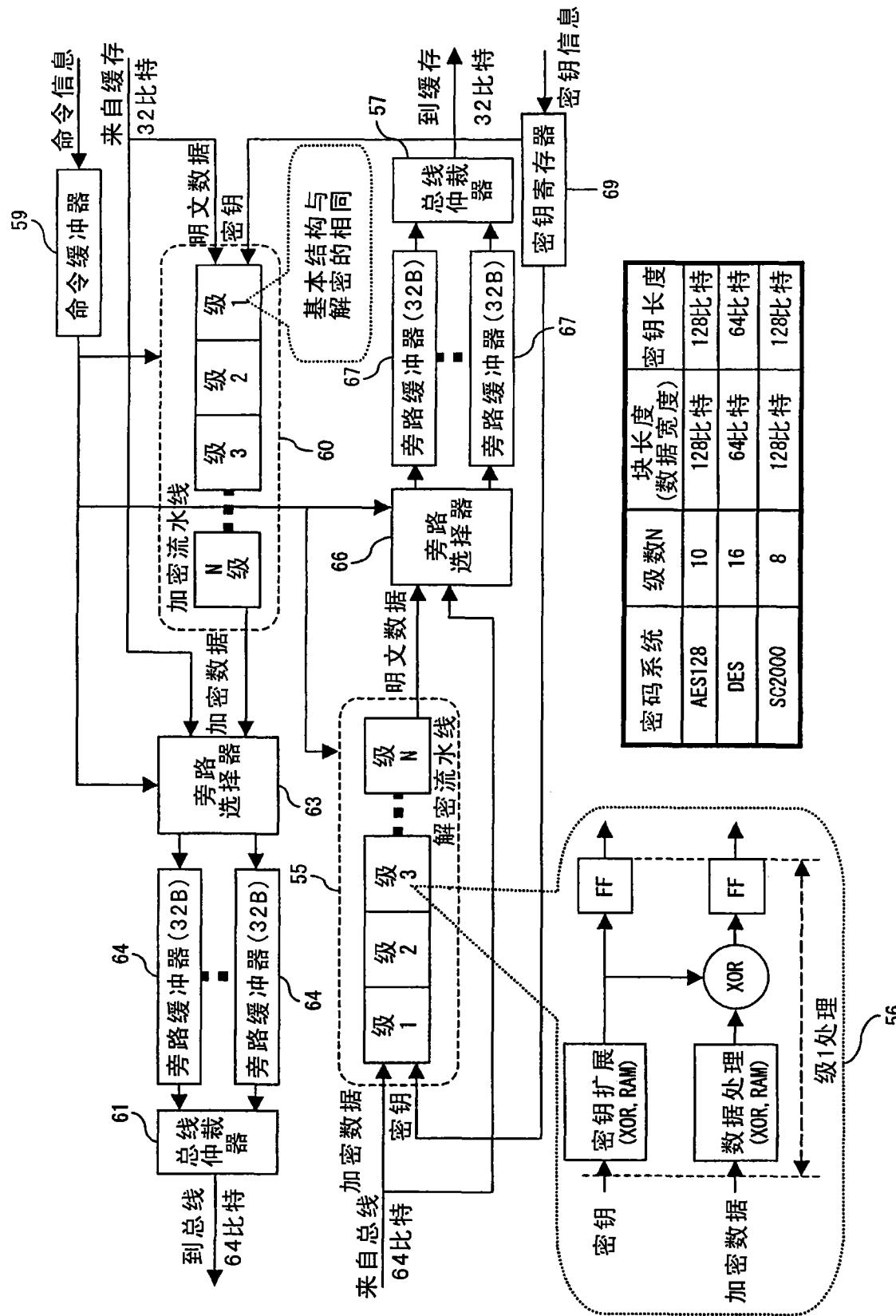


图25





27

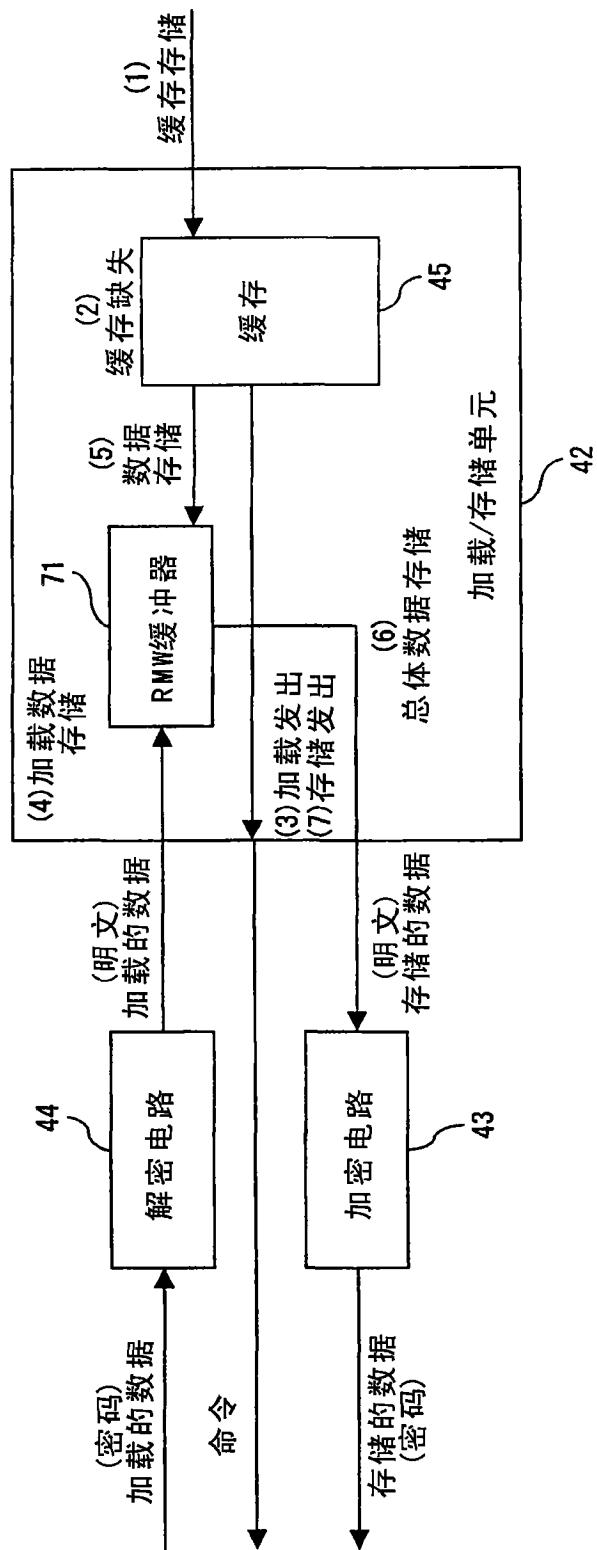


图 28

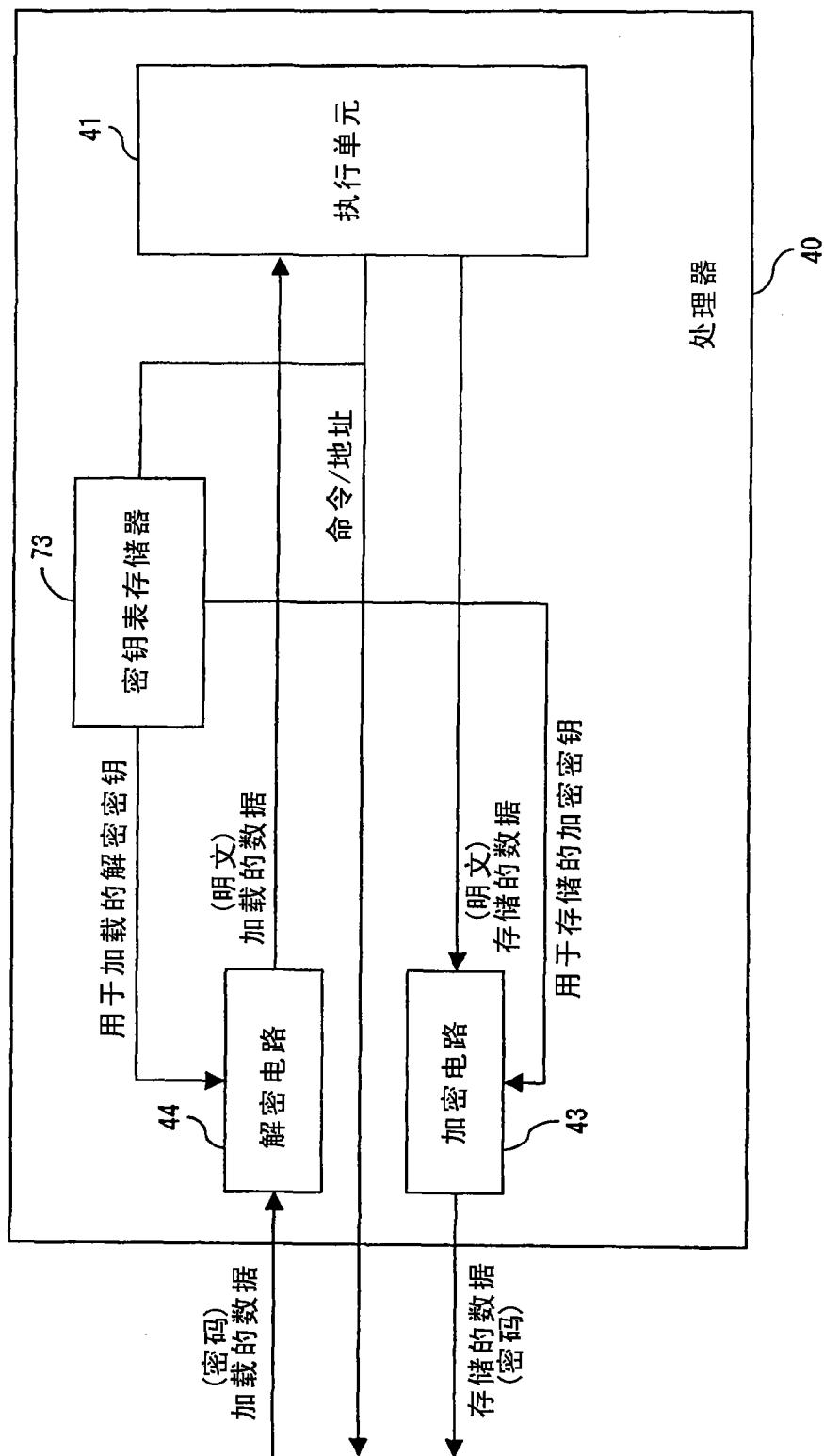


图29

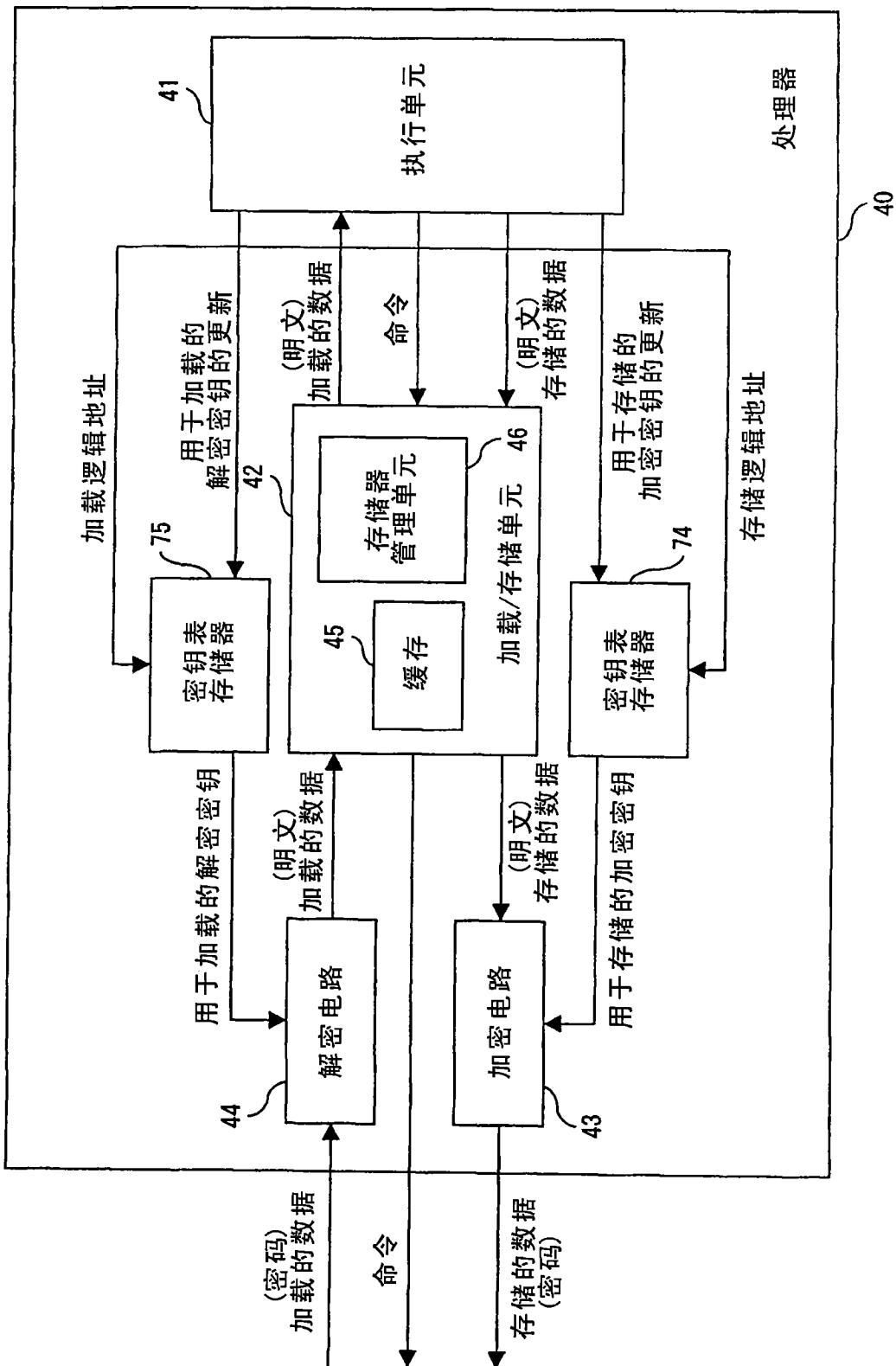


图 30

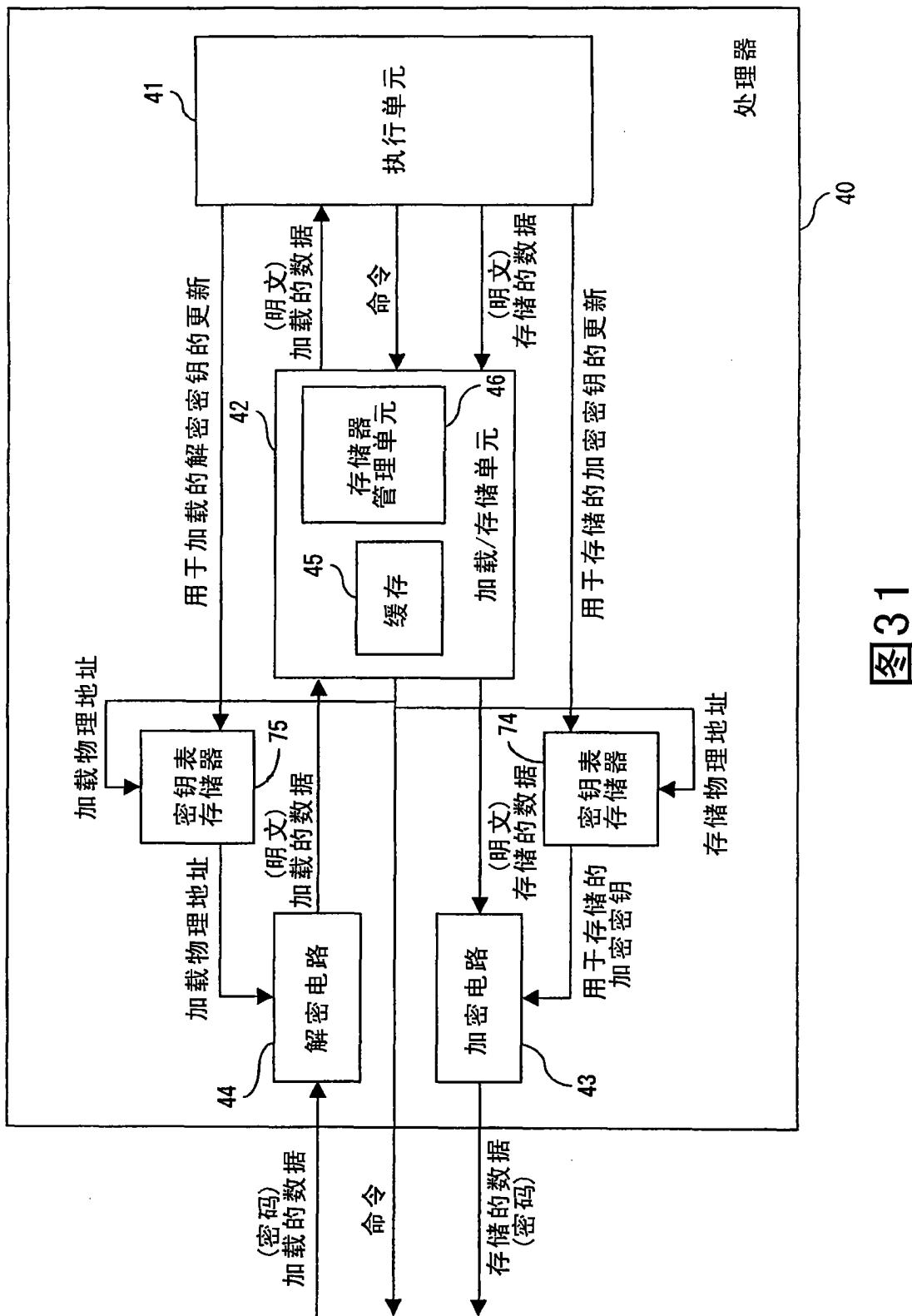


图31

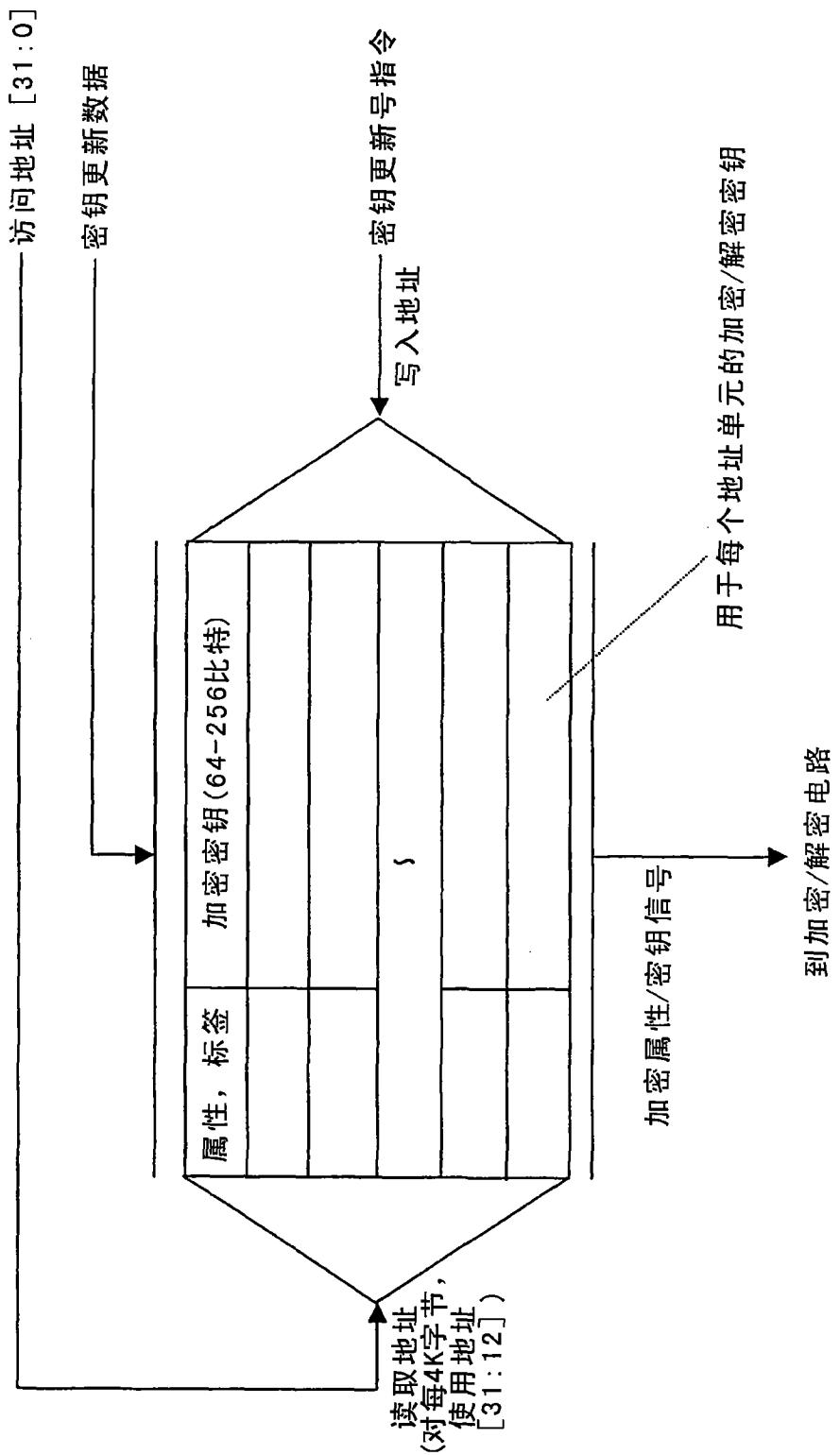


图32

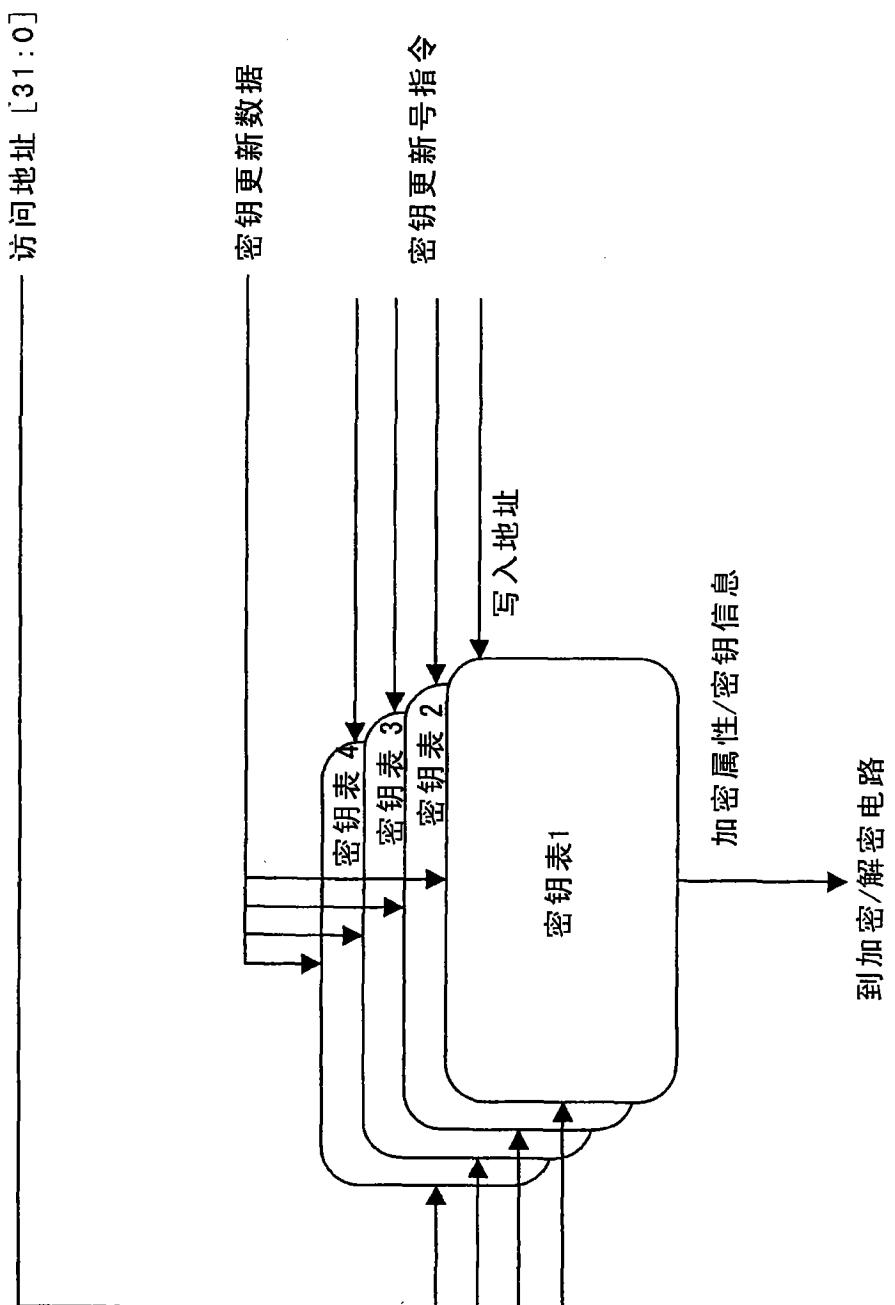


图33

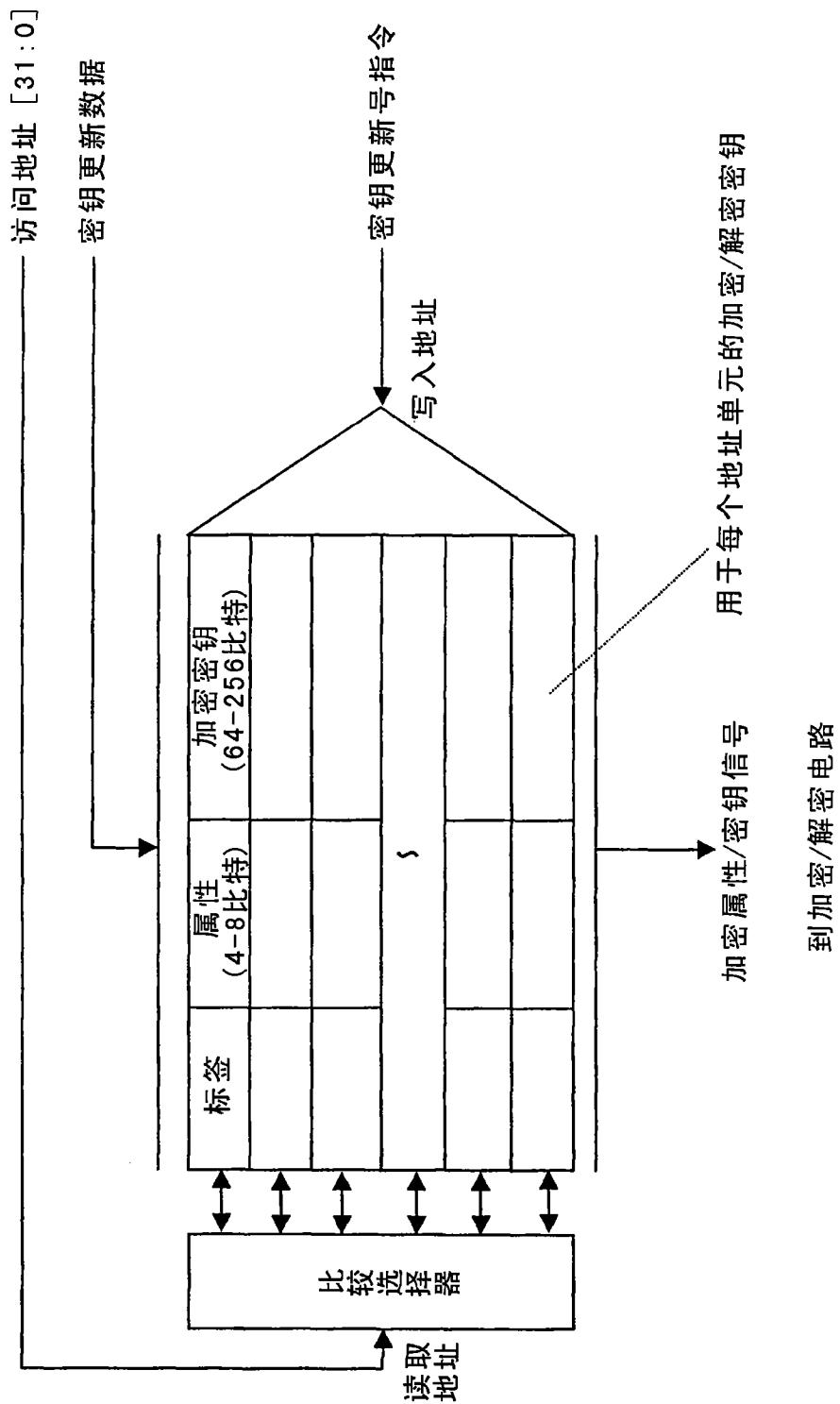


图 34

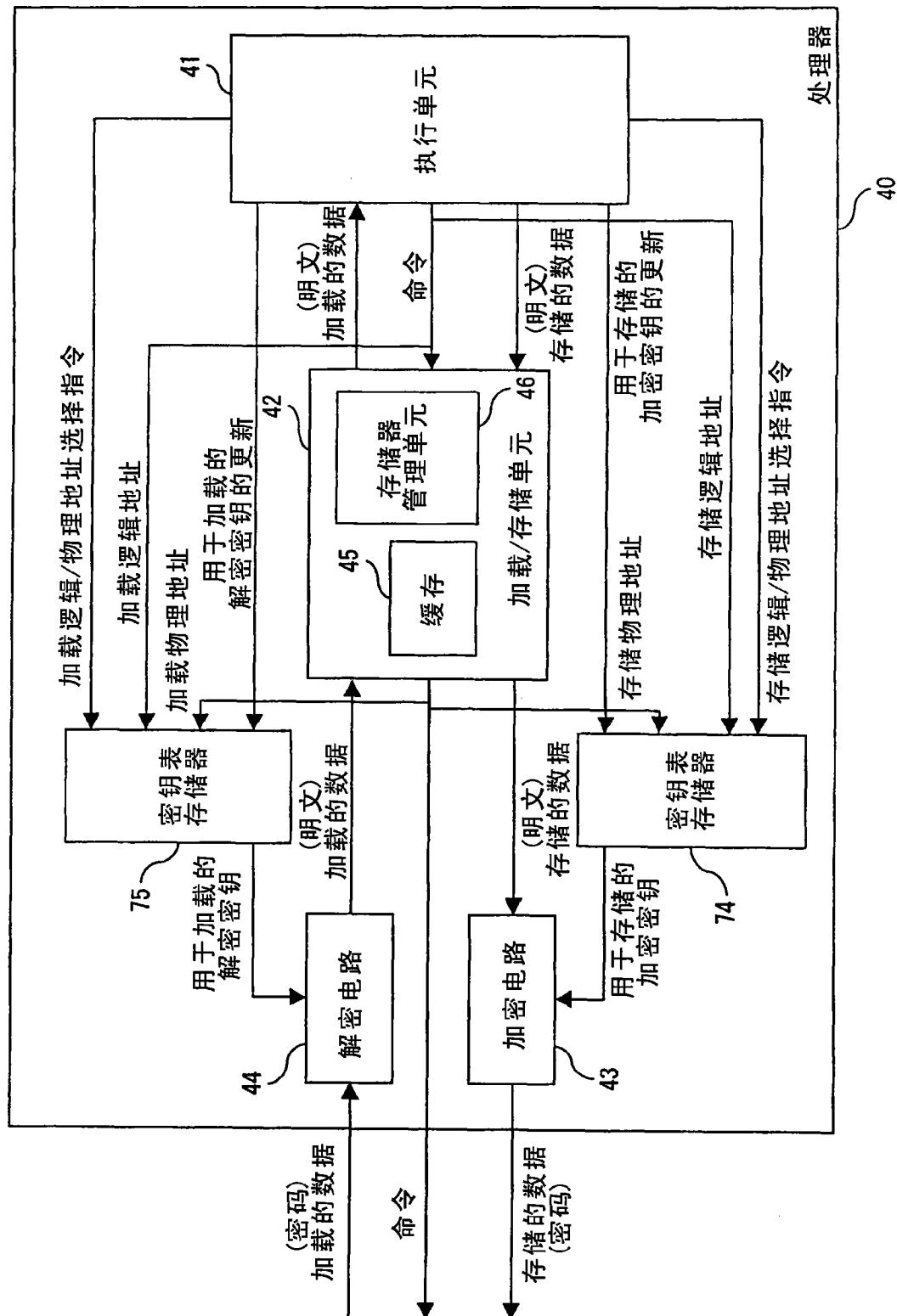
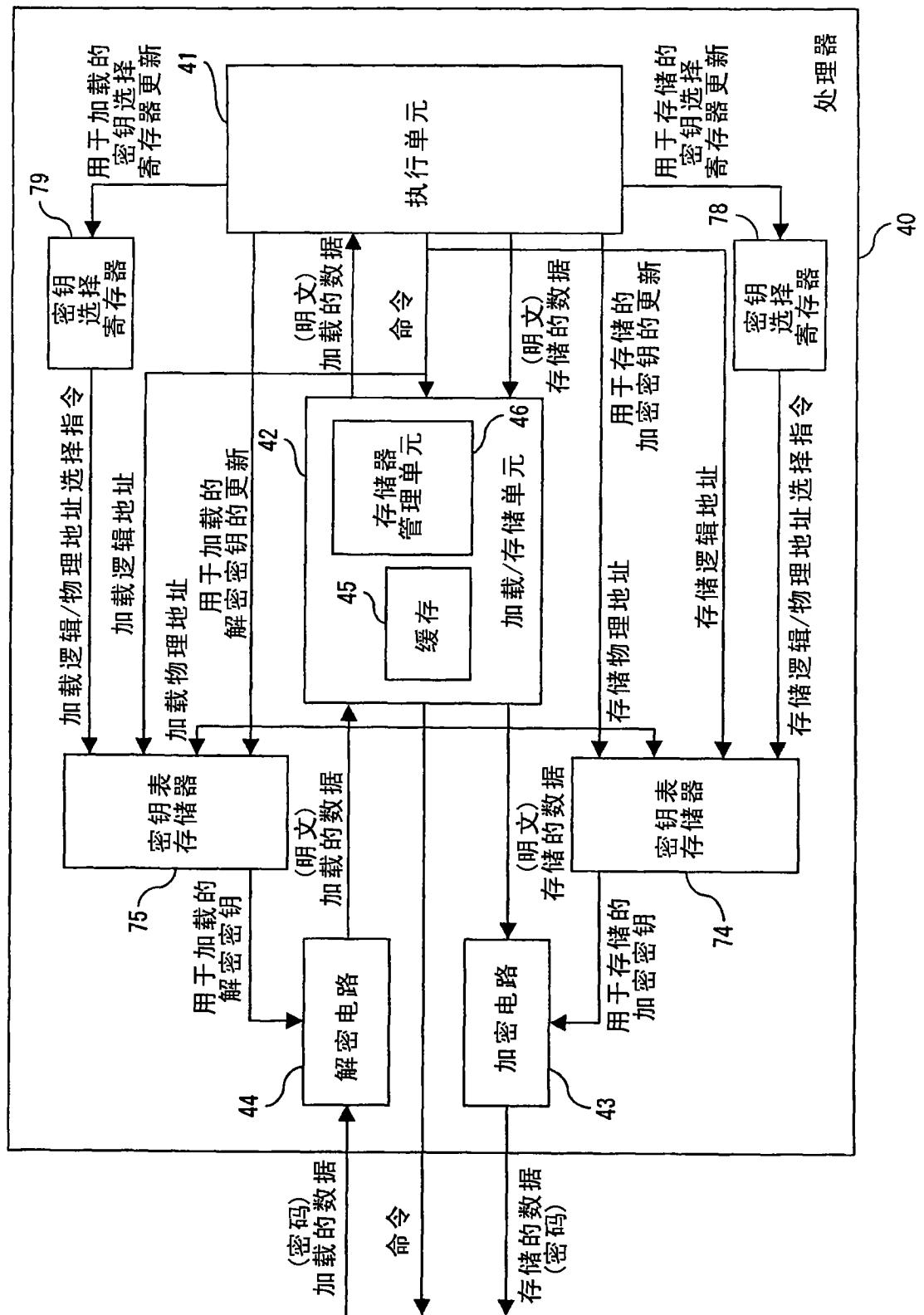


图 35



36

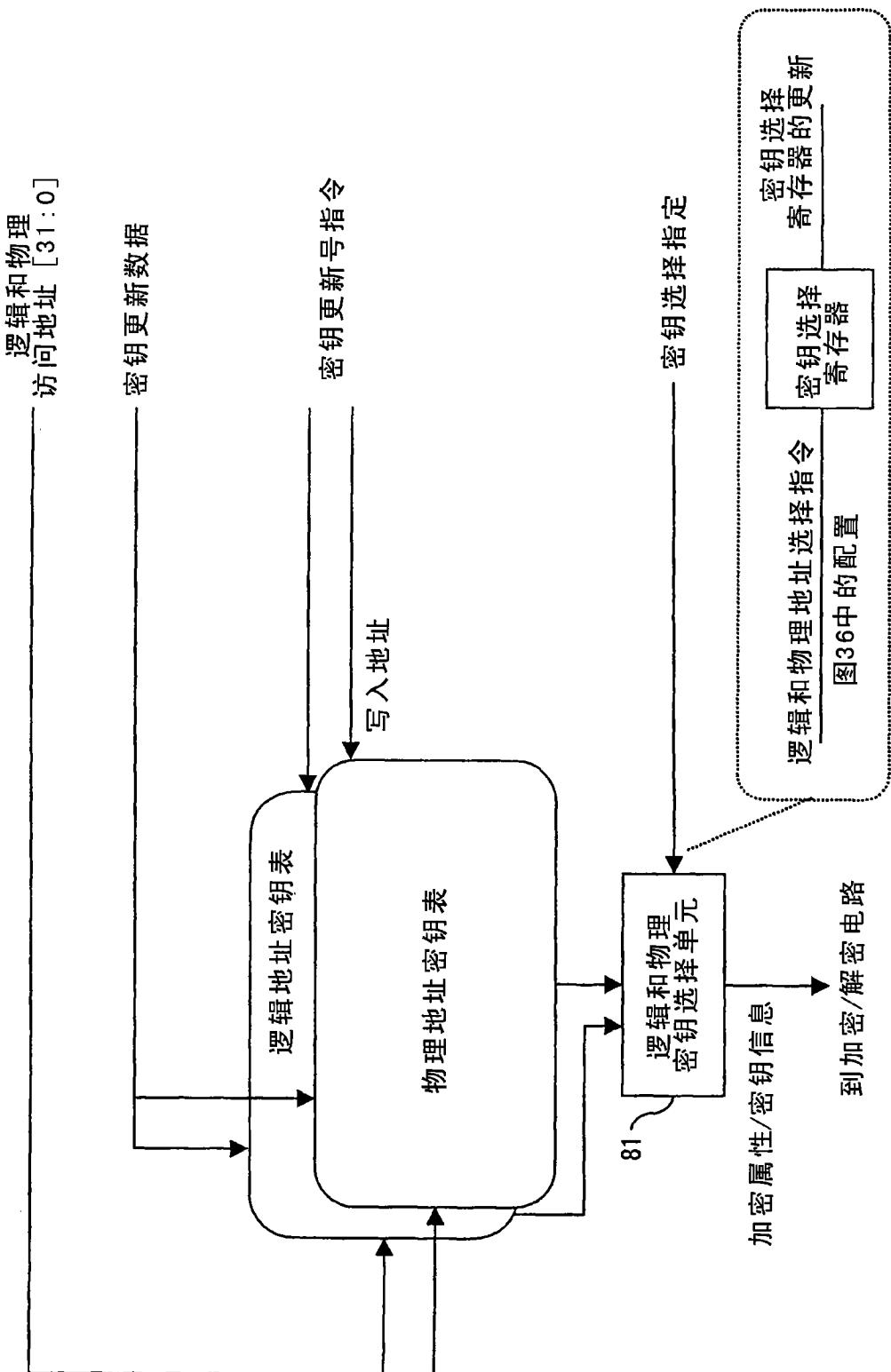


图37

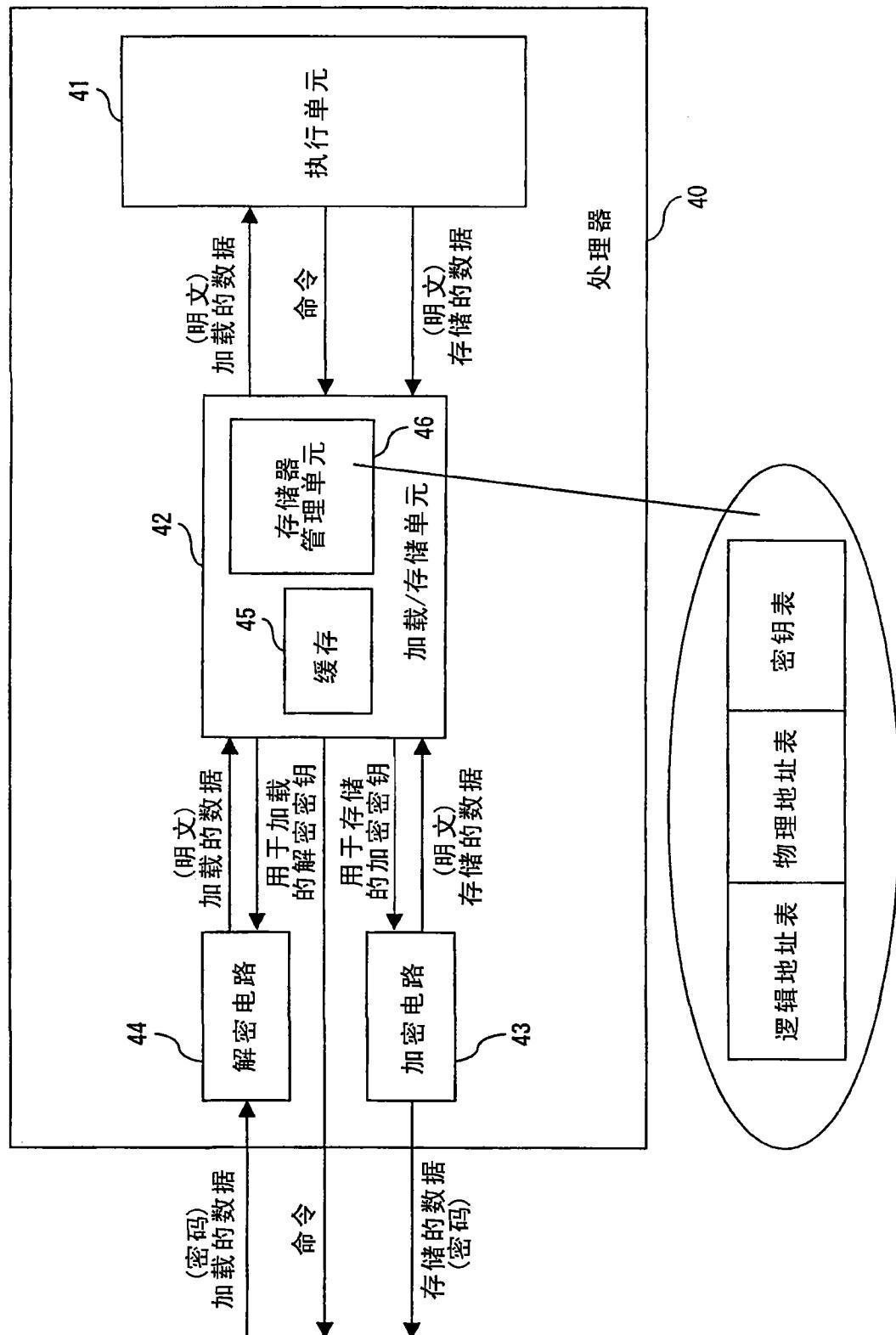


图 38

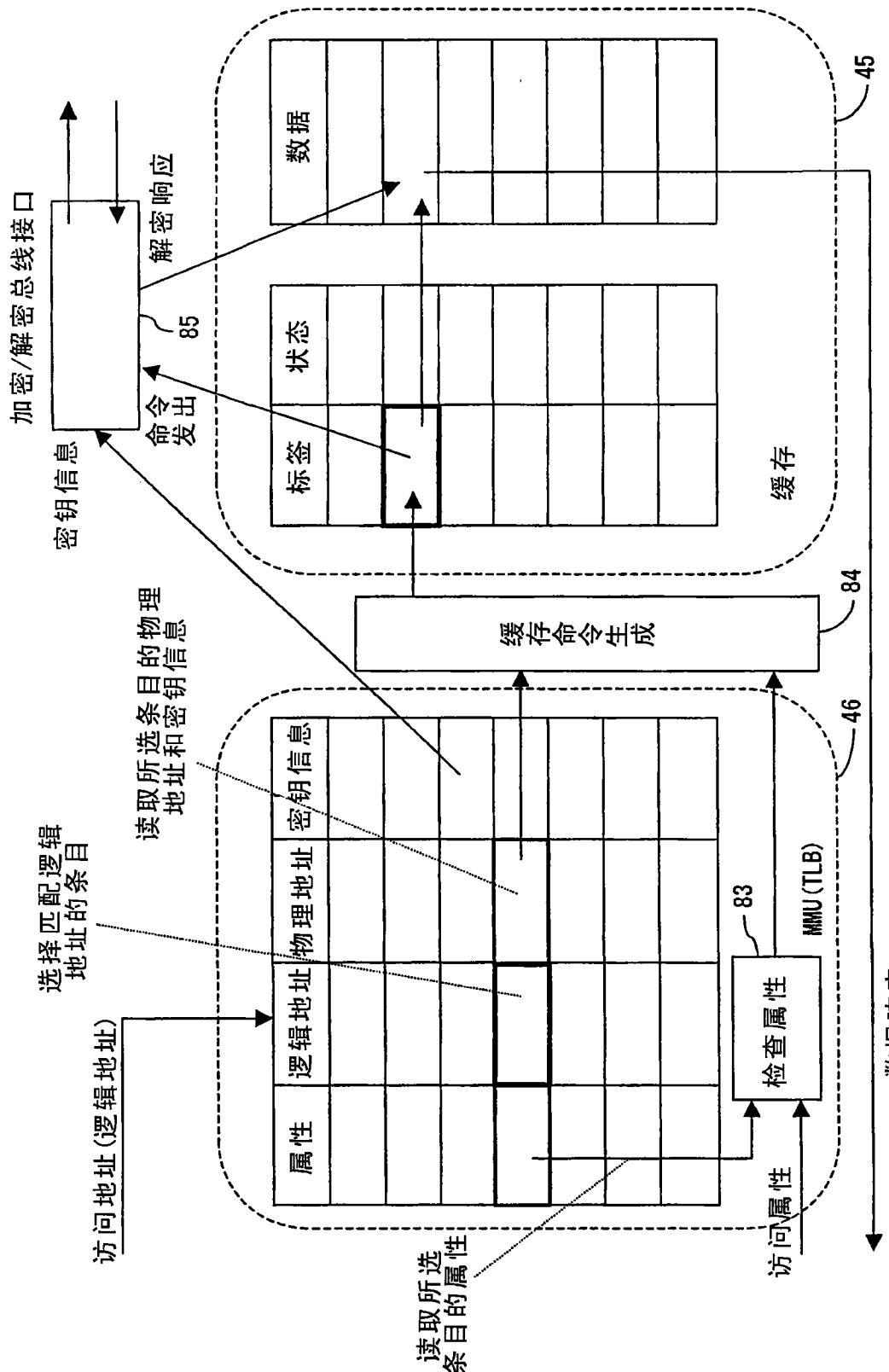
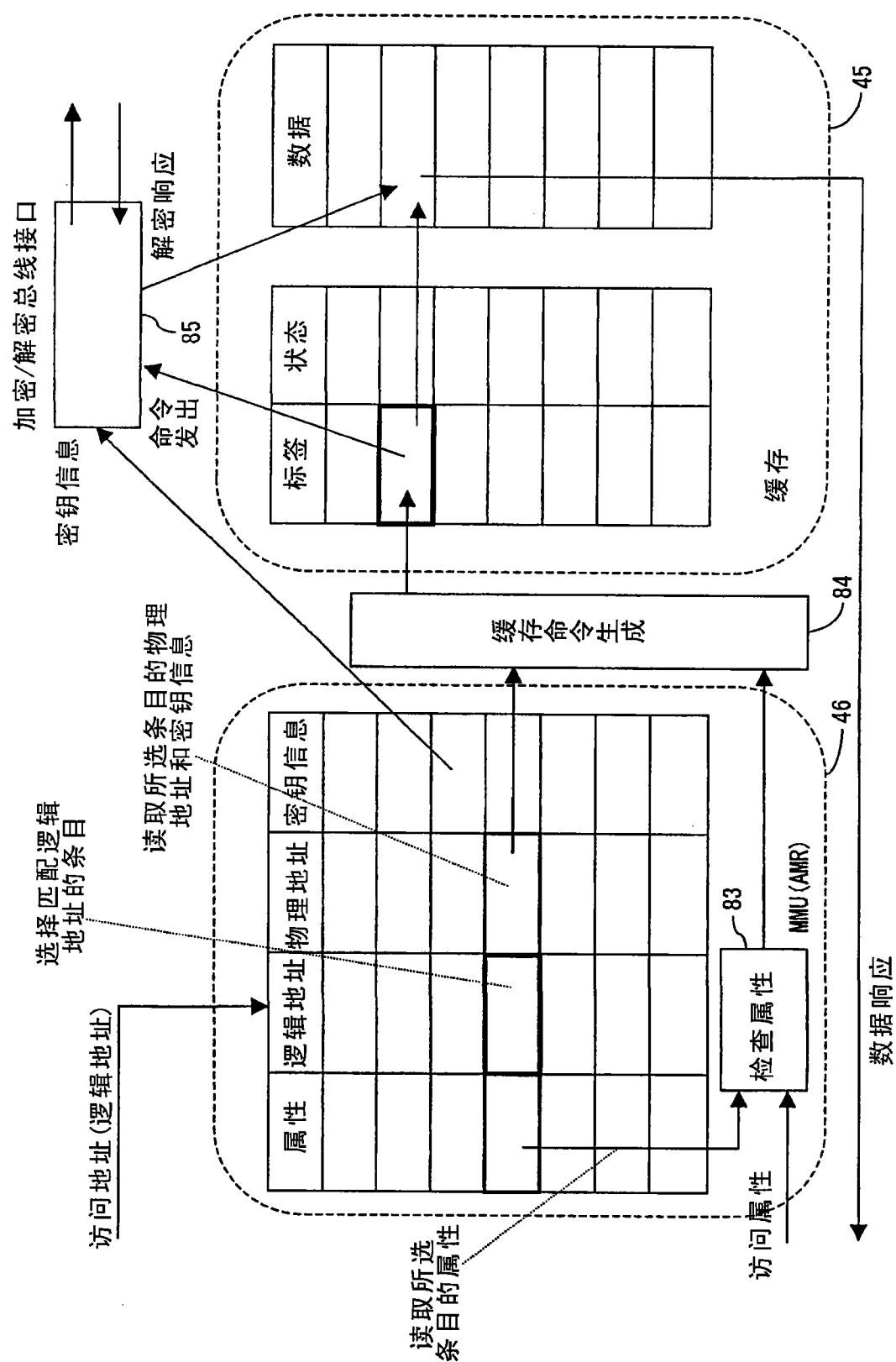


图39



冬 40

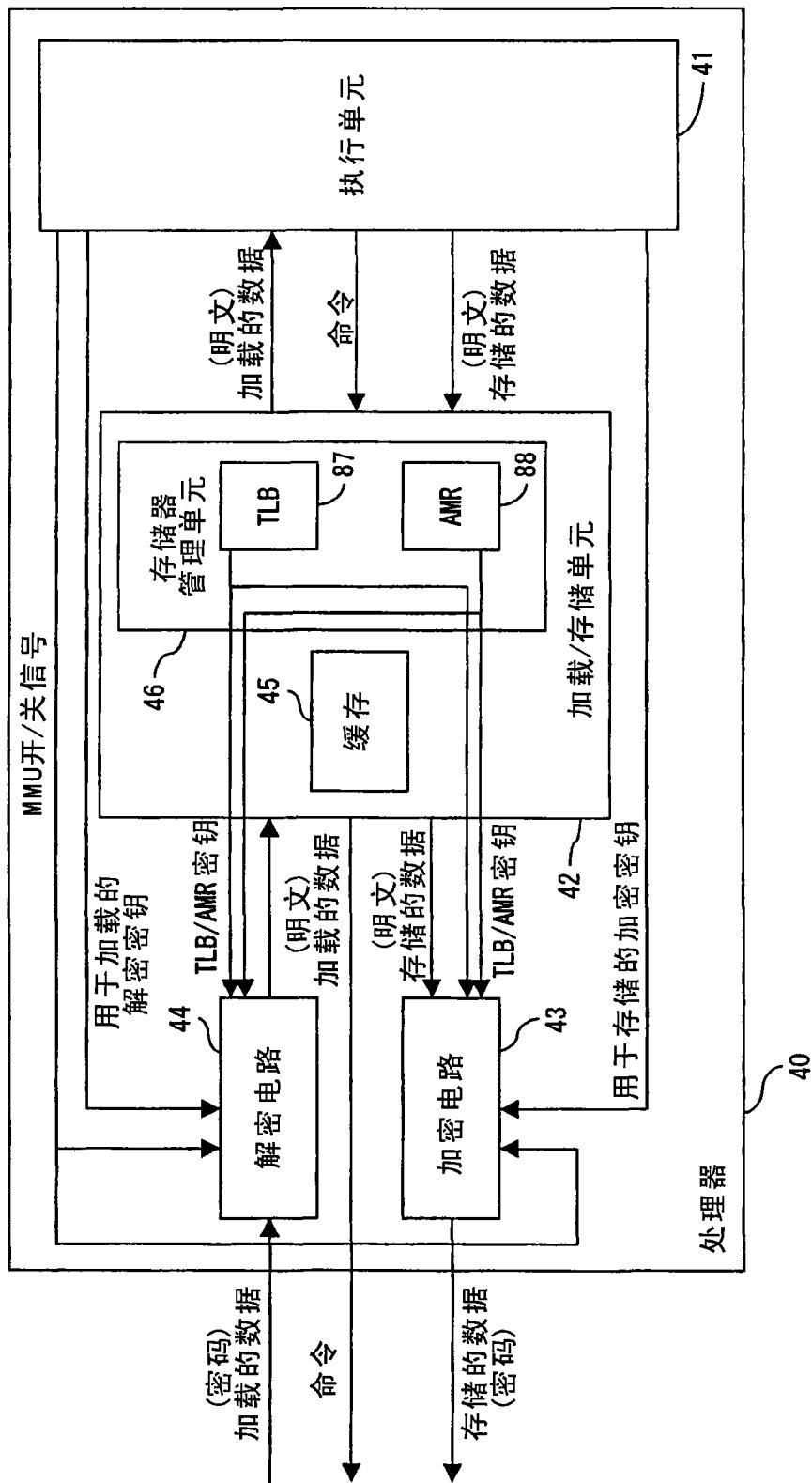


图 4-1

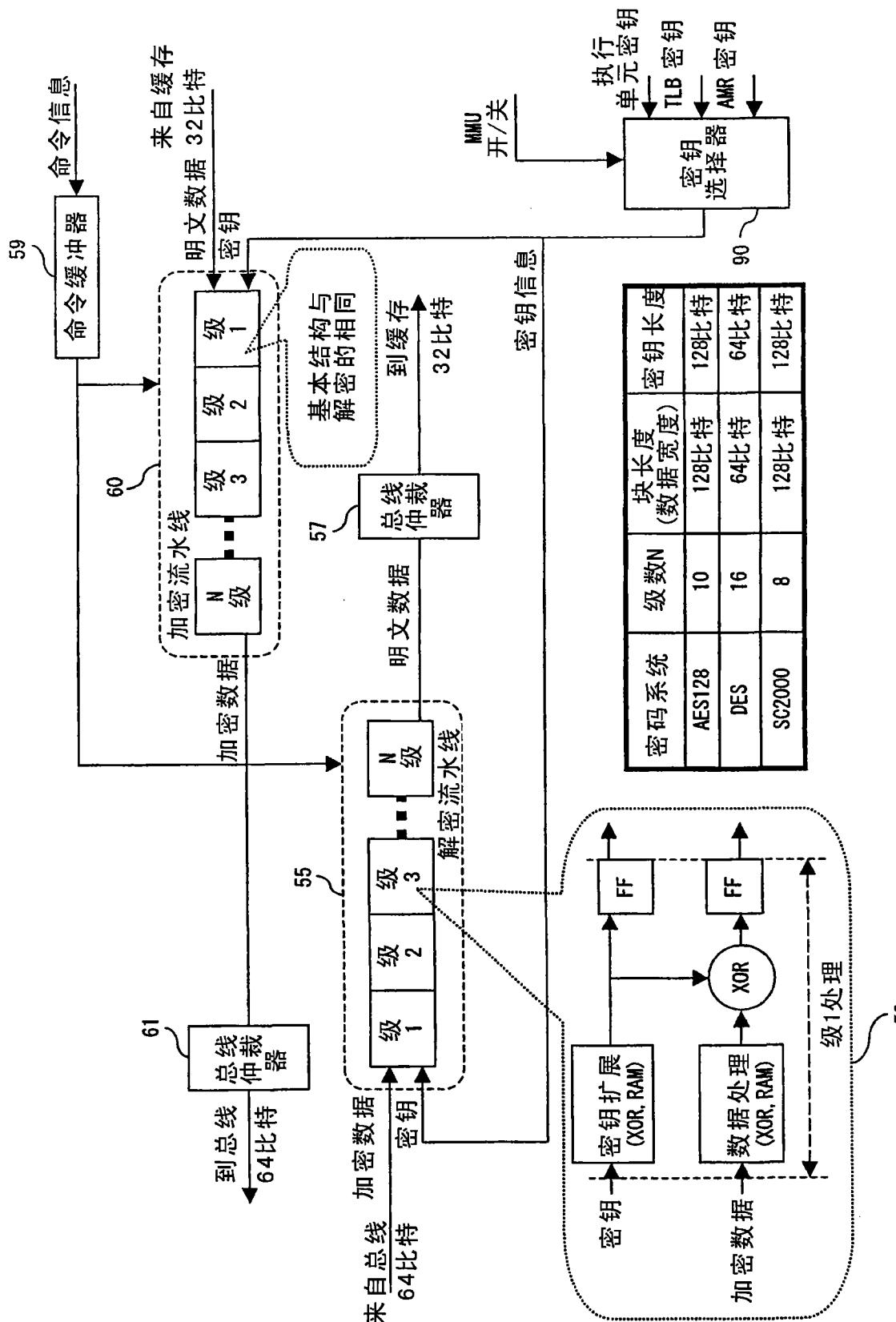


图42

项目	方向	图号												说明		
		20	21	22	23	24	21+22+23	27	29	30	31	35	36	38	40	41
加载解密钥	输出	○														○
加载数据	输入	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
加载密钥号指令	输出		○	○	○	○	○	○	○	○	○	○	○	○	○	○
命令取得数据	输入		○	○	○	○	○	○	○	○	○	○	○	○	○	○
存储加密钥	输出	○														○
存储数据	输出	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
存储密钥号指令	输出		○	○	○	○	○	○	○	○	○	○	○	○	○	○
命令	输出	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
执行状态信号	输出		○	○	○	○	○	○	○	○	○	○	○	○	○	○
寄存器写入信号	输出	○	○	○	○	○	○	○	△	○	○	○	○	△	○	○
寄存器读取数据	输入	○	○	○	○	○	○	○	○	○	○	○	○	△	○	○
寄存器写入数据	输出		○	○	○	○	○	○	○	○	○	○	○	△	○	○
访问地址	输出	△	△	△	△	△	△	○	○	○	○	○	○	○	○	○
加载/存储状态信号	输出	△	△	△	△	△	△	△	△	○	○	○	○	△	○	○
密钥选择指令	输出													○	○	○
MMU状态信号	输出													○	○	○
管理员/用户状态信号	输出													○	○	○
上下文ID数据	输出													○	○	○

图43

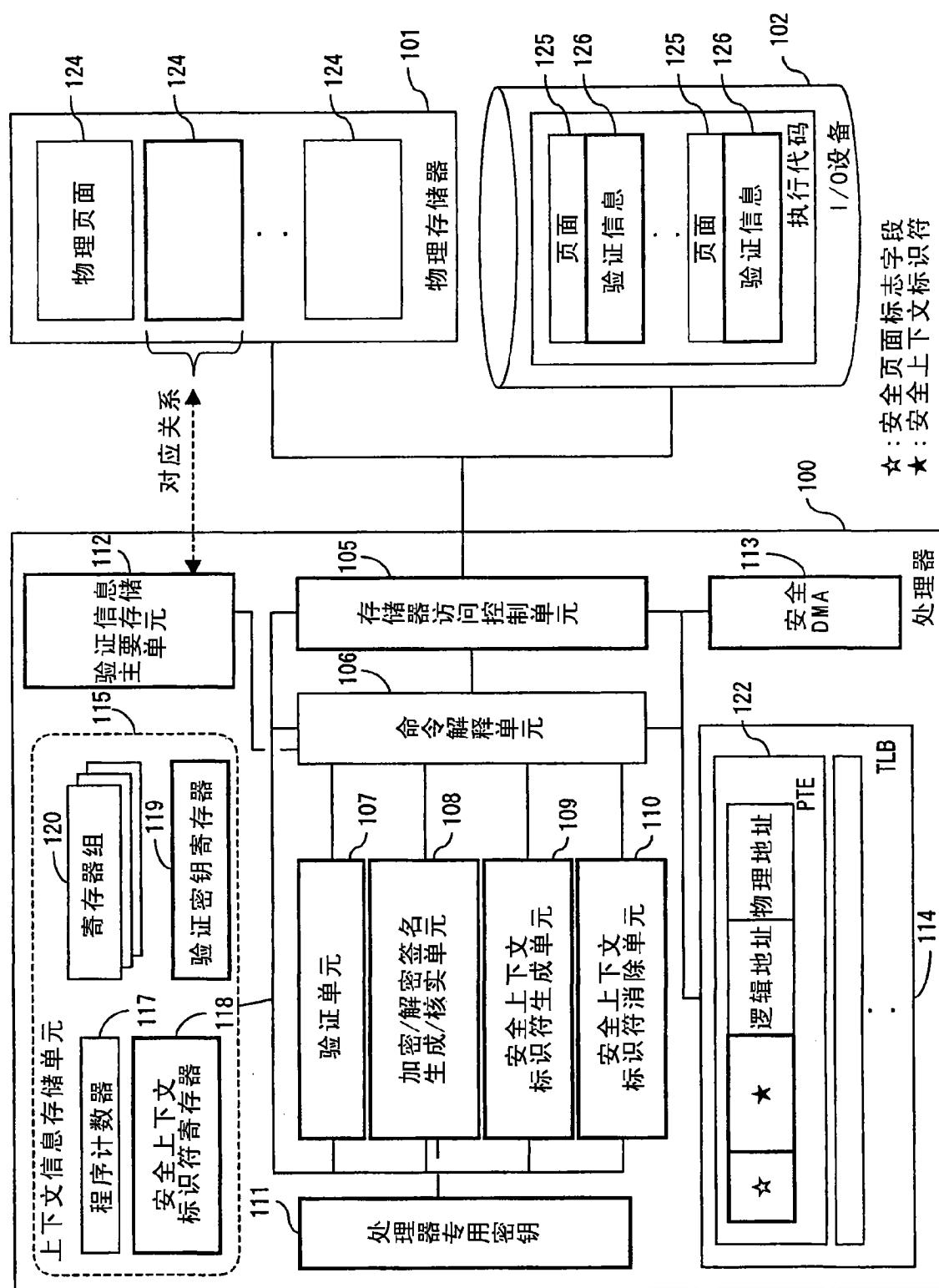


图44

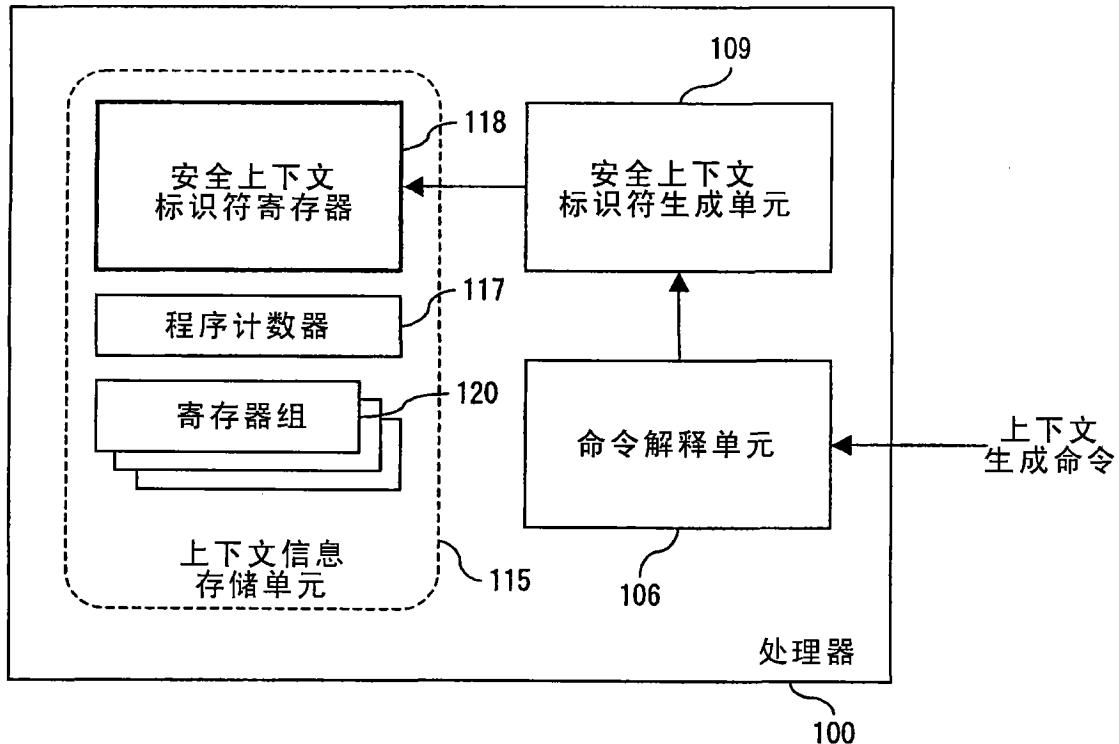


图 45

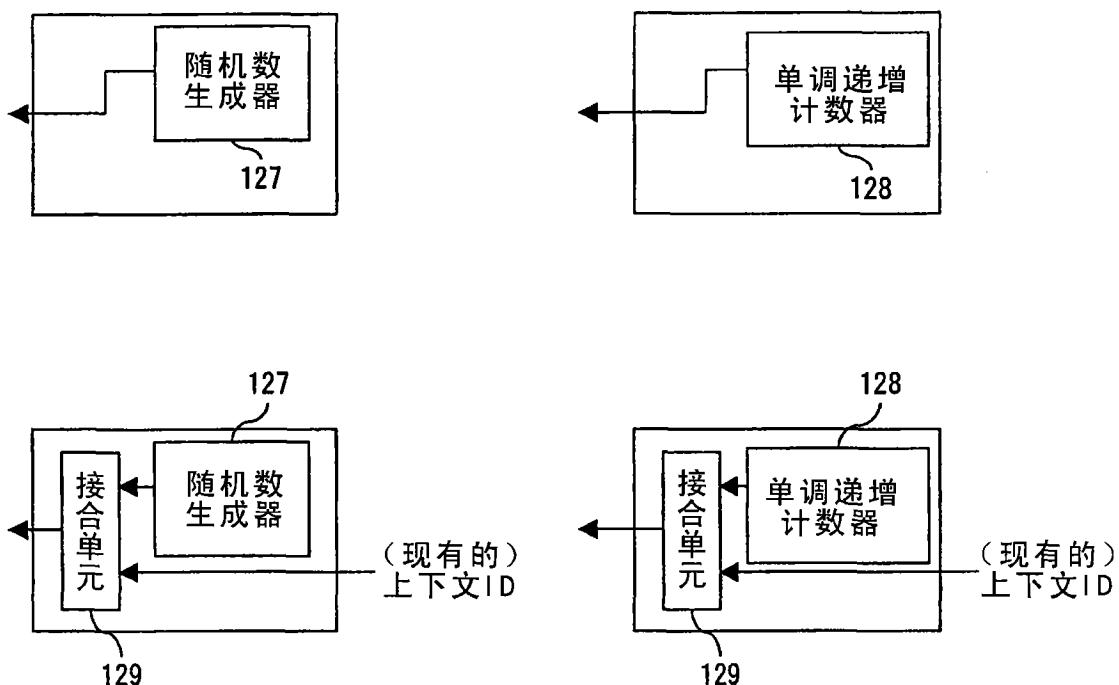


图 46

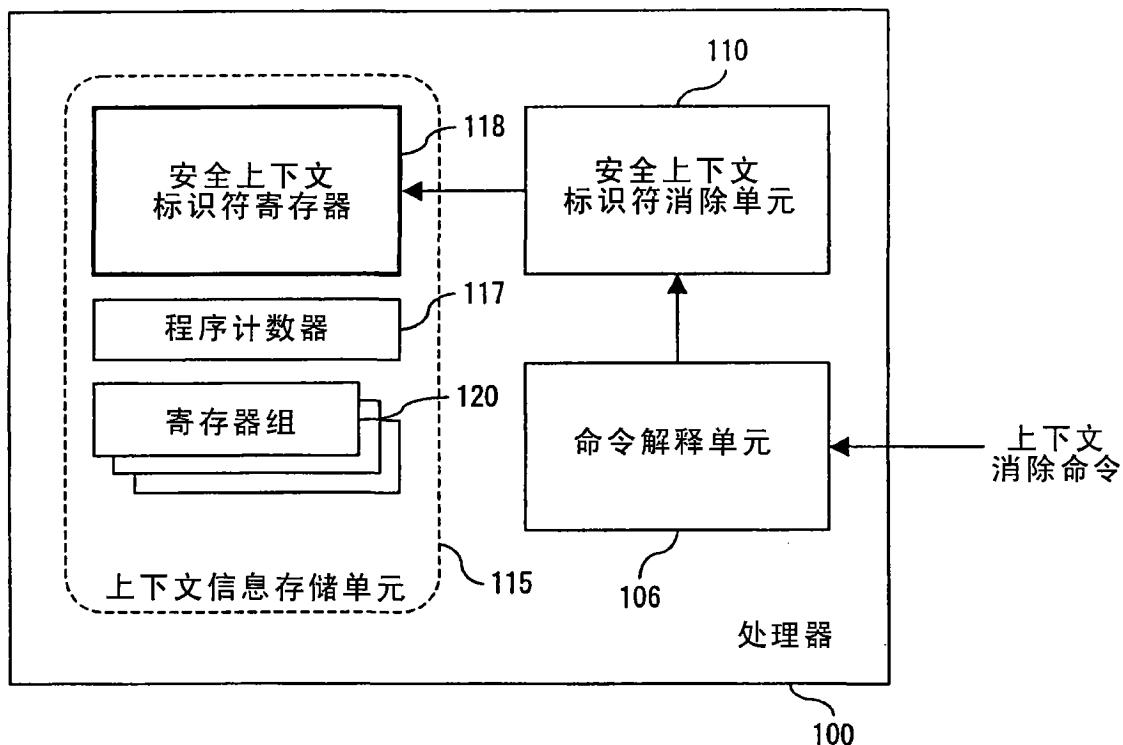


图 47

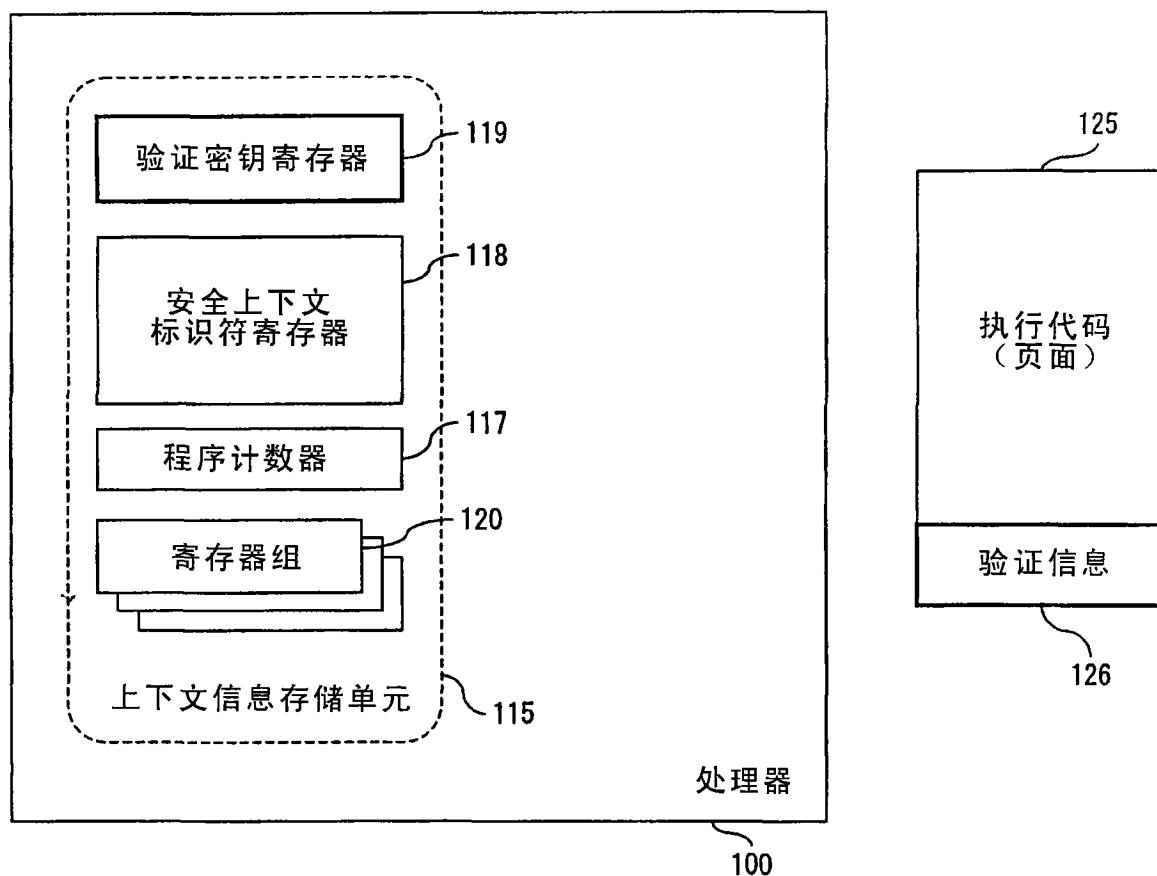


图 48

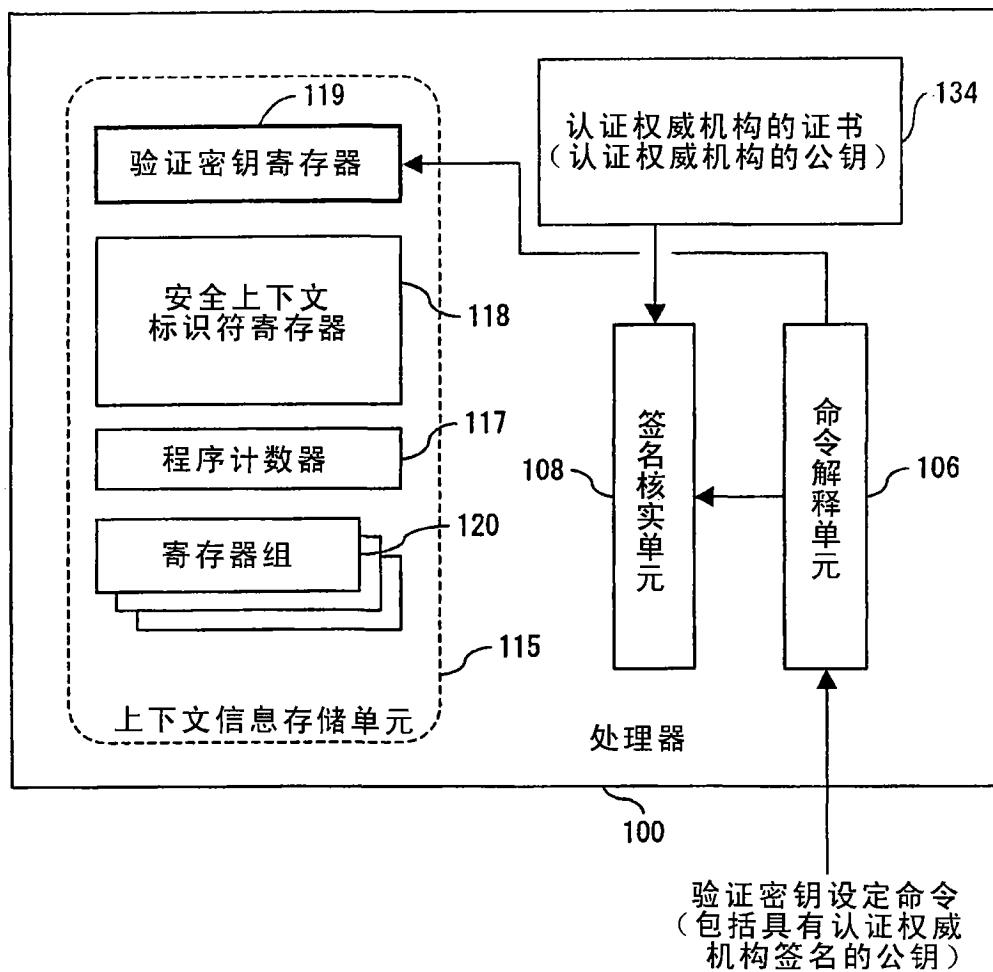


图 49

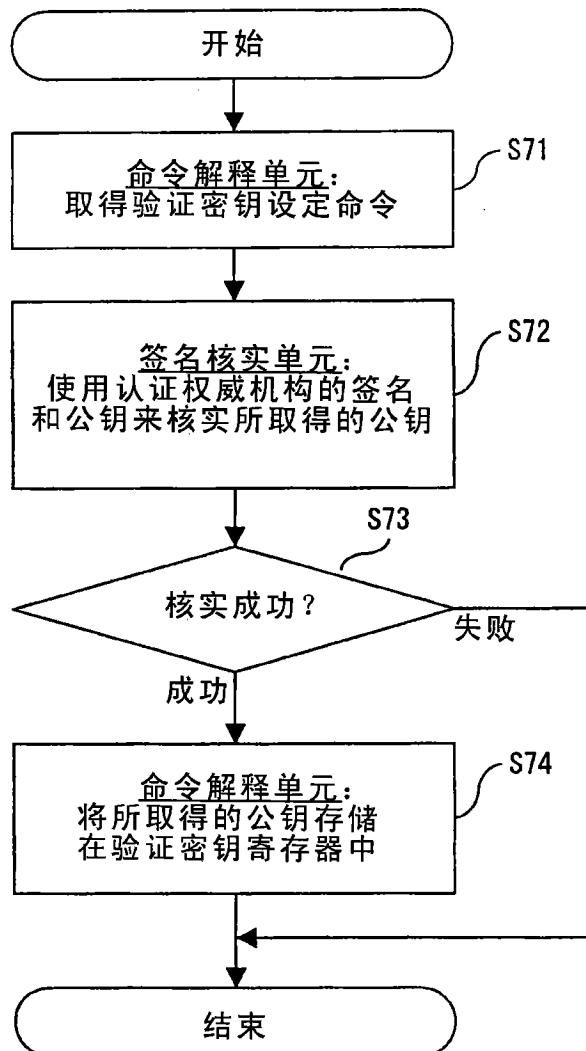


图 50

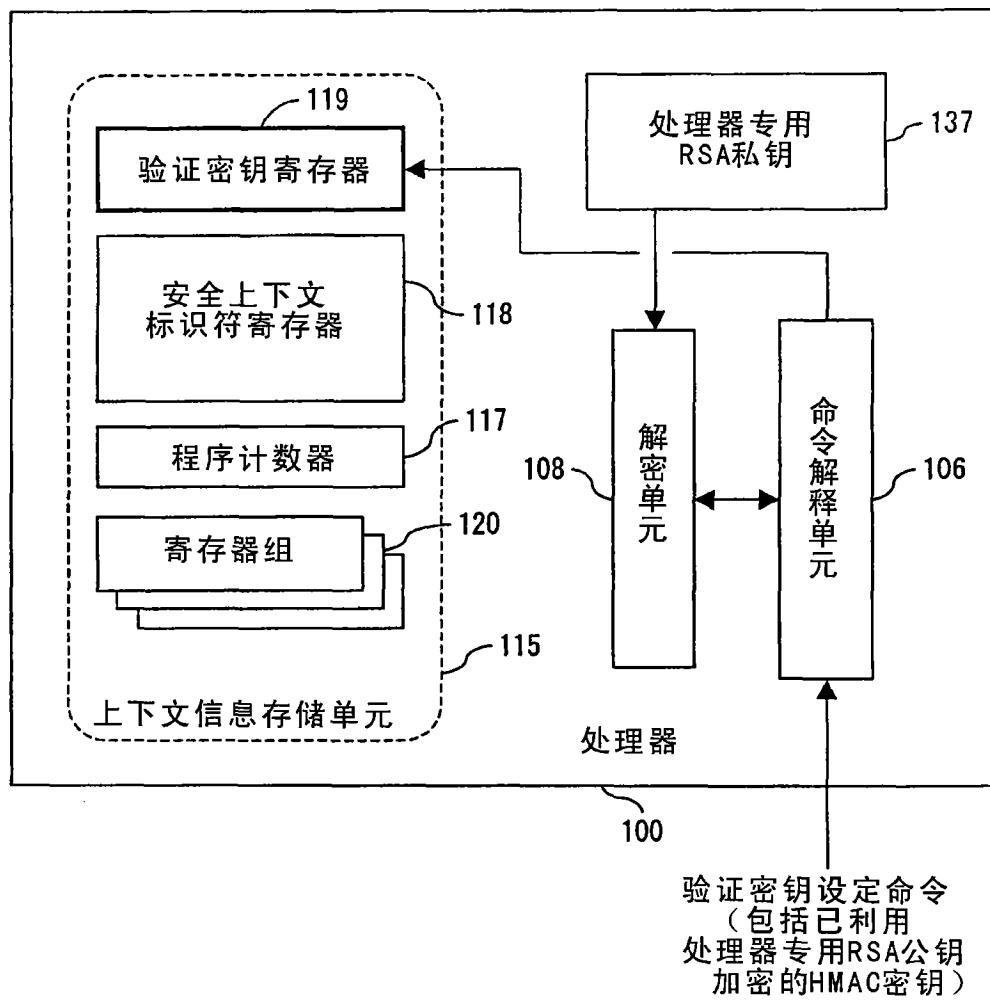


图 51

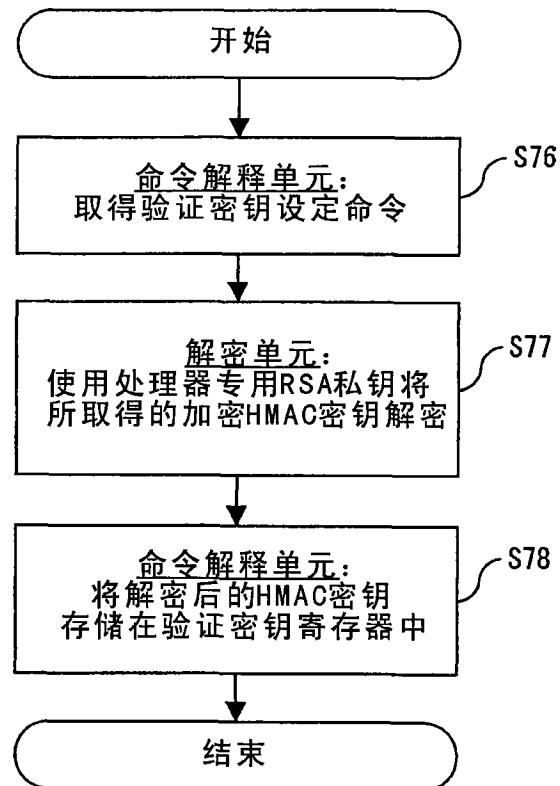


图 52

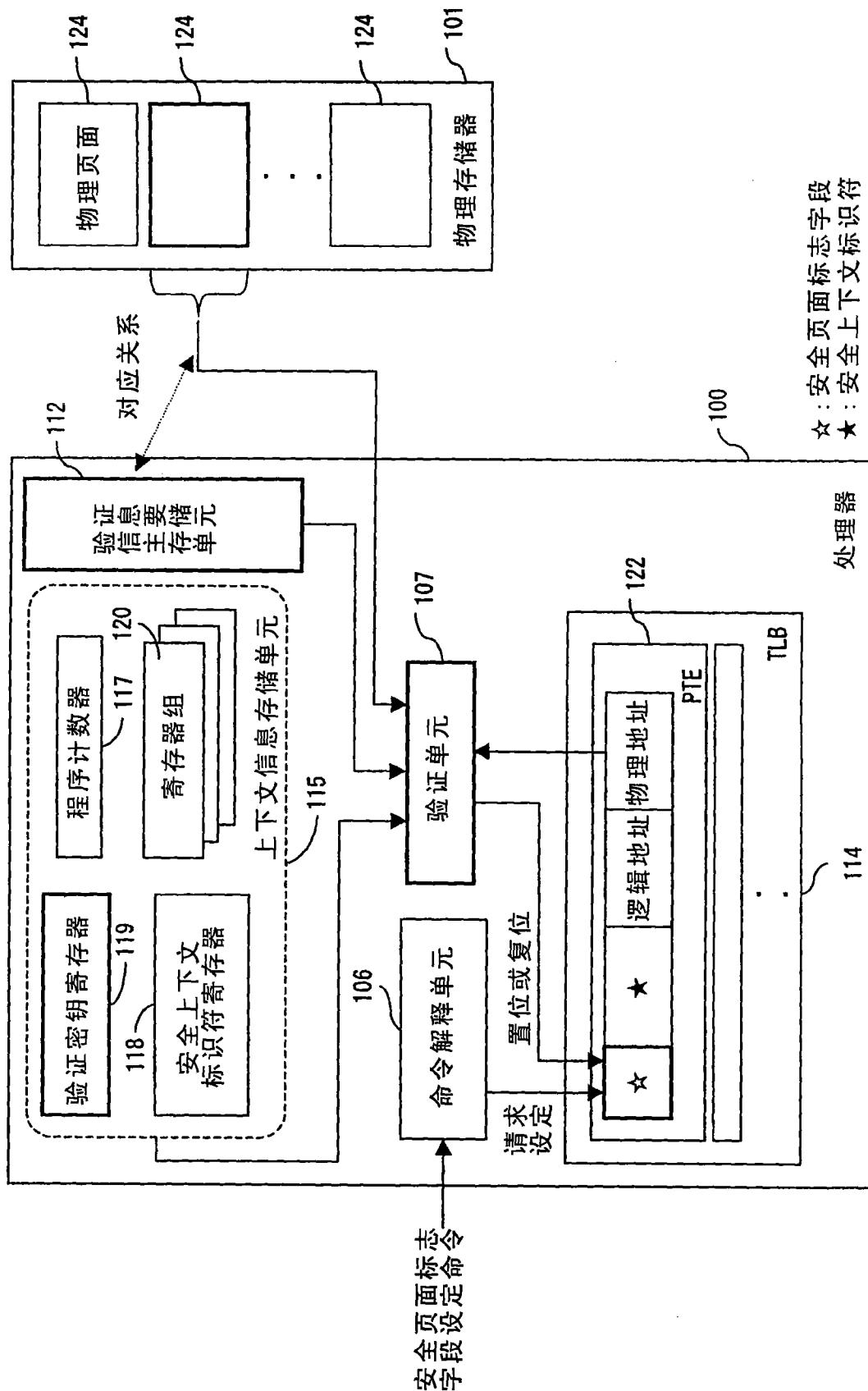


图 5.3

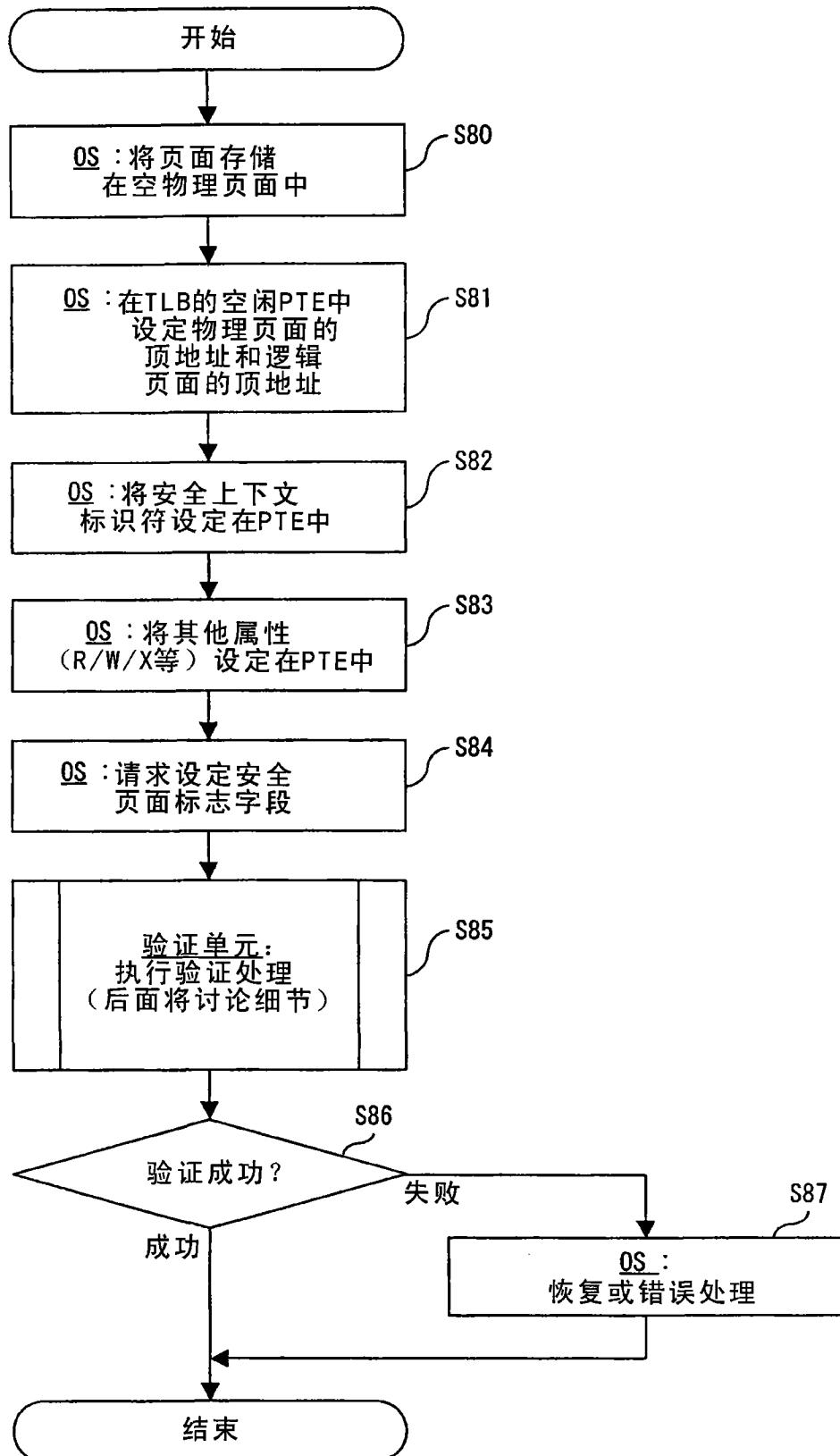


图 54

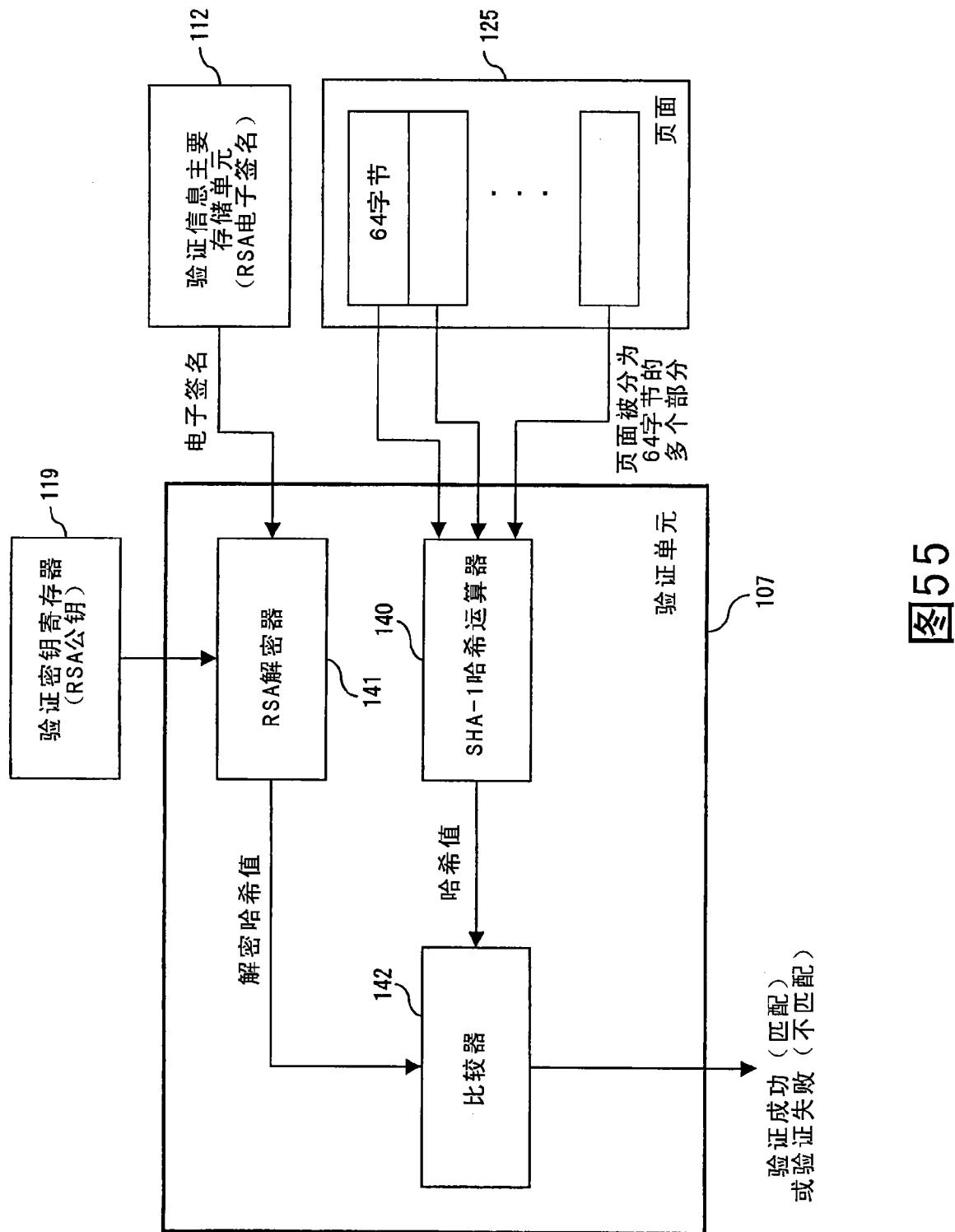


图55

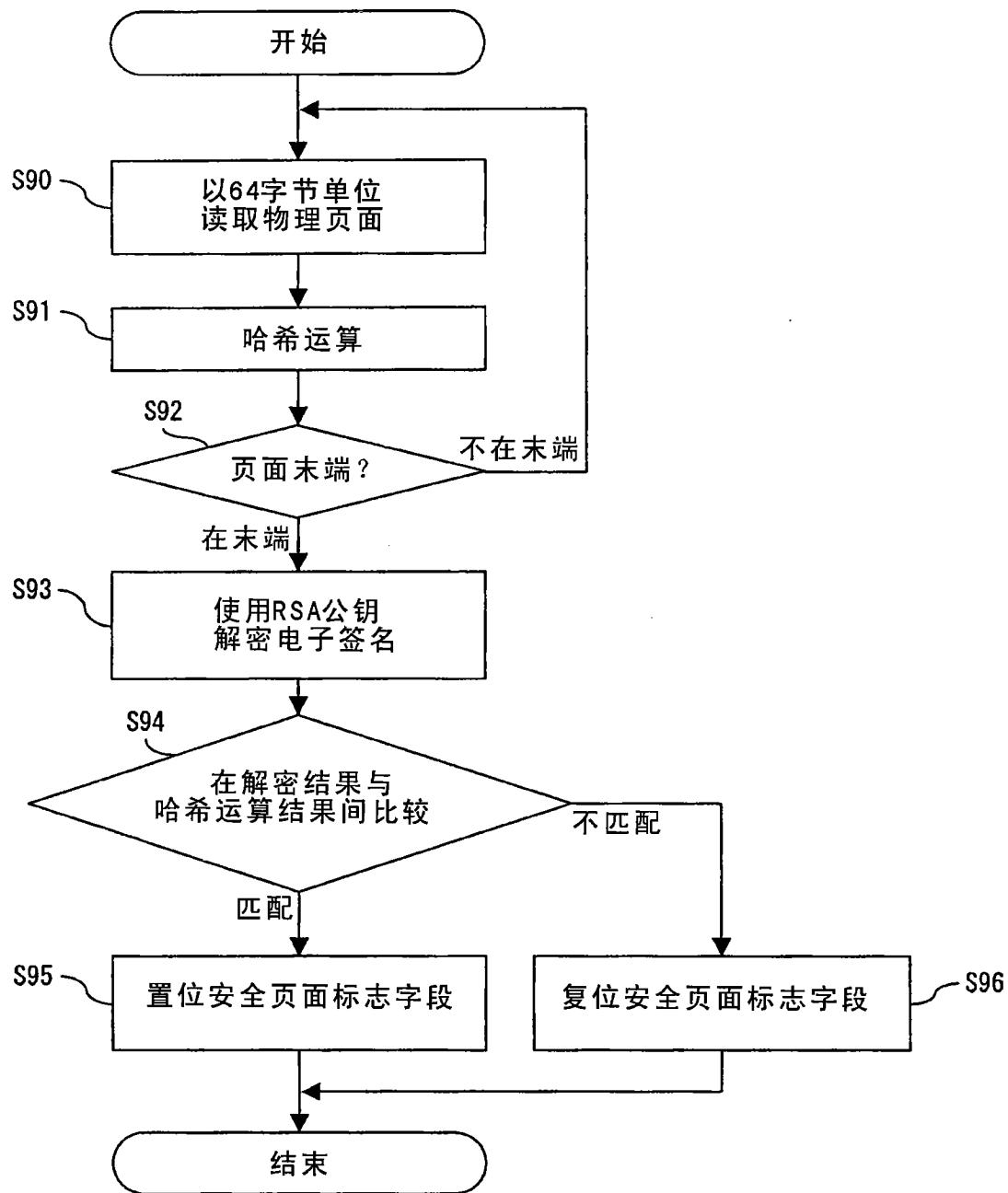


图 56

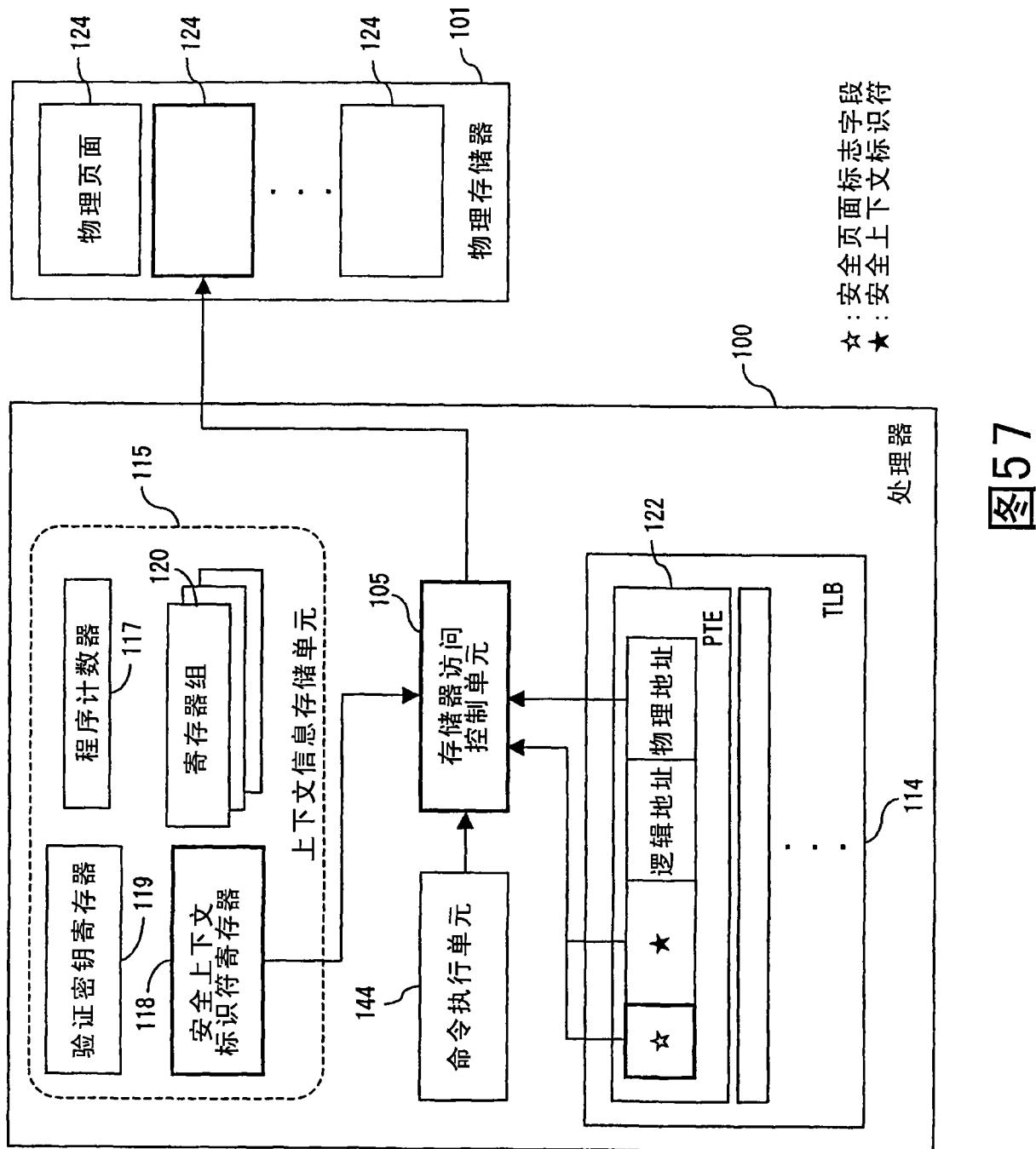
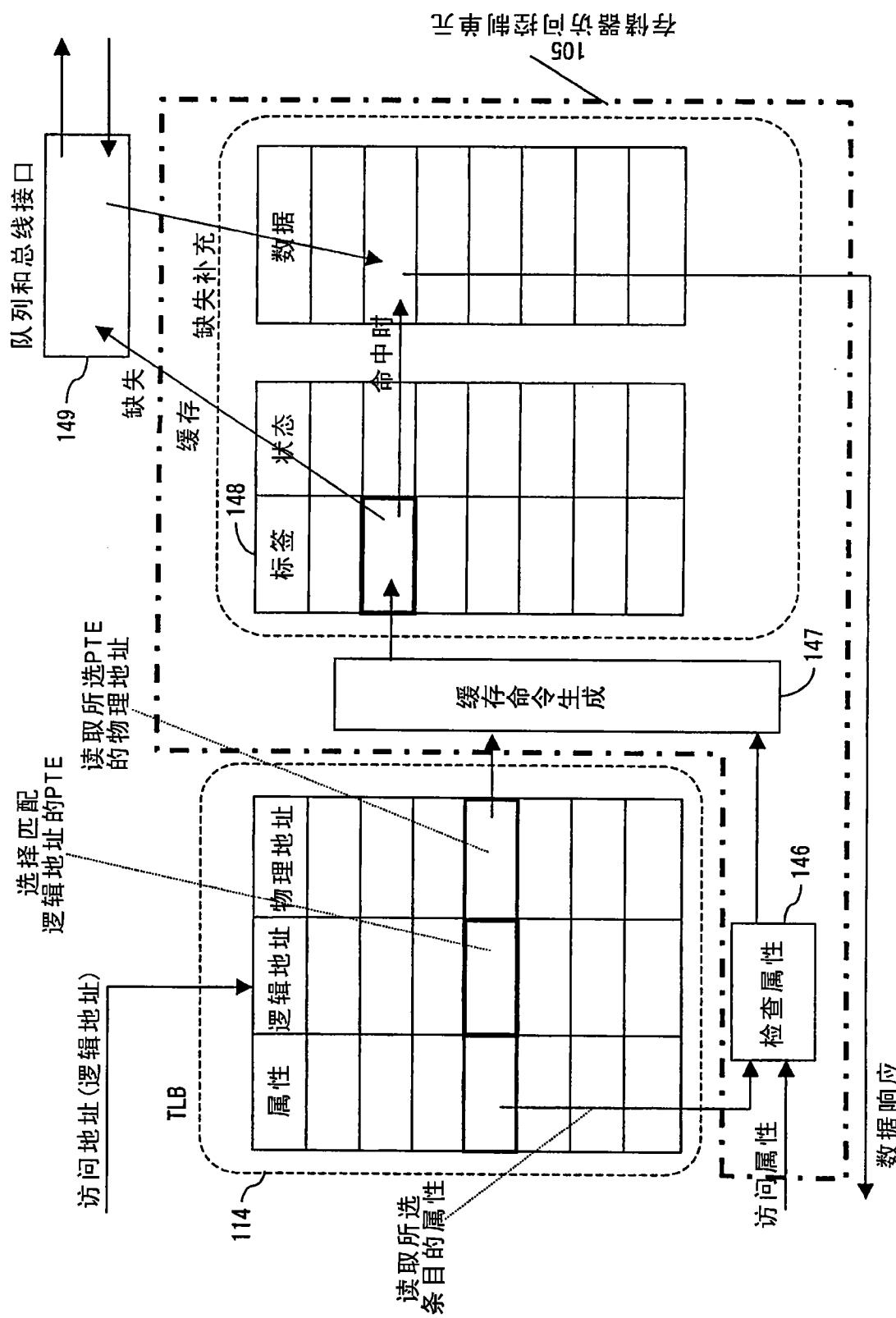


图 57



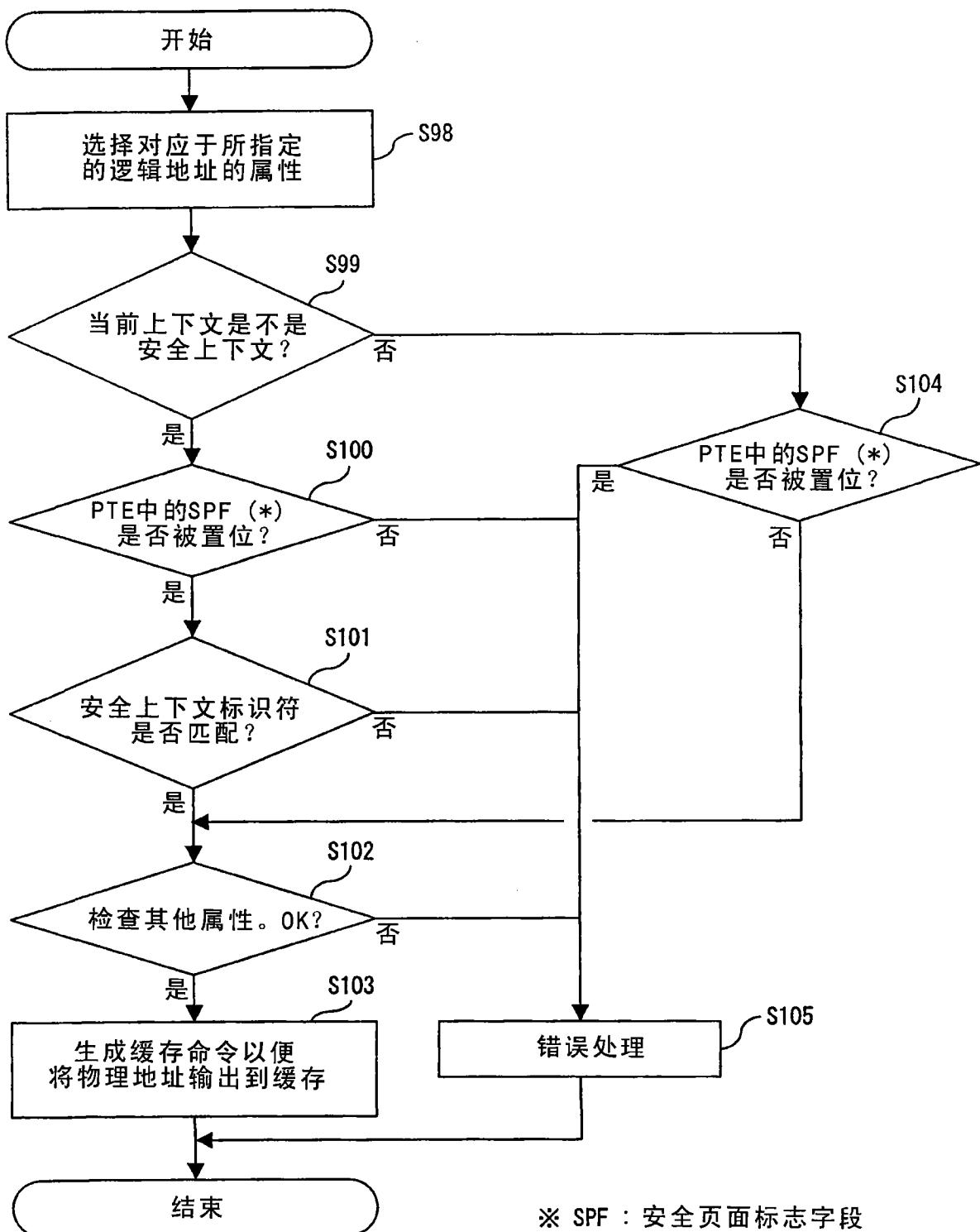


图 59

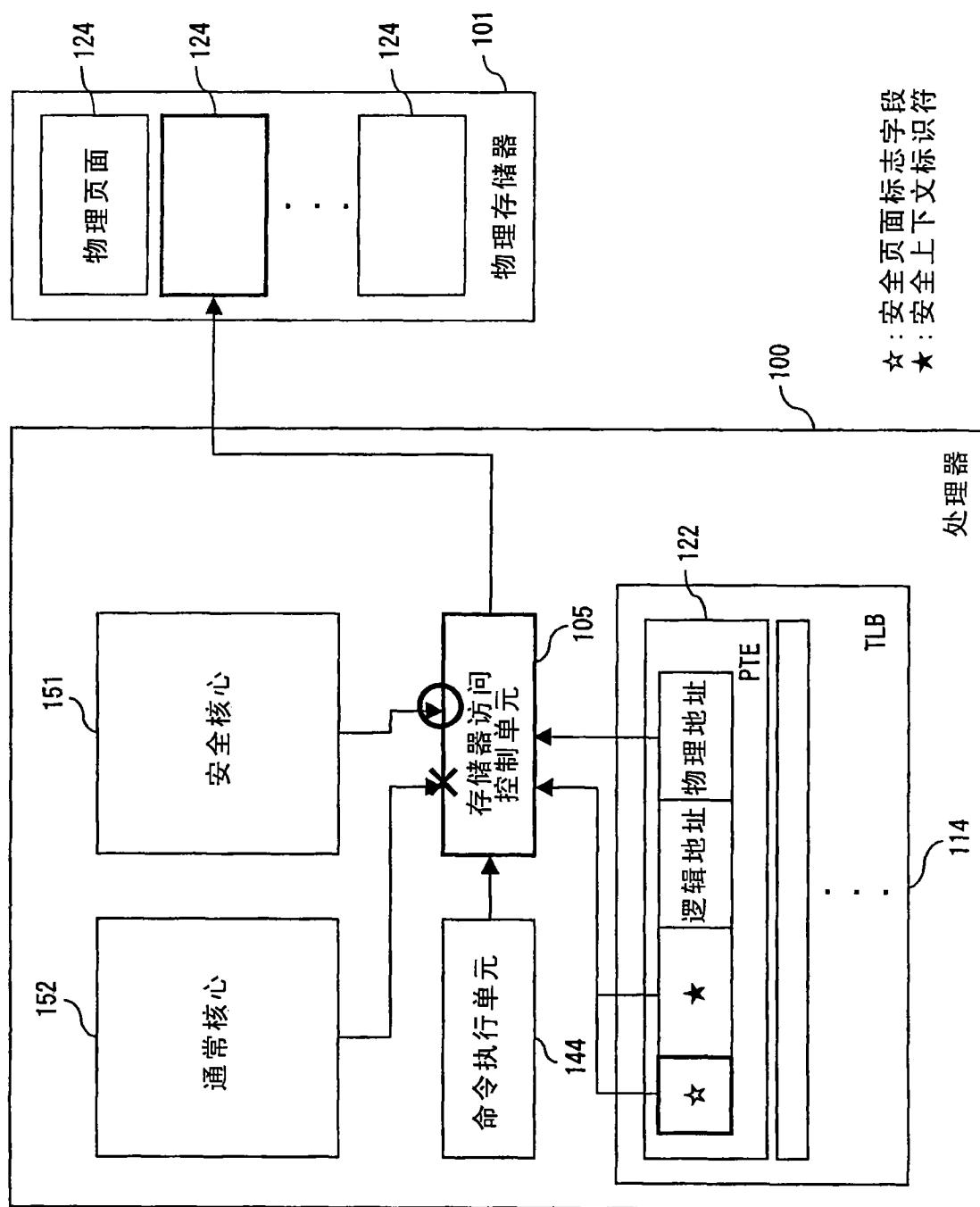
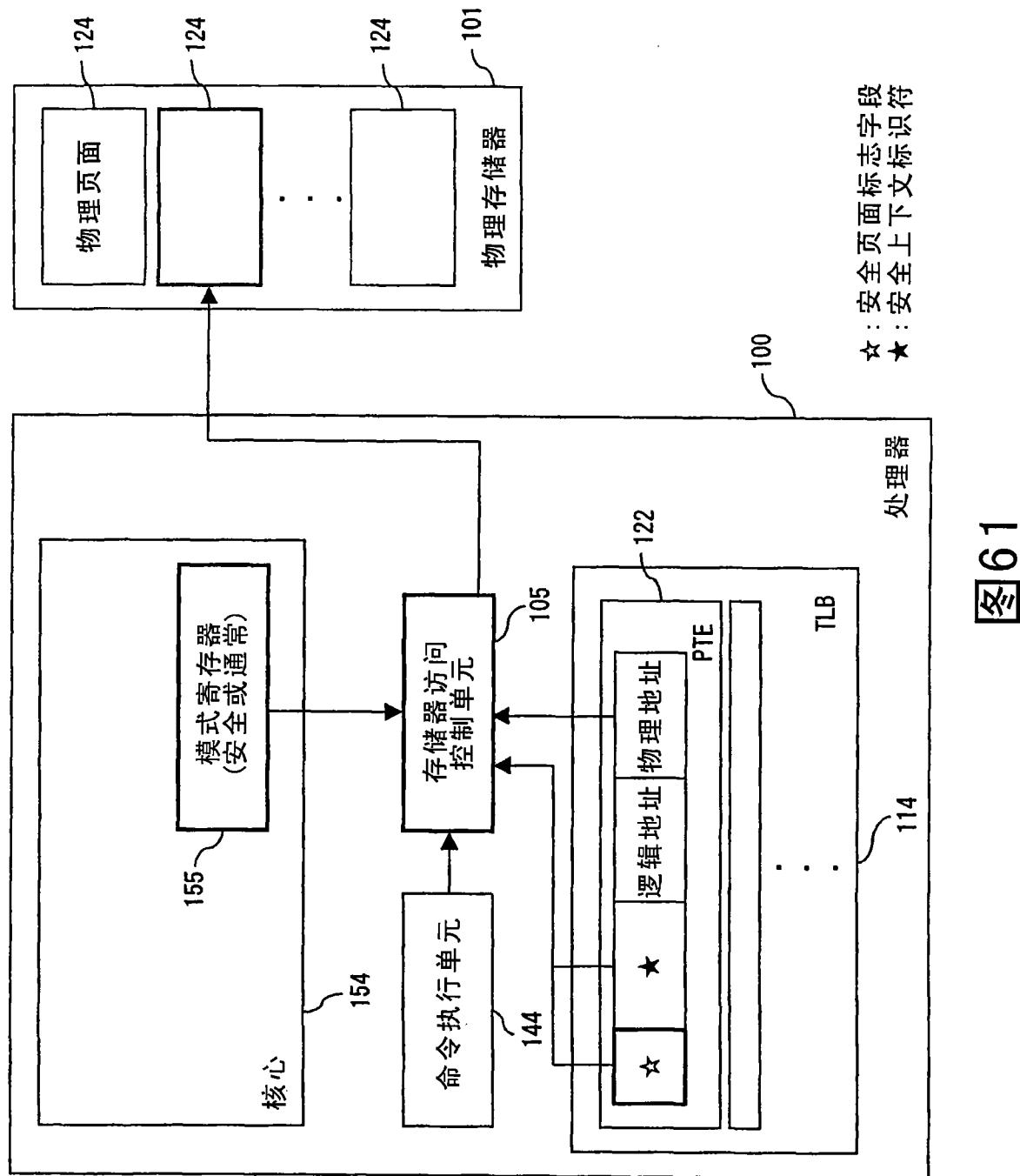


图 60



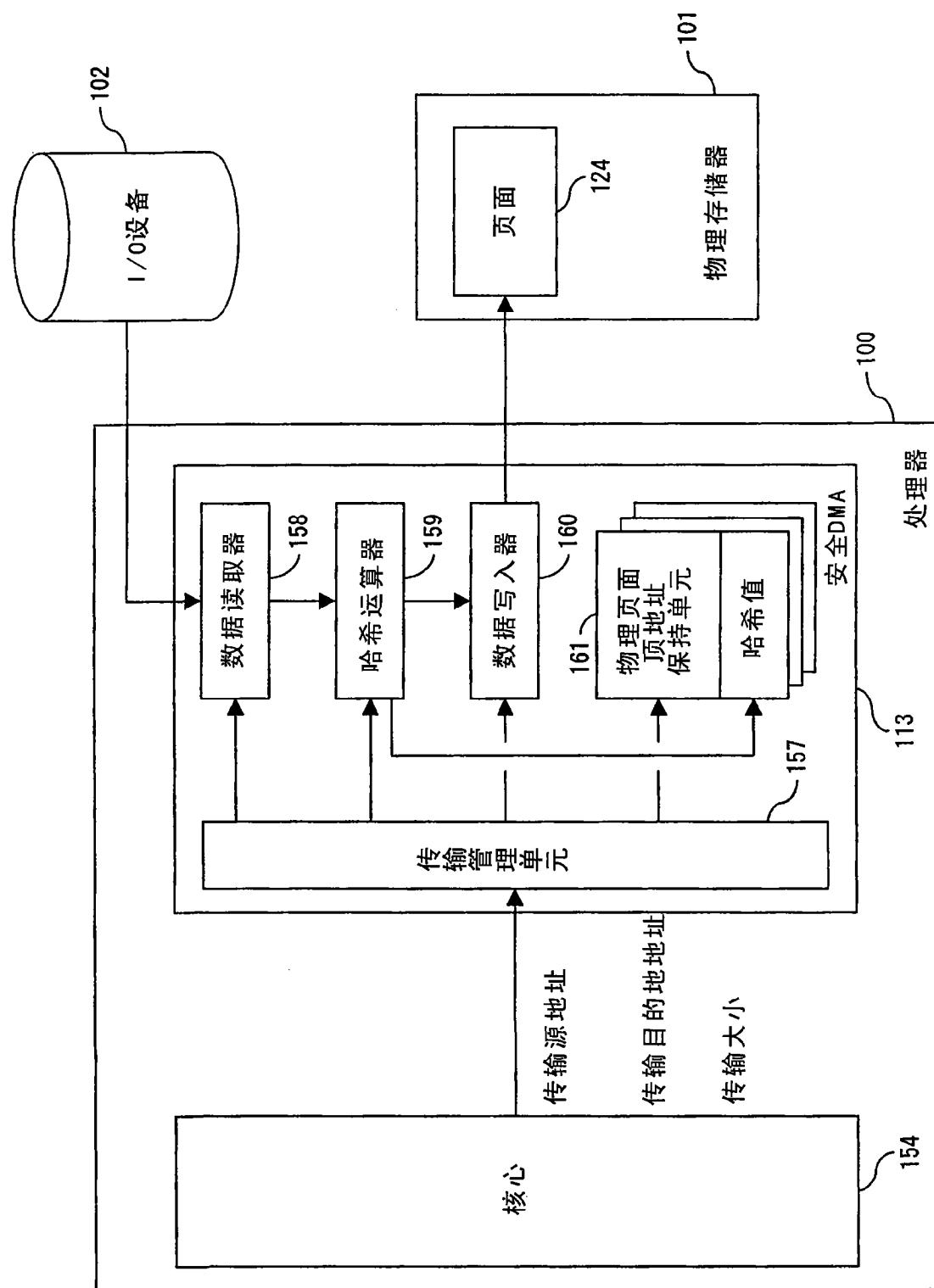


图 62

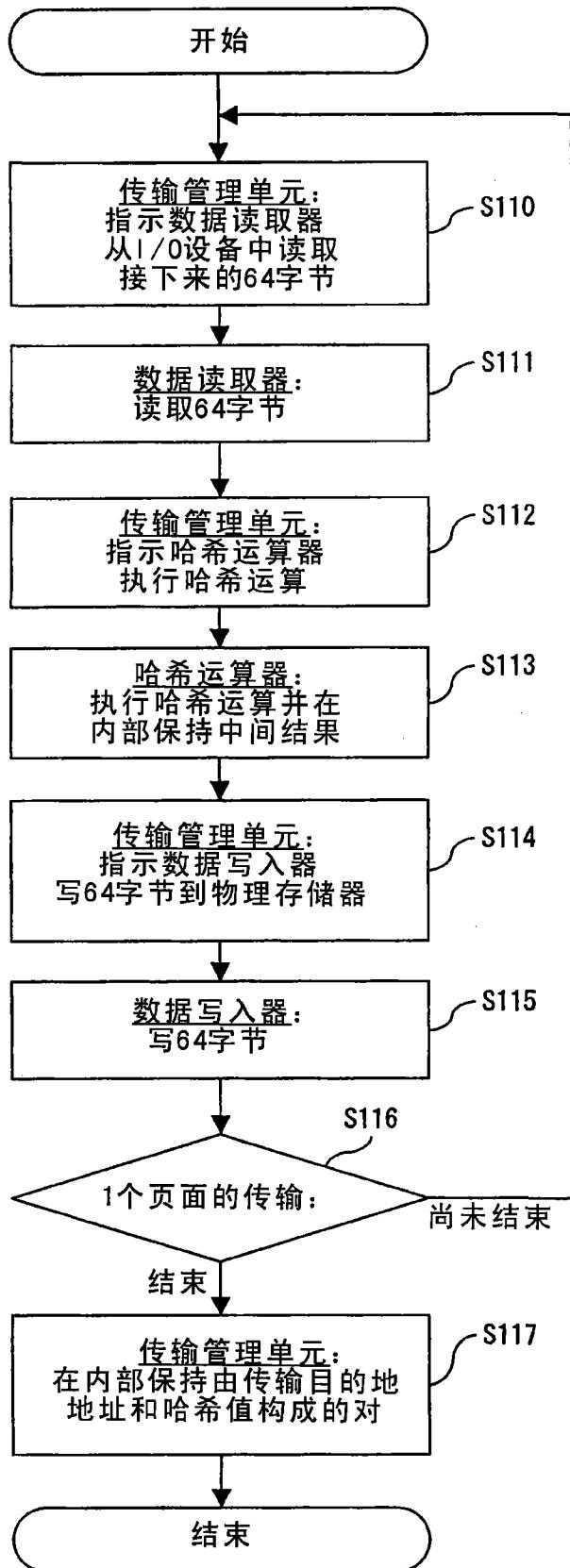


图 63

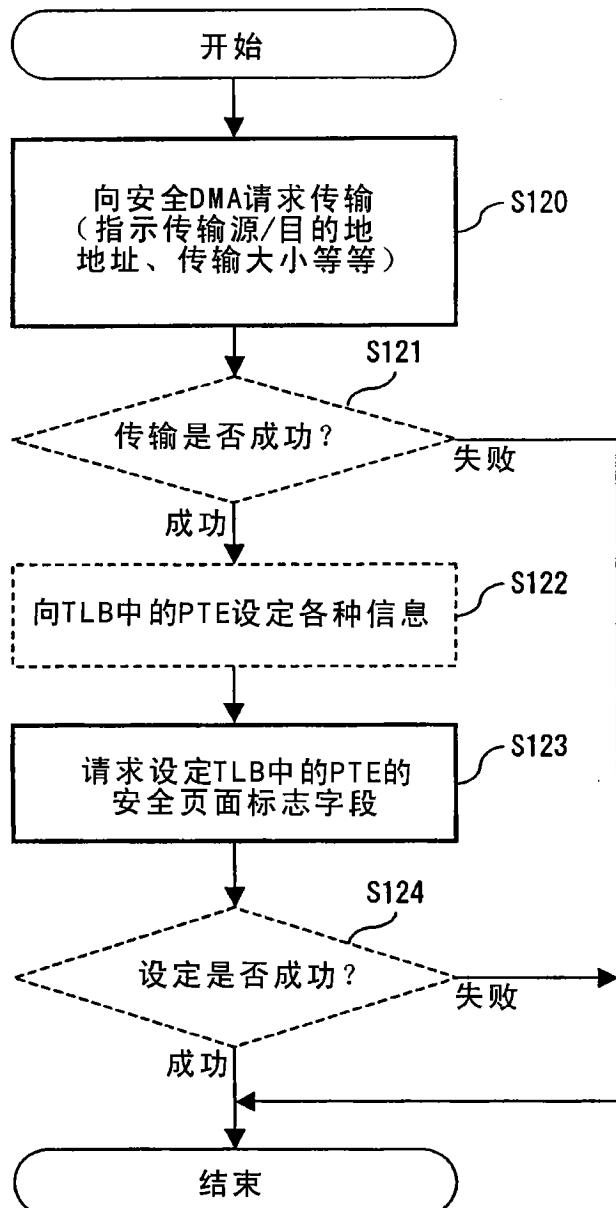
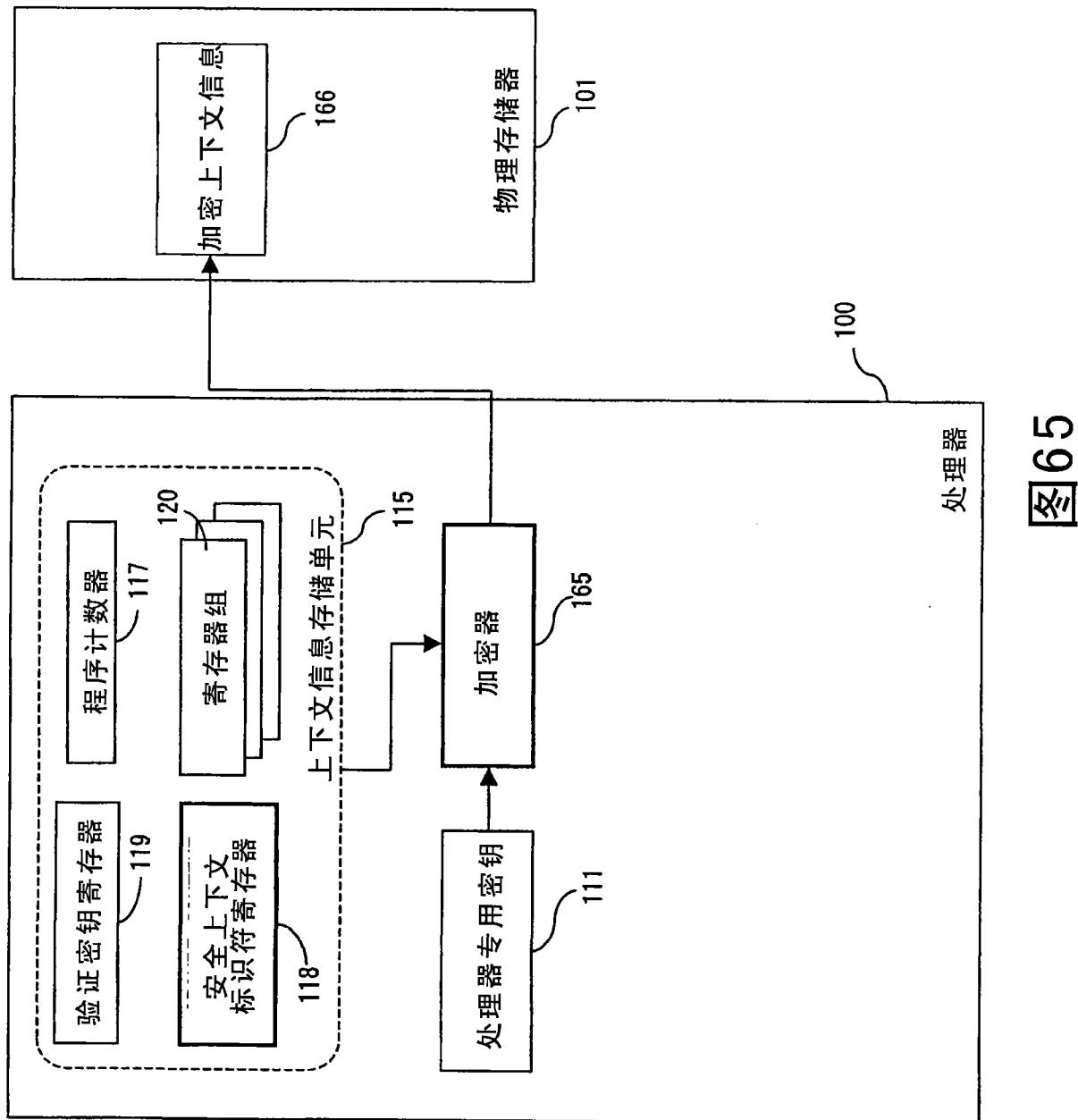
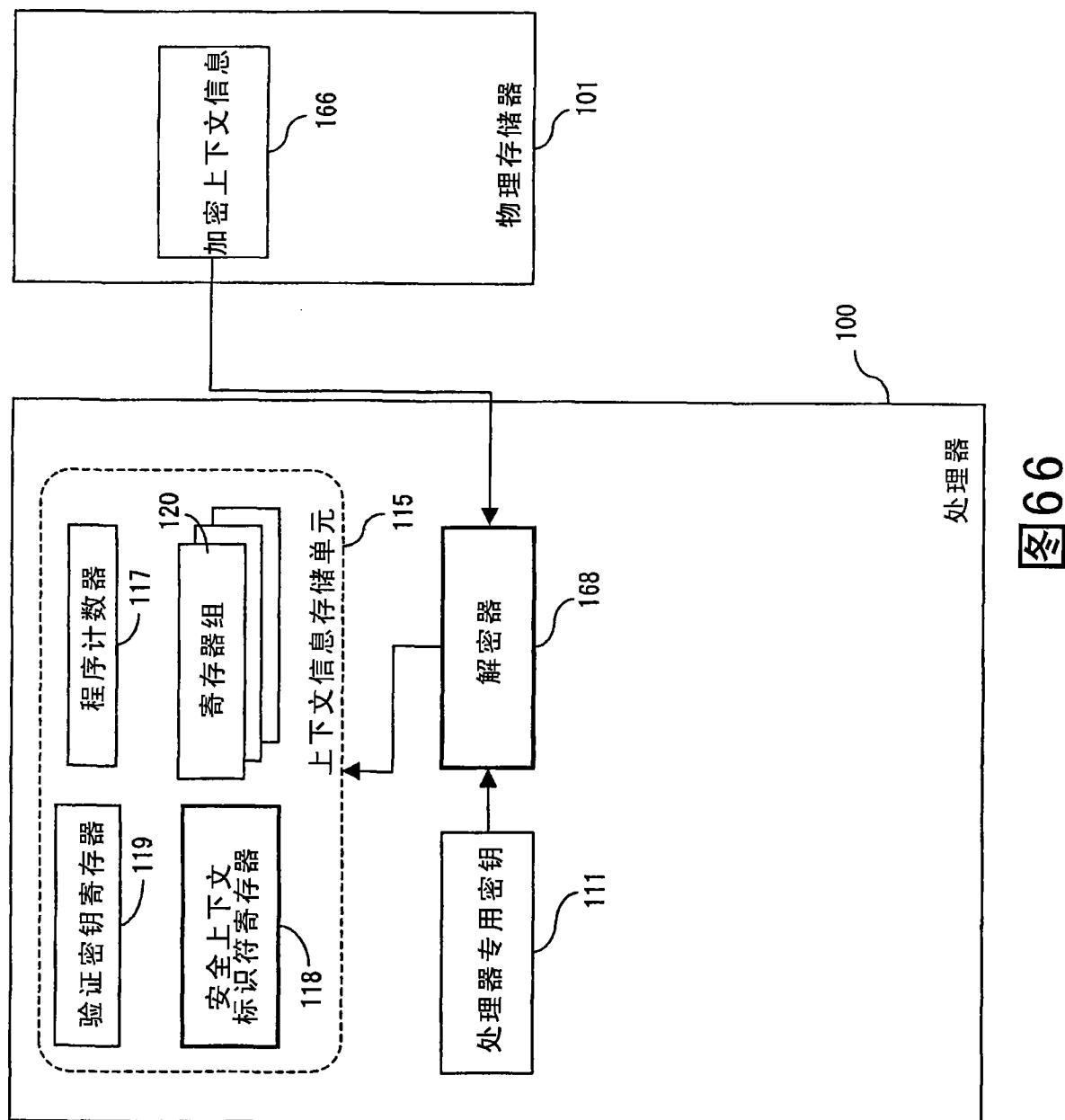


图 64





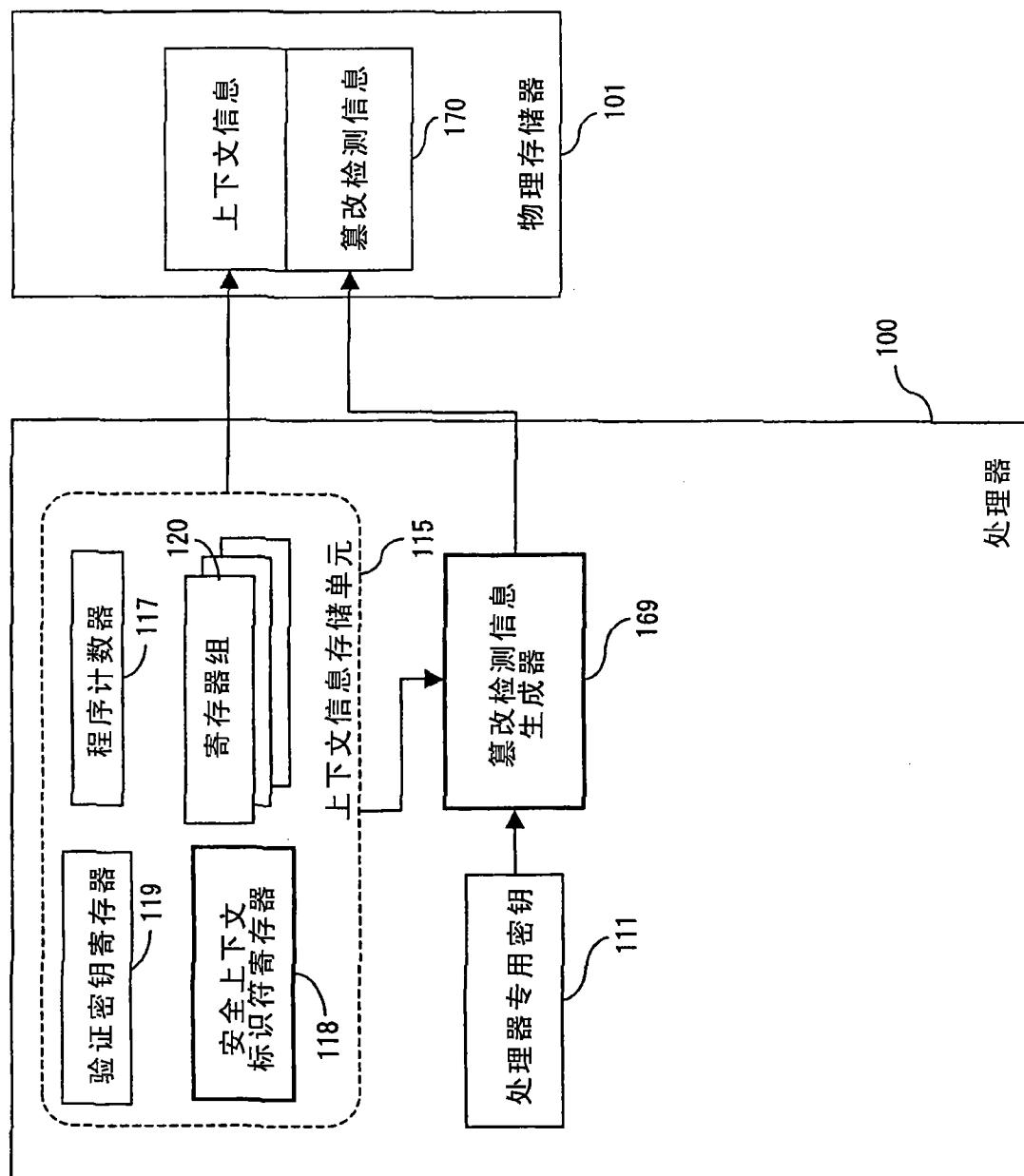


图67

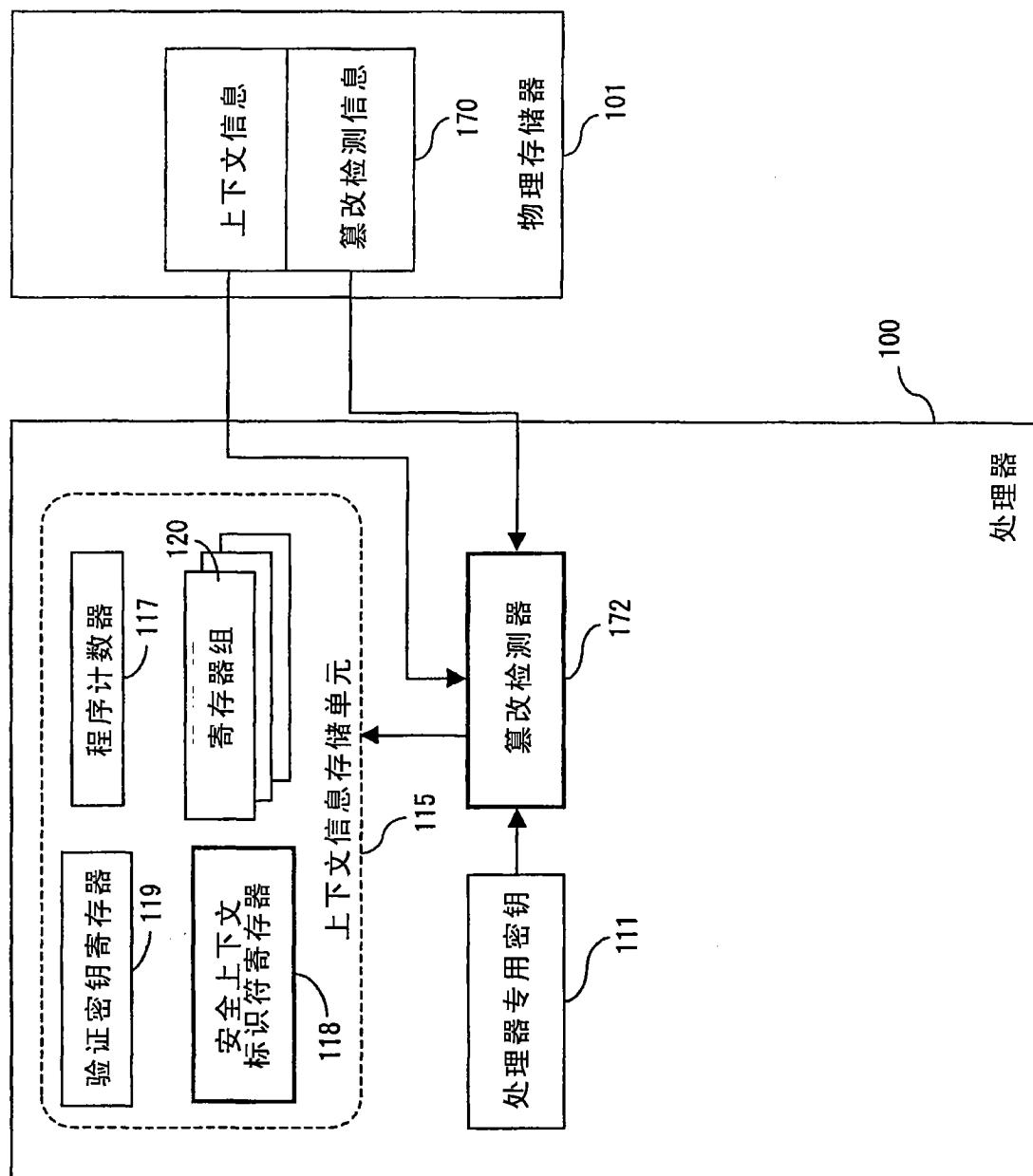


图68

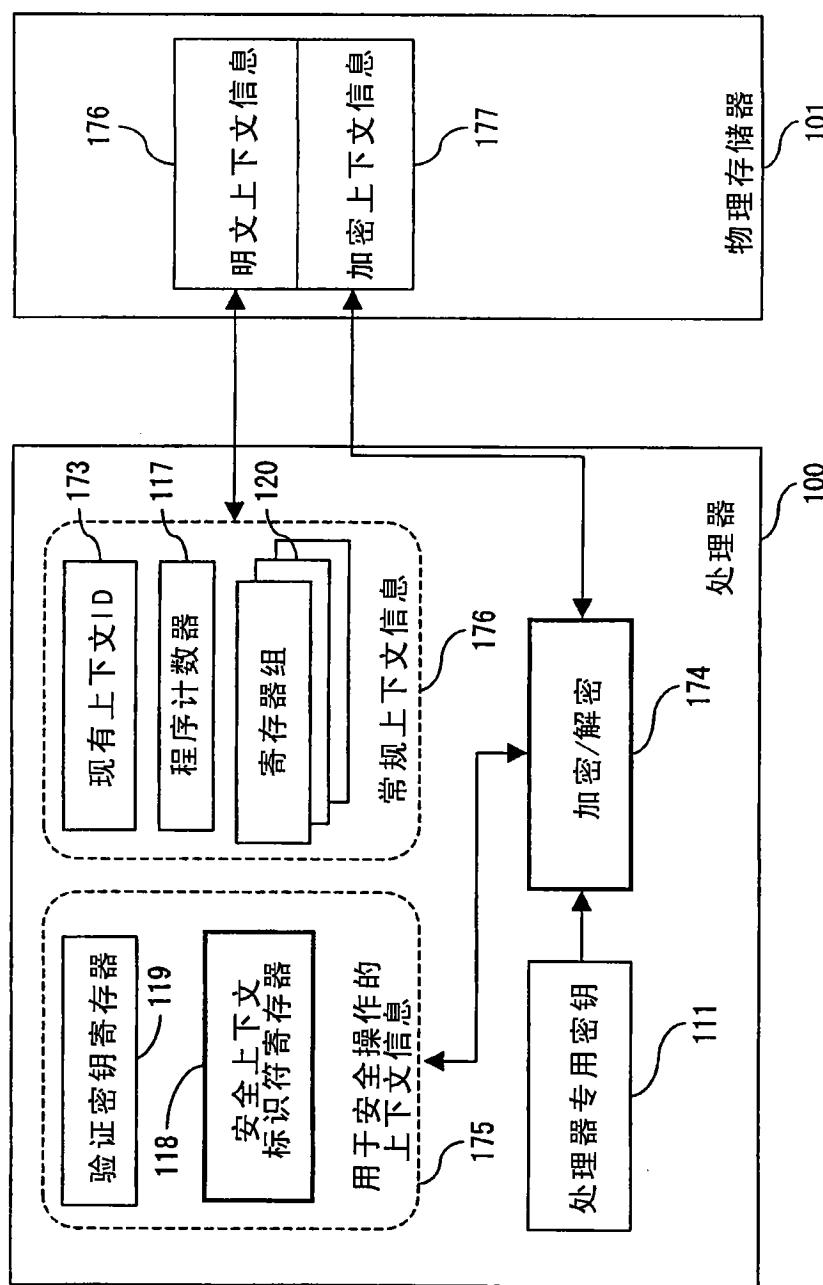


图69

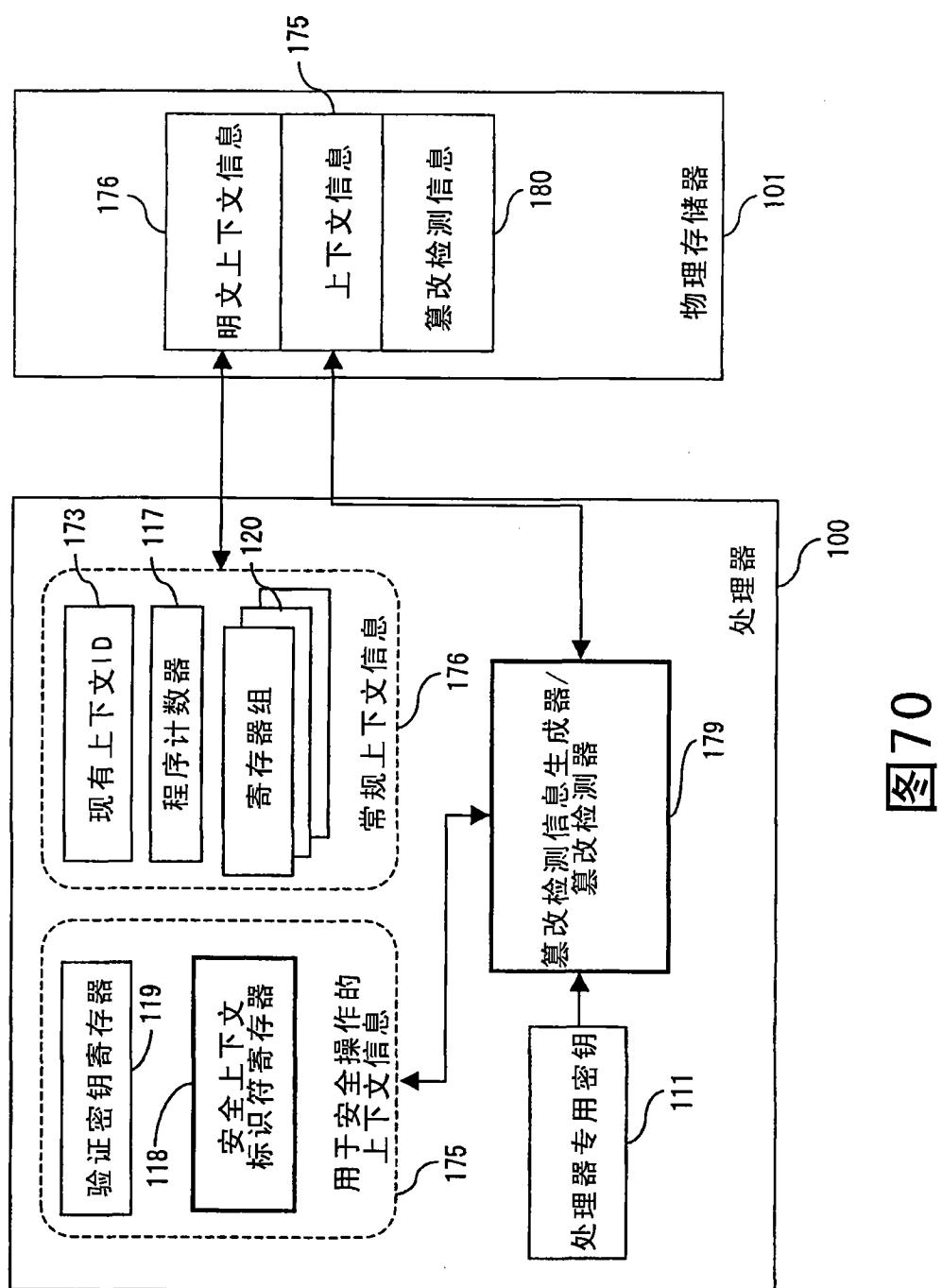


图70

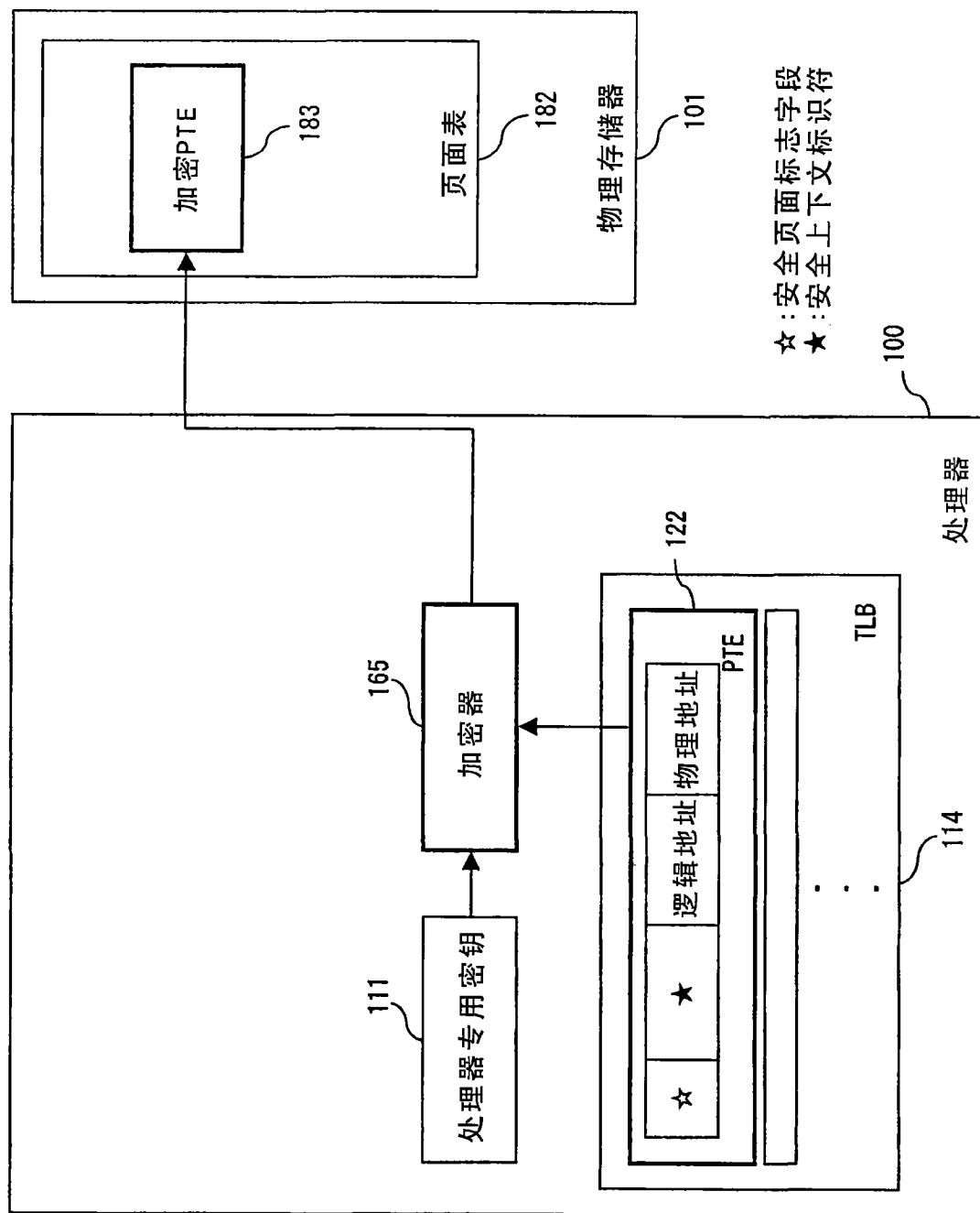


图 71

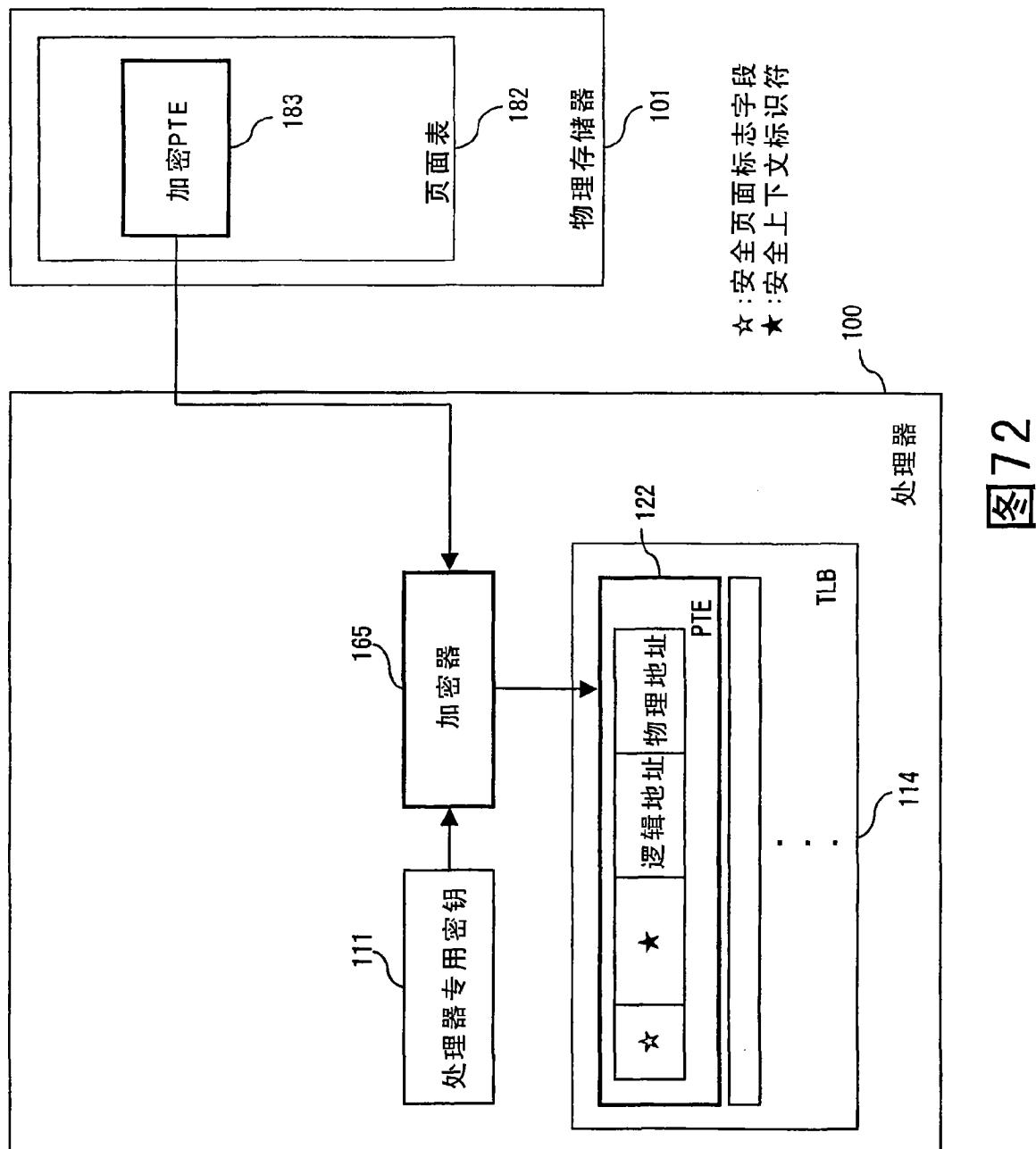
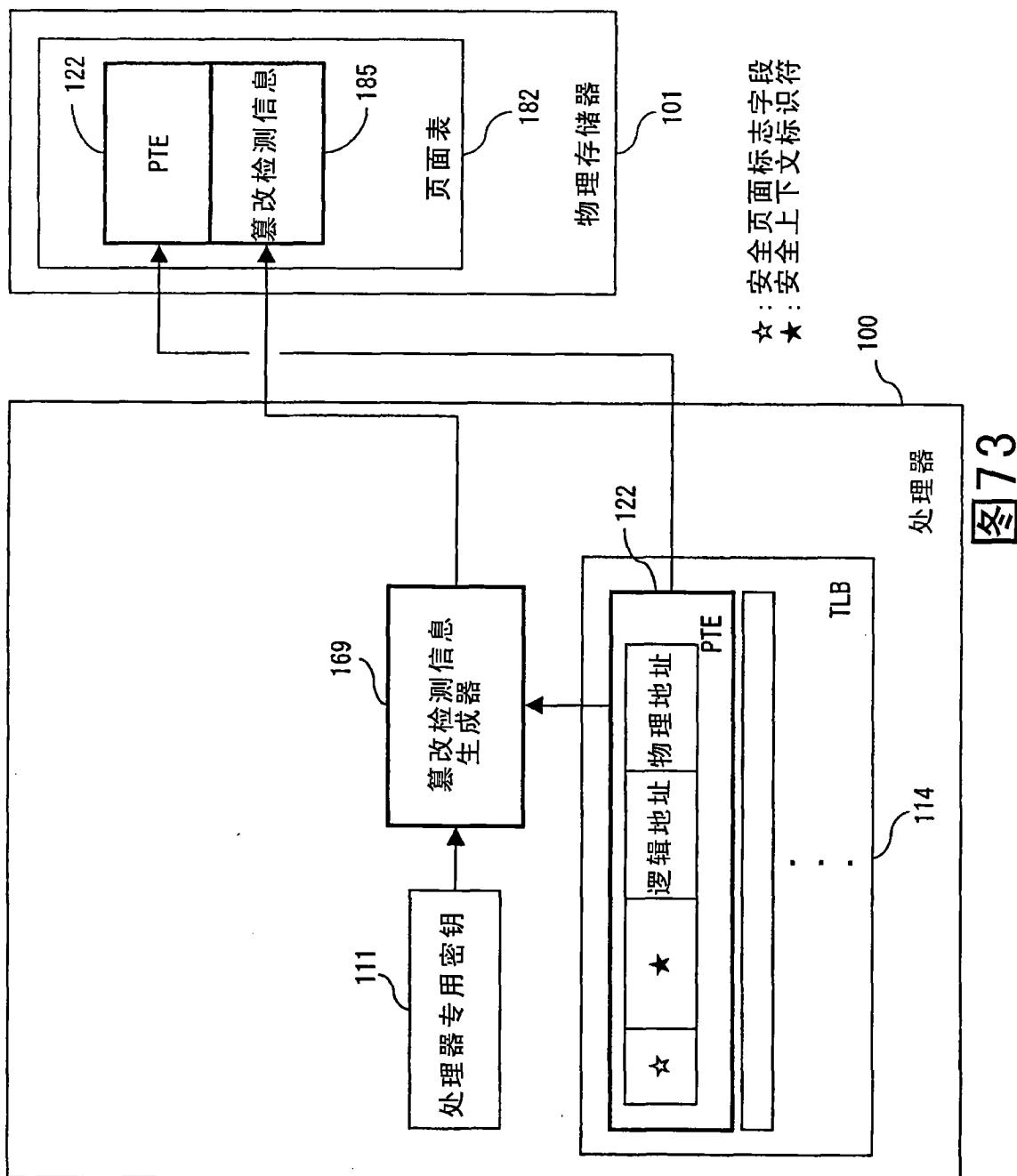


图 72



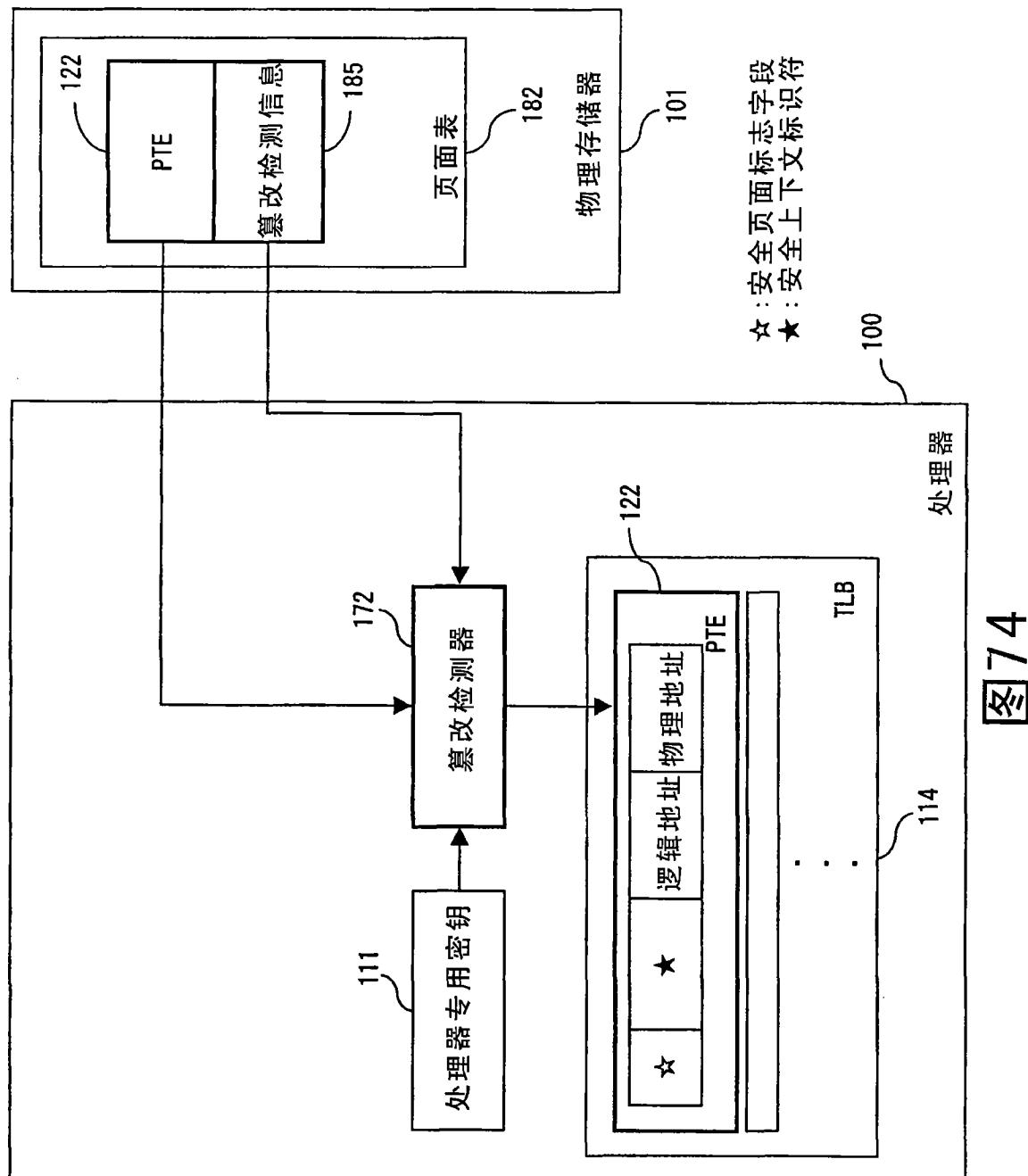


图 74

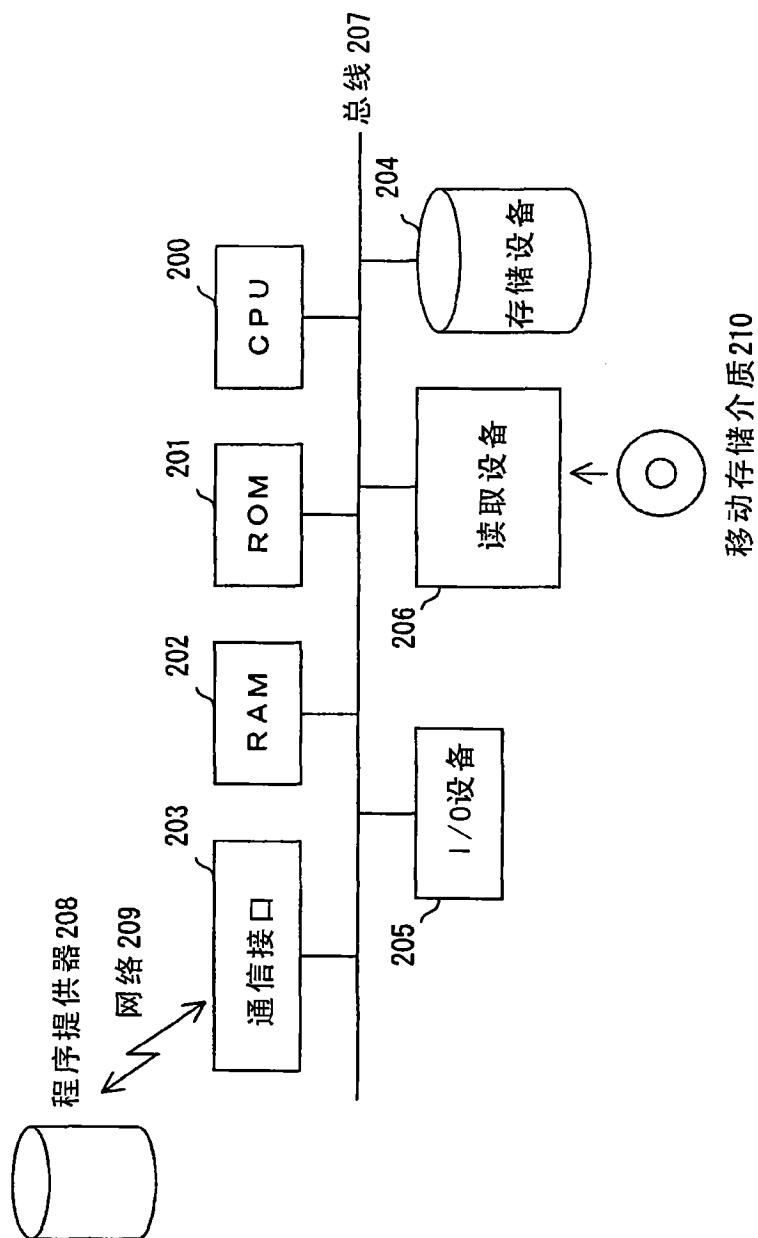


图 75