



(12) 发明专利申请

(10) 申请公布号 CN 101930794 A

(43) 申请公布日 2010. 12. 29

(21) 申请号 200910221782. 0

(22) 申请日 2001. 07. 27

(30) 优先权数据

09/626190 2000. 07. 28 US

(62) 分案原申请数据

01816420. X 2001. 07. 27

(71) 申请人 微米技术有限公司

地址 美国爱达荷州

(72) 发明人 F·F·鲁帕瓦尔

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

代理人 王小衡 李家麟

(51) Int. Cl.

G11C 11/4094(2006. 01)

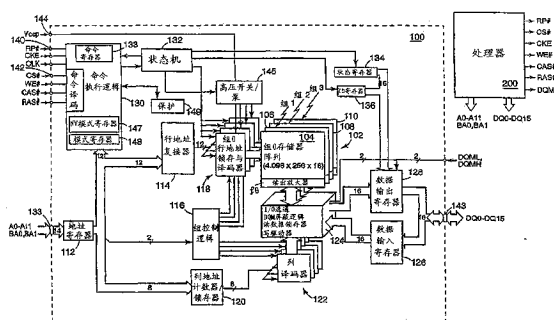
权利要求书 3 页 说明书 25 页 附图 35 页

(54) 发明名称

具有状态脉冲串输出的同步快闪存储器

(57) 摘要

公开了一种具有状态脉冲串输出的同步快闪存储器,同步快闪存储器包括非易失性存储单元的阵列。存储器阵列被排列成行和列,以及可被进一步排列成可寻址的块。数据通信连接被使用来与外部设备(诸如,处理器或其他存储器控制器)进行双向数据通信。存储器可以在一系列时钟周期期间在数据通信连接上输出来自存储寄存器的数据,以提供寄存器数据的脉冲串。存储器可以按照规定的时钟等待时间值提供寄存器数据。寄存器数据可包括状态数据、操作设置值数据、制造识别、和存储器器件识别。



1. 一种操作存储器器件的方法,包括:
为所述存储器器件设置脉冲串长度;
为所述存储器器件设置时钟等待时间周期,以及
响应于在写操作期间接收的读寄存器命令控制来自寄存器的数据输出。
2. 权利要求1的方法,其中脉冲串长度和时钟等待时间被存储在模式寄存器中。
3. 权利要求1的方法,其中控制数据输出还包括在所述存储器器件的数据通信连接上从所述寄存器输出。
4. 权利要求1的方法,其中:
操作所述存储器器件还包括使用脉冲串长度设置和时钟等待时间设置对状态读寄存器进行编程;并且其中
输出存储器器件状态还包括
将存储器置于状态寄存器读模式;以及
根据状态读寄存器编程,对于一系列时钟周期,从所述状态读寄存器输出非阵列数据;
其中所述状态读寄存器允许外部处理器在写入、擦除和保护操作期间监视内部状态机的状态,并且其中模式寄存器被用于限定所述存储器器件的特定操作,包括选择脉冲串长度、脉冲串类型、CAS 等待时间和工作模式中的至少一个。
5. 权利要求4的方法,其中输出还包括:
使用编程的脉冲串长度设置和时钟等待时间设置控制状态读寄存器数据输出。
6. 权利要求4的方法,还包括:
在状态读命令开始时提供来自所述状态读寄存器的存储器状态信号。
7. 权利要求4的方法,其中所述脉冲串长度设置是四个字并且所述时钟等待时间设置是三个周期。
8. 权利要求1的方法,其中设置脉冲串长度和设置时钟等待时间周期还包括:
在模式寄存器中存储所述脉冲串长度和所述时钟等待时间周期,并且使用存储在所述模式寄存器中的所述脉冲串长度和所述时钟等待时间对所述存储器器件进行编程。
9. 权利要求8的方法,其中控制数据输出还包括:
当在存储器操作运行时接收到状态请求时,基于接收到所述状态请求输出存储器器件状态,其中在对应于所述脉冲串长度的多个时钟周期上执行输出。
10. 权利要求8的方法,还包括在接收到所述状态请求后延迟输出所述存储器器件状态所述时钟等待时间周期。
11. 权利要求8的方法,其中输出存储器器件状态还包括在存储器操作期间从状态寄存器输出内部状态机的状态。
12. 权利要求1或8的任何一个的方法,其中对脉冲串长度进行编程包括建立x个周期的读脉冲串长度。
13. 权利要求1或8的任何一个的方法,其中对时钟等待时间进行编程包括建立y个周期的时钟等待时间。
14. 权利要求1或8的任何一个的方法,其中控制数据输出还包括在x个接连的时钟周期上从所述存储器器件输出数据。

15. 权利要求 13 的方法,还包括在接收到所述状态请求后延迟输出所述存储器器件状态所述时钟等待时间周期。

16. 权利要求 15 的方法,还包括:

在接收寄存器读命令之后延迟来自寄存器的数据输出所述时钟等待时间周期。

17. 权利要求 16 的方法,其中控制数据输出还包括使用模式寄存器来限定所述存储器器件的特定操作,包括选择脉冲串长度、脉冲串类型、时钟等待时间周期和工作模式中的至少一个。

18. 权利要求 17 的方法,还包括在写入、擦除和保护操作期间监视所述存储器器件的状态寄存器,所述状态寄存器在存储器工作期间输出内部状态机的状态。

19. 一种存储器器件,包括:

存储器单元阵列;

寄存器;

时钟信号输入连接;和

控制电路,响应于寄存器读命令在输出连接上提供来自所述寄存器的数据,所述控制电路适于执行用于输出数据的方法,包括:

为将从所述寄存器输出的数据设置脉冲串长度;

为将从所述寄存器输出的数据设置时钟等待时间周期;以及

响应于在存储器器件工作期间正被接收的读寄存器命令控制来自所述寄存器的数据输出。

20. 权利要求 19 的存储器器件,其中所述控制电路还适于设置 x 个周期的脉冲串长度。

21. 权利要求 20 的存储器器件,其中所述控制电路还适于在 x 个接连的时钟周期从所述寄存器输出数据。

22. 权利要求 20 的存储器器件,其中所述控制电路还适于在接收到读寄存器命令后延迟来自所述寄存器的输出所述时钟等待时间周期。

23. 权利要求 22 的存储器器件,其中所述控制电路还适于使用模式寄存器来限定所述存储器器件的特定操作,包括选择脉冲串长度、脉冲串类型、时钟等待时间周期和工作模式中的至少一个。

24. 权利要求 23 的存储器器件,其中所述控制电路还适于在写入、擦除和保护操作期间监视存储器器件的状态寄存器,所述状态寄存器在存储器工作期间包含内部状态机的状态。

25. 权利要求 19 的存储器器件,其中所述控制电路还适于在所述存储器器件的数据通信连接上从所述寄存器输出。

26. 一种用于操作存储器器件的方法,所述方法包括:

对所述存储器器件的阵列执行写操作;

在写操作的执行期间在第一时钟周期向所述存储器器件的输入提供寄存器数据的状态寄存器读命令;以及

在多个时钟周期期间在所述存储器器件的数据通信连接上接收状态寄存器数据,其中在提供所述状态寄存器读命令后延迟接收所述寄存器数据预定的时钟等待时间周期。

27. 权利要求 26 的方法,其中所述多个时钟周期包括四个字并且所述多个时钟周期基

本上为三个周期。

28. 权利要求 26 的方法,还包括接收所述多个时钟周期和所述预定的时钟等待时间周期。

29. 权利要求 26 的方法,其中执行写操作、提供寄存器读命令和接收寄存器数据是由耦合到所述存储器器件的处理器实现的。

30. 权利要求 26 的方法,还包括接收操作数据、状态日期、制造商标识和存储器器件标识。

31. 权利要求 26 的方法,其中写操作能够在在一组阵列上执行,同时从剩余的存储器阵列组读取数据。

32. 权利要求 31 的方法,其中所述存储器器件能够执行一组特定的读操作,同时在其它组上执行写操作。

33. 一种用于在具有处理器的系统中操作存储器器件的方法,所述方法包括:
设置所述存储器器件的脉冲串长度以限定输出寄存器数据的多个时钟周期;
设置时钟等待时间周期;
在所述存储器器件上执行写操作;
在所述写操作的执行期间在所述存储器器件上执行寄存器数据的寄存器读命令;以及在脉冲串长度期间在所述存储器器件的数据通信连接上接收寄存器数据的脉冲串,其中在提供所述寄存器读命令后延迟接收寄存器数据的脉冲串等于所述时钟等待时间周期的时间。

34. 权利要求 33 的方法,其中所述时钟等待时间周期是从 2,4,8 或 16 个时钟周期选择的。

35. 权利要求 33 的方法,其中所述寄存器数据包括操作数据,状态数据,制造商标识或存储器器件标识之一。

36. 权利要求 33 的方法,其中所述处理器通过双向数据通信总线与所述存储器器件进行通信。

37. 权利要求 33 的方法,其中所述多个时钟周期包括四个字并且所述多个时钟周期是基本上三个周期。

38. 权利要求 33 的方法,其中所述存储器器件是包括具有非易失性存储器单元的存储器阵列的闪存储器器件。

具有状态脉冲串输出的同步快闪存储器

[0001] 本申请是申请日为 2001 年 7 月 27 日、申请号为 01816420. X、发明名称为“具有状态脉冲串输出的同步快闪存储器”的申请的分案申请。

技术领域

[0002] 本发明总的涉及非易失性存储器器件,具体地,本发明涉及同步非易失性快闪存储器。

背景技术

[0003] 存储器器件典型地被提供用作为计算机中的内部存储区域。术语存储器表示具有集成电路芯片形式的数据存储。存储器具有几种不同的类型。一种类型是 RAM(随机存取存储器)。这典型地被用作为计算机环境下的主存储器。RAM 是指读出和写入存储器,也就是,可以把数据写入到 RAM 和从 RAM 读出数据。这是与 ROM 不同的,ROM 只允许读出数据。大多数 RAM 是易失性的,这意味着它需要稳定的电流来保持它的内容,只要电源被去激活,在 RAM 中无论什么数据都被丢失。

[0004] 计算机几乎总是包含小量的只读存储器(ROM),它保存用于激活计算机的指令。与 RAM 不同,ROM 不能被写入。EEPROM(电可擦可编程只读存储器)是一种特殊类型的非易失性 ROM,它可以通过把它暴露在电荷中而被擦除。像其他类型的 ROM 一样,EEPROM 传统上没有 RAM 那样快。EEPROM 包括很大数目的、具有电隔离的栅极(浮动栅)的存储单元。数据是以浮动栅上的电荷的形式被存储在存储单元的。通过编程和擦除操作,电荷分别被输送到浮动栅或从浮动栅被去除。

[0005] 另一种类型的非易失性存储器是快闪存储器。快闪存储器是一种 EEPROM,它可以以块的形式,而不是一次一个字节地,被擦除和重新编程。许多现代的 PCS 具有它们自己的、被存储在快闪存储器芯片上的 BIOS,这样,如果必要它可容易地被更新。这样的 BIOS 有时被称为快速 BIOS。快闪存储器在调制解调器中也很通用,因为它使得调制解调器制造商能够支持新的协议(当这些协议成为标准化时)。

[0006] 典型的快闪存储器包括存储器阵列,它包括很大数目的、以行和列的形式排列的存储单元。每个存储单元包括能够保持电荷的、浮动栅场效应晶体管。存储单元通常被编组为块。在一个块内的每个单元可以通过对浮动栅进行充电而被随机地电编程。通过块擦除操作,电荷可以从浮动栅上被去除。存储单元中的数据是由浮动栅上的电荷的存在与否而被确定。

[0007] 同步 DRAM(SDRAM) 是一种 DRAM,它能够以比传统的 DRAM 存储器高得多的时钟速度运行。SDRAM 使它本身与 CPU 总线同步,以及能够以 100MHz 速率运行,比传统的 FPM(快速页模式)RAM 约快三倍,以及约为快速 EDO(扩展数据输出)DRAM 和 BEDO(脉冲串扩展数据输出)DRAM 的两倍。可以迅速地访问 SDRAM,但它是易失性的。许多计算机系统被设计成使用 SDRAM 来操作,而非易失性存储器得到好处。

[0008] 由于上述的理由,以及由于下面阐述的其他理由,这些理由对于本领域技术人员在阅读和了解本技术说明后将是显而易见的,在本技术中需要一种可以类似于 SDRAM 操

作的方式操作的、非易失性存储器器件。

发明内容

[0009] 存储器器件的上述的问题和其他问题由本发明解决,以及可通过阅读和研究以下的技术说明而了解它们。

[0010] 在一个实施例中,操作同步存储器器件的方法包括确定 x 个周期的读脉冲串长度以使来自同步存储器器件的数据输出以 x 个接连的时钟周期被输出,起动寄存器读操作,以便读出被存储在内部寄存器中的数据,以及在 x 个接连的时钟周期内在外部数据连接上输出被存储在内部寄存器中的数据。

[0011] 在另一个实施例中,操作同步存储器器件的方法包括接受读寄存器命令,以及响应于读寄存器命令在 x 个时钟周期内输出来自同步存储器器件的寄存器数据。

[0012] 在再一个实施例中,操作同步存储器器件的方法包括在存储器器件的阵列上执行写操作,以及在执行写操作期间,把来自处理器的寄存器读命令提供到存储器器件。寄存器读命令在第一时钟周期在存储器器件的输入端处被接受,以及被存储在存储器寄存器中的寄存器数据被读出。寄存器数据在多个时钟周期期间在同步存储器器件的数据通信连接上被输出。寄存器数据的输出在接受到寄存器读命令后、被延时了预定的时钟等待时间间隔。

[0013] 同步存储器器件包括存储单元阵列、数据寄存器、以及时钟信号输入连接。控制电路响应于寄存器读命令在输出连接处提供来自数据寄存器的数据。控制电路根据编程的读脉冲串长度值在预定数目的时钟周期内输出数据。

附图说明

[0014] 图 1A 是本发明的同步快闪存储器的方框图;

[0015] 图 1B 是本发明的一个实施例的集成电路管脚互联图;

[0016] 图 1C 是本发明的一个实施例的集成电路互联凸点网格阵列 (bumpgrid array) 图;

[0017] 图 2、2A 和 2B 表示本发明的一个实施例的模式寄存器;

[0018] 图 3 表示具有一个、两个和三个时钟周期的 CAS 等待时间的读操作;

[0019] 图 4 表示激活在本发明的一个实施例的存储器的一个组中的一个特定的行;

[0020] 图 5 表示在工作的命令与读或写命令之间的时序;

[0021] 图 6 表示读命令;

[0022] 图 7 表示本发明的一个实施例的接连的读脉冲串的时序;

[0023] 图 8 表示在本发明的一个实施例的一页内的随机读访问;

[0024] 图 9 表示后面跟随写操作的读操作;

[0025] 图 10 表示按照本发明的一个实施例的、通过使用脉冲串终结命令终结的读脉冲串操作;

[0026] 图 11 表示写命令;

[0027] 图 12 表示后面跟随读操作的写操作;

[0028] 图 13 表示本发明的一个实施例的电源功率降低操作;

[0029] 图 14 表示在脉冲串读出期间时钟中止操作;

- [0030] 图 15 表示具有两个导引扇区的存储器的一个实施例的存储器地址变换；
- [0031] 图 16 是按照本发明的一个实施例的、自定时的写序列的流程图；
- [0032] 图 17 是按照本发明的一个实施例的、完整的写状态检验序列的流程图；
- [0033] 图 18 是按照本发明的一个实施例的、自定时的块擦除序列的流程图；
- [0034] 图 19 是按照本发明的一个实施例的、完整的块擦除状态检验序列的流程图；
- [0035] 图 20 是按照本发明的一个实施例的、块保护序列的流程图；
- [0036] 图 21 是按照本发明的一个实施例的、完整的块状态检验序列的流程图；
- [0037] 图 22 是按照本发明的一个实施例的、器件保护序列的流程图；
- [0038] 图 23 是按照本发明的一个实施例的、块的非保护序列的流程图；
- [0039] 图 24 表示起动和装载模式寄存器操作的时序；
- [0040] 图 25 表示时钟中止模式操作的时序；
- [0041] 图 26 表示脉冲串读操作的时序；
- [0042] 图 27 表示交替组读出访问的时序；
- [0043] 图 28 表示全页脉冲串读操作的时序；
- [0044] 图 29 表示通过使用数据屏蔽信号的脉冲串读操作的时序；
- [0045] 图 30 表示后面跟随对不同组读出的写操作的时序；
- [0046] 图 31 表示后面跟随对同一个组读出的写操作的时序；
- [0047] 图 32 表示本发明的存储器系统；以及
- [0048] 图 33 表示本发明的多处理器系统。

具体实施方式

[0049] 在本发明的以下的详细的说明中，参考了构成本发明的一部分的附图，以及其中显示了可以实施本发明的特定的实施例。这些实施例充分详细地被描述，以使得本领域技术人员能够实施本发明，以及将会看到，可以利用其他实施例和可以作出逻辑的、机械的和电的改变，而不背离本发明的精神和范围。所以，以下的详细说明并不是在限制的意义作出的，以及本发明的范围仅仅由权利要求限定。

[0050] 以下的详细说明分成两个主要的节。第一节是接口功能说明，它详细阐述与 SDRAM 存储器的兼容性。第二节是功能性说明，它具体说明快闪结构功能性命令。

[0051] 接口功能说明

[0052] 参照图 1A，描述本发明的一个实施例的方框图。存储器器件 100 包括非易失性快速存储单元 102 的阵列。阵列被排列成多个可寻址的组。在一个实施例中，存储器包含四个存储器组 104、106、108 和 110。每个存储器组包含存储单元的可寻址的扇区。被存储在存储器中的数据可以通过使用由地址寄存器 112 接受的、外部提供的位置地址被访问。利用行地址复用电路 114，对这些地址进行译码。这些地址也可以通过使用组控制逻辑 116 和行地址锁存与译码电路 118 被译码。为了访问存储器的适当的列，列地址计数和锁存电路 120 把接受的地址耦合到列译码电路 122。电路 124 提供输入/输出选通、数据屏蔽逻辑、读数据锁存电路和写驱动器电路。数据通过数据输入寄存器 126 被输入和通过数据输出寄存器 128 被输出。命令执行逻辑 130 被提供来控制存储器器件的基本操作。状态机 132 也被提供来控制存储在存储器阵列和单元上执行的特定的操作。状态寄存器 134 和识别寄存器 136

也可被提供来输出数据。

[0053] 图 1B 表示本发明的一个实施例的互联的管脚分配。存储器封装 150 具有 54 个互联管脚。管脚结构基本上类似于已可提供的 SDRAM 封装。对于本发明特定的两个互联管脚是 RP# 152 和 Vccp 154。虽然本发明可以共享看来与 SDRAM 的相同的互联标签,但在互联管脚上提供的信号的功能在这里被描述,以及除非这里阐述的,它不应当等同于 SDRAM 的信号的功能。图 1C 表示存储器封装 160 的一个实施例,它具有凸起连接,而不是图 1C 的管脚连接。所以,本发明并不限于特定的封装结构。

[0054] 在描述存储器器件的操作特性之前,先给出互联管脚和它们的各个信号的更详细的说明。输入时钟连接被使用来提供时钟信号 (CLK)。时钟信号可以被系统时钟驱动,以及所有的同步快闪存储器输入信号在 CLK 的上升沿处被采样。CLK 也给内部脉冲串计数器加增量,以及控制输出寄存器。

[0055] 输入时钟允许 (CKE) 连接端被使用来激活 (HIGH(高) 状态) 和去激活 (LOW(低) 状态) CLK 信号输入。去激活时钟输入可提供电源功率降低和等待操作 (此处所有的存储器组是空闲的),工作的电源功率降低 (在任一个组中存储器行是 ACTIVE(工作的)),或时钟中止操作 (脉冲串 / 访问在进行中)。CKE 是同步的,除非在器件进入功率降低模式以后,此处 CKE 成为非同步的,直至退出这个模式以后为止。输入缓冲器,包括 CLK,在功率降低模式期间是被禁止的,以便提供低的等待功率。CKE 在不需要功率降低模式 (不同于 RP# 深度功率降低) 的系统中可以维持在 HIGH(高)。

[0056] 芯片选择 (CS#) 输入连接端提供一个信号,以便允许 (寄存的 LOW(低)) 和禁止 (寄存的 HIGH(高)) 在命令执行逻辑中提供的命令译码器。当 CS# 被寄存为 HIGH 时所有的命令被屏蔽。另外,CS# 在具有多个组的系统上提供外部的组选择,以及 CS# 可被看作为命令代码的一部分;但不一定是必须的。

[0057] 用于 RAS#、CAS# 和 WE#(连同 CAS#、CS# 一起) 的输入命令输入连接端规定要被存储器执行的命令,正如下面详细地描述的。输入 / 输出屏蔽 (DQM) 连接端被使用来提供用于写访问的输入屏蔽信号,和用于读访问的输出允许信号。当 DQM 在写周期期间被采样 HIGH 时,输入数据被屏蔽。当 DQM 在读周期期间被采样 HIGH 时,输出缓冲器被置于高阻抗 (高-Z) 状态 (在两个时钟等待时间后)。DQML 相应于数据连接 DQ0-DQ7,以及 DQMH 相应于数据连接 DQ8-DQ15。DQML 和 DQMH 在表示为 DQM 时被认为是相同的状态。

[0058] 地址输入端 133 主要被使用来提供地址信号。在所表示的实施例中,存储器具有 12 条线 (A0-A11)。其他的信号可以在地址连接处被提供,正如下面描述的。地址输入端在 ACTIVE(工作的) 命令 (行地址 A0-A11) 和 READ/WRITE(读 / 写) 命令 (列地址 A0-A7) 期间被采样,以便选择各个存储器组中的一个位置。地址输入也被使用来在 LOADCOMMAND REGISTER(装载命令寄存器) 操作期间提供操作代码 (OpCode),在下面解释。地址线 A0-A11 也被使用来在 LOAD MODE REGISTER(装载模式寄存器) 操作期间输入模式设置值。

[0059] 输入复位 / 功率降低 (RP#) 连接 140 被使用于复位和功率降低操作。在初始器件功率上升时,在一个实施例中,在发出可执行的命令之前,RP# 从低转移到高以后需要 100 μ s 延时,以便内部器件初始化。RP# 信号清除状态寄存器,把内部状态机 (ISM) 132 设置在阵列读模式,以及当处在 LOW(低) 时,把器件置于深度功率降低模式。在功率降低模式期间,所有的输入连接,包括 CS# 142,都处在“不用管它”,以及所有的输出处在 High-Z (高

阻) 状态。当 RP# 信号等于 VHH 电压 (5 伏) 时, 在 WRITE 和 ERASE 期间, 所有的保护模式被忽略。RP# 信号也允许器件保护比特被设置为 1 (保护) 以及允许 16 比特寄存器的块保护比特 (在位置 0 和 15) 被设置为 0 (非保护), 当被加上 VHH 时。保护比特将在下面更详细地被描述。在所有其他操作模式期间, RP# 被保持为 HIGH。

[0060] 组地址输入连接, BA0 和 BA1 规定哪个组加上 ACTIVE、READ、WRITE、或 BLOCK PROTECT 命令。DQ0-DQ15 连接 143 是数据总线连接, 被使用于双向数据通信。参照图 1B, VCCQ 连接被使用来提供隔离的功率到 DQ 连接, 以便改善对噪声的抗扰性。在一个实施例中, $VCCQ = V_{cc}$ 或 $1.8V \pm 0.15V$ 。VSSQ 连接被使用来提供隔离的地到 DQ 连接, 以便改善噪声抗扰性。VCC 连接提供供电电源, 诸如 3V。通过 Vss 连接来提供地连接。另一个工作电压被提供到 VCCP 连接 144。VCCP 连接在器件初始化、WRITE 和 ERASE 操作期间可以外部地连接到 VCC 以及源电流。也就是, 写入或擦除存储器器件可以通过使用 VCCP 电压被执行, 而所有其他的操作可以用 VCC 电压来执行。VCCP 连接被耦合到高电压开关 / 泵电路 145。

[0061] 以下的节提供同步快闪存储器的操作的更详细的说明。本发明的一个实施例是非易失性、扇区电可擦的 (快闪)、可编程只读存储器, 包含被组织为 4, 194, 304 字 $\times$ 16 比特的 67, 108, 864 比特。其他流行的密度也是预期的, 但本发明并不限于示例的密度。每个存储器组被组织为四个独立的可擦除的块 (总共 16 个)。为了确保关键的固件被保护免受偶然的擦除或过写入, 存储器可包括 16 个 256K 字硬件和软件可锁定的块。存储器的四个组的结构支持真正的同时发生的操作。

[0062] 到任何组的读访问可以与到任何其他组的背景写入或擦除操作同时发生。同步快闪存储器具有同步的接口 (所有的信号在时钟信号 CLK 的上升沿处被登记)。到存储器的读访问可以是面向脉冲串的。也就是, 存储器访问在选择的位置处开始以及在编程的序列中编程的数目的位置上继续。读访问从后面跟随 READ 命令的 ACTIVE 命令的登记开始。与 ACTIVE 命令一致的登记的地址比特被使用来选择要被访问的组和行。与 READ 命令一致的登记的地址比特被使用来选择脉冲串访问的开始列的位置和组。

[0063] 同步快闪存储器提供可编程的读脉冲串长度, 1、2、4、或 8 个位置或全页, 带有脉冲串终结任选项。另外, 同步快闪存储器使用内部管线结构达到高速度操作。

[0064] 同步快闪存储器可以操作在低功率存储器系统中, 诸如工作在 3 伏的系统。深度功率降低模式可以连同功率节省等待模式一起被提供。所有的输入和输出是低压晶体管 - 晶体管逻辑 (LVTTTL) 可兼容的。同步快闪存储器提供快闪工作性能方面重大的优点, 包括与自动列地址生成同步地闪现高数据速率的数据的能力以及在脉冲串访问期间在每个时钟周期随机地改变列地址的能力。

[0065] 通常, 同步快闪存储器类似于操作在低压的多组 DRAM 被配置。以及包括同步接口。每个组被组织为行和列。在正常操作之前, 同步快闪存储器被初始化。以下的节提供详细的信息, 包括器件初始化, 寄存器规定, 命令说明和器件操作。

[0066] 同步快闪存储器被加上电源, 以及以预定的方式被初始化。在功率加到 VCC, VCCQ 和 VCCP (同时地) 以及时钟信号稳定后, RP# 140 从 LOW 状态转移到 HIGH 状态。在 RP# 转移到 HIGH 后需要一个延时, 诸如 100 μs 延时, 以便完成内部器件初始化。在延迟的时间过去以后, 存储器被置于阵列读模式, 以及准备好接受模式寄存器编程或可执行的命令。在非易失性模式寄存器 147 (NV 模式寄存器) 的初始编程后, 内容在初始化期间被自动装载到易

失性模式寄存器 148。器件在编程状态下被加上功率,以及不需要在发出工作命令之前重新装载非易失性寄存器 147。这将在下面更详细地说明。

[0067] 模式寄存器 148 被使用来规定同步快闪存储器的特定的工作模式。这个规定包括选择脉冲串长度、脉冲串类型、CAS 等待时间、和工作模式,如图 2 所示。模式寄存器通过 LOAD MODE REGISTER(装载模式寄存器)命令被编程以及保持存储的信息,直至它被重新编程为止。模式寄存器的内容可被复制到 NV 模式寄存器 147。NV 模式寄存器设置值在初始化期间自动装载模式寄存器 148。关于 ERASE NVMODE REGISTER(擦除 NV 模式寄存器)和 WRITE NVMODE REGISTER(写入 NV 模式寄存器)命令序列的细节在下面被提供。本领域技术人员将会看到,SDRAM 要求模式寄存器在每次初始化操作期间必须被外部地装载。本发明允许缺省模式被存储在 NV 模式寄存器 147。NV 模式寄存器的内容然后被复制到易失性模式寄存器 148,以便在存储器操作期间访问。

[0068] 模式寄存器比特 M0-M2 规定脉冲串长度,M3 规定脉冲串类型(顺序的或交错的),M4-M6 规定 CAS 等待时间,M7 和 M8 规定工作模式,M9 被设置为 1,以及 M10 和 M11 在本实施例中保留。因为当前不执行写脉冲串,故 M9 被设置为逻辑 1,以及写访问是单个位置(非脉冲串)访问。当所有的组是空闲时,模式寄存器必须被装载,以及在初始化以后的操作之前,控制器必须等待规定的时间。

[0069] 对同步快闪存储器的读访问可以是面向脉冲串的,脉冲串长度是可编程的,如表 1 所示。脉冲串长度确定对于给定的 READ 命令可被自动访问的列位置的最大数目。对于顺序的和交错的脉冲串类型都可得到 1、2、4、或 8 个位置的脉冲串长度,以及对于顺序的类型可得到全页的脉冲串长度。全页的脉冲串可以结合 BURST TERMINATE(脉冲串终结)命令被使用来生成任意的脉冲串长度,也就是,脉冲串可被选择地终结,以便提供定做的长度的脉冲串。当发出 READ 命令时,等于脉冲串长度的列的块实际上被选择。对于该脉冲串的所有的访问发生在这个块内,意味着如果到达边界,脉冲串将隐蔽在该块内。当脉冲串长度被设置为 2 时这个块被 A1-A7 唯一地选择,当脉冲串长度被设置为 4 时这个块被 A2-A7 选择,以及当脉冲串长度被设置为 8 时这个块被 A3-A7 选择。其余的(最低有效的)地址比特被使用来选择该块内的开始位置。如果到达边界,则全页的脉冲串隐蔽在该页内。

[0070] 在给定的脉冲串内的访问可被编程为顺序的或交错的;这被称为脉冲串类型,以及通过比特 M3 被选择。在一个脉冲串内的访问的次序由脉冲串长度、脉冲串类型和开始的列地址确定,如表 1 所示。

[0071] 表 1 脉冲串的定义

[0072]

		脉冲串内访问的次序	
脉冲串长度	开始的列地址	类型=顺序的	类型=交错的
2	A0	0-1	0-1
	0	1-0	1-0
	1		
4	A1 A0		
	0 0	0-1-2-3	0-1-2-3
	0 1	1-2-3-0	1-0-3-2
	1 0	2-3-0-1	2-3-0-1
	1 1	3-0-1-2	3-2-1-0
8	A2 A1 A0		
	0 0 0	0-1-2-3-4-5-6-7	0-1-2-3-4-5-6-7
	0 0 1	1-2-3-4-5-6-7-0	1-0-3-2-5-4-7-6
	0 1 0	2-3-4-5-6-7-0-1	2-3-0-1-6-7-4-5
	0 1 1	3-4-5-6-7-0-1-2	3-2-1-0-7-6-5-4
	1 0 0	4-5-6-7-0-1-2-3	4-5-6-7-0-1-2-3
	1 0 1	5-6-7-0-1-0-3-2	5-4-7-6-1-0-3-2
	1 1 0	6-7-0-1-2-3-4-5	6-7-4-5-2-3-0-1
	1 1 1	7-0-1-2-3-4-5-6	7-6-5-4-3-2-1-0
全页	n=A0-A7	Cn, Cn+1, Cn+2, Cn+3	不支持
256	(位置 0 - 255)	Cn+4, ...Cn-1, Cn...	

[0073] 列地址选通 (CAS) 等待时间是在 READ 命令的登记与 DQ 连接端处的第一片输出数据的可提供性之间的延时 (以时钟周期为单位)。等待时间可被设置为一, 二, 或三个时钟周期。例如, 如果 READ 命令在时钟边沿 n 处被登记, 以及等待时间是 m 个时钟, 则数据将是由时钟边沿 n+m 可提供的。由于提早一个周期的时钟边沿 (n+m-1) 的结果, DQ 连接将开始驱动数据, 以及如果相关的访问时间是满足的话, 数据在时钟边沿 n+m 之前是正确的。例如, 假设时钟周期时间是使得所有的相关的访问时间是满足的, 如果 READ 命令在 T0 时被登记, 以及等待时间被编程为两个时钟, DQ 将在 T1 后开始驱动, 以及数据在 T2 之前是正确的, 如图 3 所示。图 3 表示示例的工作频率, 不同的时钟等待时间设置值可以以这个频率被使用。正常的操作模式通过把 M7 和 M8 设置为零而被选择, 以及编程的脉冲串长度应用到 READ 脉冲串。

[0074] 以下的真值表提供在本发明的存储器的实施例的操作命令方面的更多的细节。这里提供命令的说明, 以及接着是真值表 2。

[0075] 真值表 1 接口命令和 DQM 操作

[0076]

名称 (功能)	CS#	RAS#	CAS#	WE#	DQM	ADDR	DQs
COMMAND INHIBIT (NOP)	H	X	X	X	X	X	X
NO OPERATION (NOP)	L	H	H	H	X	X	X
ACTIVE (选择组和工作的行)	L	L	H	H	X	组 / 行	X
READ (选择组、列和开始 READ 脉冲串)	L	H	L	H	X	组 / 列	X
WRITE (选择组、列和开始 WRITE)	L	H	L	L	X	组 / 列	正确的
BURST TERMINATE	L	H	H	L	X	X	工作的
ACTIVE TERMINATE	L	L	H	L	X	X	X
LOAD COMMAND	L	L	L	H	X	Com	X
REGISTER						代码	
LOAD MODE REGISTER	L	L	L	L	X	操作码	X
写允许 / 输出允许	—	—	—	—	L	—	工作的
写禁止 / 输出 High-Z	—	—	—	—	H	—	高阻

[0077] 真值表 2 快闪存储器命令序列

[0078]

操作	第一周期					第二周期					第三周期				
	CMD	ADDR	ADDR	DQ	RP#	CMD	ADDR	ADDR	DQ	RP#	CMD	ADDR	ADDR	DQ	RP#
读器件结构	LCR	90H	组	X	H	工作的	行	组	X	H	读	CA	组	X	H
读状态寄存器	LCR	70H	X	X	H	工作的	X	X	X	H	读	X	X	X	H
清除状态寄存器	LCR	50H	X	X	H										
擦除建立/确认	LCR	20H	组	X	H	工作的	行	组	X	H	写	X	组	D0H	H/VHH
写建立/写	LCR	40H	组	X	H	工作的	行	组	X	H	写	列	组	D1H	H/VHH
保护块/确认	LCR	60H	组	X	H	工作的	行	组	X	H	写	X	组	01H	H/VHH
保护器件/确认	LCR	60H	组	X	H	工作的	X	组	X	H	写	X	组	F1H	VHH
非保护块/确认	LCR	60H	组	X	H	工作的	X	组	X	H	写	X	组	D0H	H/VHH
擦除 NV 模式寄存器	LCR	30H	组	X	H	工作的	X	组	X	H	写	X	组	C0H	H
写 NV 模式寄存器	LCR	A0H	组	X	H	工作的	X	组	X	H	写	X	组	X	H

[0079] COMMAND INHIBIT(命令禁止)功能阻止同步快闪存储器执行新的命令,不管 CLK 信号是否允许。同步快闪存储器实际上被去选择,但已在进行的操作不受影响。

[0080] NO OPERATION(NOP)(不操作)命令被使用来对于被选择的同步快闪存储器(CS#

为 LOW(低)) 执行 NOP。这阻止在空闲或等待状态期间登记不想要的命令,但已在进行的操作不受影响。

[0081] 模式寄存器数据通过输入端 A0-A11 被装载。在所有的阵列组是空闲时只能发出 LOAD MODE REGISTER(装载模式寄存器)命令,以及在预定的延时(MRD)被满足之前不能发出随后的可执行的命令。在 NV 模式寄存器 147 中的数据在上电初始化时被自动装载到模式寄存器 148,它是缺省数据,除非它与 LOAD MODE REGISTER 命令一起被动态地改变。

[0082] ACTIVE(工作的)命令被使用来打开(或激活)特定的阵列组中的一行,用于以后的访问。在 BA0,BA1 输入端上的数值选择那个组,以及在输入端 A0-A11 上提供的地址选择那一行。这一行保持对于访问是工作的,直至下一个 ACTIVE 命令、功率降低、或复位为止。

[0083] READ(读)命令被使用来起动对工作的行的脉冲串读访问。在 BA0,BA1 输入端上的数值选择那个组,以及在输入端 A0-A7 上提供的地址选择开始的列位置。读出的数据出现在 DQ 上,该 DQ 受到在两个时钟之前存在的、在数据屏蔽(DQM)输入端上逻辑电平的支配。如果给定的 DQM 信号被登记为 HIGH,则相应的 DQ 将在两个时钟后是 High-Z(高阻);如果 DQM 信号被登记为 LOW,则 DQ 将提供正确的数据。因此,DQM 输入端可被使用来在读操作期间屏蔽输出数据。

[0084] WRITE(写)命令被使用来起动对工作的行的单个位置的写访问。WRITE 命令前面必须有 WRITE SETUP(写建立)命令。在 BA0,BA1 输入端上的数值选择那个组,以及在输入端 A0-A7 上提供的地址选择列位置。在 DQ 上出现的输入数据被写入到存储器阵列,该 DQ 受到与数据一致的 DQM 输入逻辑电平的支配。如果给定的 DQM 信号被登记为 LOW,则相应的数据将被写入到存储器;如果 DQM 信号被登记为 HIGH,则相应的数据输入将被忽略,以及对该字/列位置不执行 WRITE。具有 DQMHIGH 的 WRITE 命令被认为是 NOP。

[0085] ACTIVE TERMINATE(工作终结)命令对于同步快闪存储器是不需要的,但它可被提供以类似于 SDRAM PRECHARGE 命令的方式来终结读操作。ACTIVE TERMINATE 命令可被发出来终结正在进行的 BURST READ(脉冲串读出),以及可以是或不一定是组特定的。

[0086] BURST TERMINATE(脉冲串终结)命令被使用来截断固定长度的脉冲串或全页的脉冲串。在 BURST TERMINATE 命令之前的最近的 READ 命令将被截断。BURST TERMINATE 不是组特定的。

[0087] 装载命令寄存器操作被使用来起动对命令执行逻辑(CEL)130 的快闪存储器控制命令。CEL 接受和解释加到器件的命令。这些命令控制内部状态机 132 和读路径(即,存储器阵列 102、ID 寄存器 136 或状态寄存器 134)的操作。

[0088] 在任何 READ 或 WRITE 命令可被发出到同步快闪存储器内的一个组之前,在该组中的一行必须被“打开”。这是通过 ACTIVE 命令(由 CS#, WE#, RAS#, CAS# 规定的)完成的,该命令选择要被激活的组和行,见图 4。

[0089] 在打开一行(发出 ACTIVE 命令)后,READ 或 WRITE 命令可被发出到由时间间隔(t_{RCD})技术说明规定的行, $t_{\text{RCD}}(\text{MIN})$ 应当由时钟周期划分,以及舍入到下一个整个数目,确定在 ACTIVE 命令后的、READ 或 WRITE 命令可被借以进入的、最早的时钟边沿。例如,对于 90MHz 时钟(11.11ns 周期)的 30ns 的 t_{RCD} 技术说明导致 2.7 个时钟,它被舍入为 3。这反映在图 5 上,它覆盖其中 $2 < t_{\text{RCD}}(\text{MIN})/t_{\text{CK}} < 3$ 的任意情形。(相同的程序过程被使用从时间单位到时钟周期来转换其他技术说明极限值)。

[0090] 以后的、对同一个组的不同的行的 ACTIVE 命令可被发出,而不必关闭先前工作的行,只要在对同一个组的接连的 ACTIVE 命令之间的最小时间间隔是由 tRC 规定的话。

[0091] 在第一组被访问的同时,以后的、对另一个组的 ACTIVE 命令可被发出,这导致总计行访问开销的减小。在对不同组的接连的 ACTIVE 命令之间的最小时间间隔是由时间间隔 tRRD 规定。

[0092] READ 脉冲串从 READ 命令(由 CS#, WE#, RAS#, CAS# 规定)起动,如图 6 所示。开始的列和组地址从 READ 命令提供。在 READ 脉冲串期间,来自开始列地址的正确的数据读出单元在 READ 命令后的 CAS 等待时间后是可以得到的。每个随后的数据读出单元在下一个时钟上升沿之前是正确的。在完成脉冲串后,假设没有其他命令被起动,DQ 将进到 High-Z 状态。全页的脉冲串将继续,直至终结为止。(在该页的结尾,它将隐蔽到列 0,并继续进行。)来自任何 READ 脉冲串的数据可以用以后的 READ 命令截断,以及来自固定长度 READ 脉冲串的数据可以紧接地被来自以后的 READ 命令的数据所跟随。在任一种情形下,连续的数据流可被保持。来自新的脉冲串的第一数据单元跟随在完成的脉冲串的最后的单元后面,或跟随在被截断的较长的脉冲串的最后的想要的数据单元后面。新的 READ 命令应当在最后的想要的数据单元是正确的时钟边沿之前的 x 周期被发出,其中 x 等于 CAS 等待时间减 1。这示于图 7 中对于 1,2,和 3 的 CAS 等待时间;数据单元 n+3 或者是 4 的脉冲串的最后的,或者是较长的脉冲串的最后想要的。同步快闪存储器使用流水线结构,所以,不需要与预取的结构有关的 2n 法则。READ 命令可以在先前的 READ 命令后的任何时钟周期被起动。可以执行在一页内的全速的随机读访问,如图 8 所示,或可以执行对不同的组的每个随后的 READ。

[0093] 来自任何 READ 脉冲串的数据可以用以后的 WRITE 命令被截断(WRITE 命令之前必须有 WRITE SETUP),以及来自固定长度 READ 脉冲串的数据可以紧接地被来自随后的 WRITE 命令(受到总线换向限制)的数据所跟随。WRITE 可以在紧接在来自 READ 脉冲串的最后的(或最后想要的)数据单元后面的时钟边沿处被起动,只要 I/O 竞争可被避免。在给定的系统设计中,可以有可能性:驱动输入数据的器件在同步快闪存储器 DQ 进到 High-Z 之前进到 Low-Z(低阻)。在这种情形下,至少单个周期延时将出现在最后的读数据与 WRITE 命令之间。

[0094] DQM 输入被使用来避免 I/O 竞争,如图 9 所示。DQM 信号必须在 WRITE 命令之前至少坚持(HIGH)两个时钟(对于输出缓冲器,DQM 等待时间是两个时钟)坚持(HIGH),以便抑制来自 READ 的数据输出。一旦 WRITE 命令被登记,DQ 将进到 High-Z(或保持在 High-Z),而不管 DQM 信号的状态。DQM 信号必须在 WRITE 命令之前被去除坚持,(对于输入缓冲器,DQM 等待时间是零),以便确保写入的数据不被屏蔽。图 9 表示其中时钟频率允许不用加上 NOP 周期而避免总线竞争的情形。

[0095] 固定长度或全页 READ 脉冲串可以用 ACTIVE TERMINATE 命令(可以是或不一定是组特定的)或 BURST TERMINATE(不是组特定的)命令截断。ACTIVE TERMINATE 或 BURST TERMINATE 命令应当在最后的想要的数据单元是正确的时钟边沿之前的 x 周期被发出,其中 x 等于 CAS 等待时间减 1。这是在图 10 上对于每个可能的 CAS 等待时间所表示的;数据单元 n+3 或者是四的脉冲串的最后的想要的数据单元,或者是较长的脉冲串的最后想要的

[0096] 单个位置 WRITE 从 WRITE 命令（由 CS#, WE#, RAS#, CAS# 规定）起动,如图 11 所示。开始的列和组地址从 WRITE 命令提供。一旦 WRITE 命令被登记,就可如真值表 4 和 5 规定的那样执行 READ 命令。图 12 上表示一个例子。在 WRITE 期间,正确的数据输入与 WRITE 命令一致地被登记。

[0097] 不像 SDRAM 那样,同步快闪存储器不需要 PRECHARGE 命令来去激活在特定的组中打开的行,或在所有的组中打开的行。ACTIVETERMINATE 命令类似于 BURST TERMINATE 命令;然而,ACTIVE TERMINATE 可以是或不一定是组特定的。在 ACTIVE TERMINATE 命令期间坚持输入 A10 是 HIGH,将终结在任何组中的 BURST READ。当 A10 在 ACTIVETERMINATE 命令期间是低时,BA0 和 BA1 将确定哪个组将接受终结操作。ACTIVE TERMINATE 对于不是由 A10、BA0、BA1 寻址的组被认为是 NOP。

[0098] 当不是正在进行访问时,如果时钟允许,CKE 与 NOP 或 COMMANDINHIBIT(当没有进行访问时)一致地被登记为 LOW,则出现功率降低。在内部状态机操作(包括 WRITE 操作)完成后,进入功率降低使得去激活输入和输出缓冲器(不包括 CKE),以便在待机状态时节省功率。

[0099] 功率降低状态可通过在想要的时钟边沿处(满足 tCKS)登记 NOP 或 COMMANDINHIBIT 和 CKE HIGH 而退出。对于示例性功率降低操作,参阅图 13。

[0100] 当列访问/脉冲串正在进行和 CKE 被登记为 LOW 时出现时钟中止模式。在时钟中止模式下,内部时钟被去激活,“冻结”同步逻辑。对于 CKE 借以被采样为 LOW 的每个时钟上升沿,下一个内部的时钟上升沿被中止。在被中止的内部时钟边沿的时间出现在输入管脚处的任何命令或数据被忽略,出现在 DQ 管脚处的任何数据将保持被驱动,以及脉冲串计数器不加增量,只要时钟被中止(见图 14 的例子)。时钟中止模式可通过登记 CKE HIGH 而退出;内部时钟和相关的操作将在以后的时钟上升沿时继续进行。

[0101] 在一个实施例中,脉冲串读/单个写模式是缺省模式。所有的 WRITE 命令导致单个列位置(长度为一的脉冲串)的访问,而 READ 命令按照编程的脉冲串长度和序列对列进行访问。以下的真值表 3 表示使用 CKE 信号的存储器操作。

[0102] 真值表 3-CKE

[0103]

CKE _{n-1}	CKE _n	当前状态	COMMAND _n (命令)	ACTION _n (动作)
L	L	功率降低 时钟中止	X X	保持功率降低 保持时钟中止
L	H	功率降低 时钟中止	命令禁止或 NOP X	退出功率降低 退出时钟中止
H	L	所有的组空闲 读或写	命令禁止或 NOP 正确	进入功率降低 进入时钟中止
H	H		见真值表 4	

[0104] 真值表 4- 当前状态组 n- 到组 n 的命令

[0105]

当前状态	CS#	RAS#	CAS#	WE#	命令 / 动作
任何	H L	X H	X H	X H	COMMAND INHIBIT(NOP/继续先前的操作) NO OPERATION(NOP/继续先前的操作)
空闲	L L L L	L L L L	H L L H	H H L L	ACTIVE(选择和激活一行) LOAD COMMAND REGISTER(装载命令寄存器) LOAD MODE REGISTER(装载模式寄存器) ACTIVE TERMINATE(工作终结)
行工作的	L L L L	H H L L	L L H L	H L L H	READ(选择列和开始 READ 脉冲串) WRITE(选择列和开始 WRITE) ACTIVE TERMINATE LOAD COMMAND REGISTER
读	L L L L L	H H L H L	L L H H L	H L L L H	READ(选择列和开始新的 READ 脉冲串) WRITE(选择列和开始 WRITE) ACTIVE TERMINATE BURST TERMINATE LOAD COMMAND REGISTER
写	L L	H L	L L	H H	READ(选择列和开始新的 READ 脉冲串) LOAD COMMAND REGISTER

[0106] 真值表 5- 当前状态组 n- 对组 m 的命令

[0107]

当前状态	CS#	RAS#	CAS#	WE#	命令 / 动作
任何	H L	X H	X H	X H	COMMAND INHIBIT(NOP/继续先前的操作) NO OPERATION(NOP/继续先前的操作)
空闲	X	X	X	X	任何命令, 否则允许到组 m
行激活的, 工作的, 或 工作终结	L L L L L	L H H L L	H L L H L	H H L L H	ACTIVE(选择和激活行) READ(选择列和开始 READ 脉冲串) WRITE(选择列和开始 WRITE) ACTIVE TERMINATE LOAD COMMAND REGISTER
读	L L L L L	L H H L L	H L L H L	H H L L H	ACTIVE(选择和激活行) READ(选择列和开始新的 READ 脉冲串) WRITE(选择列和开始 WRITE) ACTIVE TERMINATE LOAD COMMAND REGISTER
写	L L L L L	L H L H L	H L H H L	H H L L H	ACTIVE(选择和激活行) READ(选择列和开始 READ 脉冲串) ACTIVE TERMINATE BURST TERMINATE LOAD COMMAND REGISTER

[0108] 功能说明

[0109] 同步快闪存储器引用多个特性, 以使得它理想地适用于在 SDRAM 总线上的代码存储和执行适当的应用。存储器阵列被分段成各个擦除块。每个块可被擦除, 而不影响被存储在其他的块中的数据。这些存储器块通过对命令执行逻辑 130(CEL) 发出命令而被读出、写入和擦除。CEL 控制内部状态机 132(ISM) 的操作, 它完全控制所有的 ERASENVMODE REGISTER(擦除 NV 模式寄存器)、WRITE NVMODE REGISTER(写入 NV 模式寄存器)、WRITE(写入)、BLOCK ERASE(块擦除)、BLOCK PROTECT(块保护)、DEVICE PROTECT(器件保护)、UNPROTECT ALL BLOCK(非保护所有的块) 和 VERIFY(验证) 操作。ISM 132 保护每个存储

单元 不受到过分擦除,以及使得每个存储单元最佳化,以便最大数据保持。另外,ISM 大大地简化在系统中或在外编程器中为了写入器件所必须的控制。

[0110] 同步快闪存储器被组织为 16 个独立的可擦除的存储器块,它们允许擦除存储器的一部分而不影响其余的存储器数据。任何的块可以是硬件保护的,免受不利的擦除或写入。保护的块要求在被修正之前 RP# 管脚被驱动到 VHH(相当高的电压)。在位置 0 和 15 处的 256K 字的块可以具有附加的硬件保护。一旦对于这些块执行了 PROTECT BLOCK(保护块)命令,UNPROTECT ALL BLOCKS 命令就松开除了位置 0 和 15 处的块以外的所有的块,除非 RP# 管脚是处在 VHH。这在系统内固件更新期间对于关键的代码提供附加的安全性,如果出现不想要的功率扰动或系统复位的话。

[0111] 功率接通初始化、ERASE、WRITE、和 PROTECT 的时序可通过使用 ISM 来控制存储器阵列中所有的编程算法而被简化。ISM 确保保护不被过分擦除,以及使得对每个单元的写余量最佳化。在 WRITE 操作期间,ISM 自动加增量以及监视 WRITE 企图,验证在每个存储单元上写余量以及更新 ISM 状态寄存器。当 BLOCK ERASE 操作被执行时,ISM 自动地过写整个寻址的块(消除过分擦除),加增量和监视 ERASE 企图,以及设置在 ISM 状态寄存器中的比特。

[0112] 8 比特 ISM 状态寄存器 134 允许外部处理器 200 在 WRITE、ERASE 和 PROTECT 操作期间监视 ISM 的状态。8 比特状态寄存器(SR7)的一个比特完全被 ISM 设置和清除。这个比特表示 ISM 是否忙于 ERASE、WRITE 或 PROTECT 任务。附加的错误信息在三个其他比特中(SR3、SR4 和 SR5)被设置:写入和保护块错误,擦除和非保护所有的块错误,以及器件保护错误。状态寄存器比特 SR0,SR1 和 SR2 提供 ISM 操作进行中的细节。用户可监视器件级或组级 ISM 操作(包括哪个组处在 ISM 控制下)是否正在进行中。这六个比特(SR3-SR5)必须被主系统清除。下面参照表 2 更详细地描述状态寄存器。

[0113] CEL 130 接受和解释对器件的命令。这些命令控制 ISM 和读路径(即,存储器阵列,器件配置或状态寄存器)的操作。在 ISM 是工作的同时,命令可被发出到 CEL。

[0114] 为了允许最大功率保存,同步快闪存储器的特征在于非常低的电流的、深的功率降低模式。为了进入这个模式,RP# 管脚 140(复位/功率降低)被取为 $V_{SS} \pm 0.2V$ 。为了防止有害的 RESET,RP# 必须在器件进入复位模式之前保持为 V_{SS} 在 100ns 内。对于 RP# 保持在 V_{SS} ,器件将进入深度功率降低模式。在器件进入深度功率降低模式后,在 RP# 上从 LOW 到 HIGH 的转移,将导致如这里概述的、器件功率接通初始化序列。在进入复位模式后但在进入深度功率降低模式之前 RP# 从 LOW 到 HIGH 的转移需要在发出可执行的命令之前 $1\mu s$ 的延时。当器件进入到深度功率降低模式时,不包括 RP# 缓冲器的所有的缓冲器被禁止,以及电流抽取是低的,例如,在 3.3V V_{CC} 下 $50\mu A$ 的最大值。在深度功率降低期间,加到 RP# 的输入必须保持在 V_{SS} 。进入到 RESET(复位)模式,清除状态寄存器 134,以及把 ISM 132 设置为阵列读模式。

[0115] 同步快闪存储器阵列结构被设计成允许扇区被擦除,而不打扰阵列的其余部分。阵列被划分成 16 个可寻址的“块”,它们是可独立地擦除的。通过擦除块,而不是整个阵列,总的器件耐久性被增强,作为系统灵活性。只有 ERASE 和 BLOCK PROTECT 功能是面向块的。16 个可寻址的块被相等地划分成四个组 104、106、108 和 110,每个组四个块。四个组具有同时读-写功能。对于任何组的 ISM WRITE 或 ERASE 操作可以与对任何其他组的 READ 操

作同时进行。可以对状态寄存器 134 进行轮询,以确定哪个组处于 ISM 操作中。同步快闪存储器具有单个基础操作 ISM,控制功率接通初始化、ERASE、WRITE、和 PROTECT 操作。在任何时间只可进行一个 ISM 操作;然而,某些其他命令,包括 READ 操作,可以在进行 ISM 操作的同时被执行。由 ISM 控制的操作命令被规定为组级操作或器件级操作。WRITE 及 ERASE 为组级 ISM 操作。在 ISM 组操作被起动后,对组的任何位置的 READ 可能输出不正确的数据,而对任何其他组的 READ 将读出阵列。READ STATUS REGISTER 命令将输出状态寄存器 134 的内容。ISM 状态比特将表示,ISM 操作何时完成 ($SR7 = 1$)。当 ISM 操作完成时,组将自动进入阵列读模式。ERASE NVMODE REGISTER, WRITE NVMODE REGISTER, BLOCK PROTECT, DEVICE PROTECT, 和 UNPROTECT ALL BLOCK 是器件级的 ISM 操作。一旦 ISM 器件级操作被起动,对任何组的 READ 将输出阵列的内容。READ STATUS REGISTER(读状态寄存器)命令可被发出,以确定 ISM 操作的完成。当 $SR7 = 1$ 时,ISM 操作将完成以及以后的 ISM 操作可被起动。任何块可以用硬件 电路保护,不受到不想要的 ERASE 或 WRITE 的影响,它要求 RP# 管脚在 WRITE 或 ERASE 开始之前被驱动到 VHH,正如下面解释的。

[0116] 任何块可被硬件保护,以提供对于固件的最敏感部分的额外保护。在 WRITE 或 ERASE 硬件保护的块期间,RP# 管脚必须保持在 VHH 上,直至 WRITE 或 ERASE 完成为止。在不是 $RP\# = VHH$ 的情形下,对于保护的块的任何 WRITE 或 ERASE 企图将被阻止,以及将导致写或擦除错误。在位置 0 和 15 处的块可以具有附加的硬件保护,以阻止不利的 WRITE 或 ERASE 操作。在本实施例中,这些块不可以通过 UNPROTECT ALL BLOCK 命令被软件松开的,除非 $RP\# = VHH$ 。任何块的保护状态可以通过用 READSTATUS REGISTER 命令读出它的块保护比特而被检验。另外,为了保护一个块,必须与块地址一起发出三周期命令序列。

[0117] 同步快闪存储器的特征在于,具有三种不同类型的 READ。取决于模式,READ 操作将从存储器阵列、状态寄存器、或器件配置寄存器之一产生数据。对器件配置寄存器或状态寄存器的 READ 必须在 LCR-ACTIVE 周期之后,以及输出数据的脉冲串长度将由模式寄存器设置值规定。以后的 READ 或不在 LCR-ACTIVE 周期之后的 READ 就读出阵列。然而,存在几个差别,并将在以下的节中被描述。

[0118] 对任何组的 READ 命令输出存储器阵列的内容。在进行 WRITE 或 ERASE ISM 操作的同时,在 ISM 控制下对组中的任何位置的 READ 可能输出不正确的数据。在退出 RESET 操作后,器件将自动进入到阵列读模式。

[0119] 对状态寄存器 134 进行的 READ,要求与在读阵列时相同的输入顺序,除了 LCR READ STATUS REGISTER(70H) 周期必须在 ACTIVE READ 周期之前以外。状态寄存器数据输出的脉冲串长度由模式寄存器 148 规定。状态寄存器内容在经过 CAS 等待时间的下一个时钟上升沿被更新和锁存。器件将自动进入到阵列读模式,用于以后的 READ。

[0120] 对任何的器件配置寄存器 136 进行的 READ,要求与在读状态寄存器时相同的输入顺序,除了必须发出特定的地址以外。WE# 必须是 HIGH(高电平),以及 DQM 和 CS# 必须是 LOW(低电平)。为了读出制造商兼容性 ID,地址必须是在 000000H,以及为了读出器件 ID,地址必须是在 000001H。任何的块保护比特在每个擦除块内的第三地址 (xx0002H) 处被读出,而器件保护比特从位置 000003H 处被读出。

[0121] DQ 管脚被使用来输入数据或用来输入阵列。地址管脚被使用来规定地址位置或在 LOAD COMMAND REGISTER 周期期间对 CEL 输入命令。命令输入把 8 比特命令发出到 CEL,以

便控制器件的工作模式。WRITE 被使用来对存储器阵列输入数据。以下的节描述这两种输入类型。

[0122] 为了执行命令输入, DQM 必须是 LOW, 以及 CS# 和 WE# 必须是 LOW。地址管脚或 DQ 管脚被使用来输入命令。不被使用于输入命令的地址管脚是“不用管它”的, 以及必须保持为稳定的。8 比特命令在 DQ0-DQ7 或 A0-A7 处被输入, 以及在时钟上升沿处被锁存。

[0123] 对存储器阵列的 WRITE 把想要的比特设置为逻辑 0, 但不能把给定的比特从逻辑 0 改变到逻辑 1。把任何比特设置为逻辑 1 需要把整个块擦除。为了执行 WRITE, DQM 必须是 LOW, CS# 和 WE# 必须是 LOW, 以及 VCCP 必须保持在 VCC。对保护的块的写入也要求 RP# 管脚保持在 VHH。A0-A11 提供要被写入的地址, 而要被写入到阵列的数据在 DQ 管脚处被输入。数据和地址在时钟的上升沿被锁存。WRITE 的前面必须有 WRITESSETUP 命令。

[0124] 为了简化写入存储器块, 同步快闪存储器引用 ISM, 它在 WRITE 和 ERASE 周期内控制所有的内部算法。8 比特命令组被使用来控制器件。对于正确的命令的清单, 可参阅真值表 1 和 2。

[0125] 8 比特 ISM 状态寄存器 134 (见表 2) 被轮询, 以检验 ERASE NVMODEREGISTER、WRITE NVMODE REGISTER、WRITE、ERASE、BLOCK PROTECT、DEVICE PROTECT、或 UNPROTECT ALL BLOCK 的完成或任何相关的错误。ISM 操作的完成可以通过发出 READ STATUS REGISTER(70H) 命令而被监视。状态寄存器的内容被输出到 DQ0-DQ7, 以及在由模式寄存器设置值规定的、固定脉冲串长度内在下一个时钟上升沿 (经过 CAS 等待时间) 处被更新。ISM 操作将在 SR7 = 1 时完成。所有的规定的比特由 ISM 设置, 但仅仅 ISM 状态比特被 ISM 复位。擦除的 / 非保护的块, 写入 / 保护的块, 器件保护必须通过使用 CLEAR STATUS REGISTER(50H) 命令清除。这允许用户选择何时轮询和清除状态寄存器。例如, 主系统可以在检验状态寄存器之前执行多个 WRITE 操作, 而不是在每次单独的 WRITE 以后检验。坚持 RP# 信号或将器件功率降低也将清除状态寄存器。

[0126] 表 2 状态寄存器

[0127]

状态比特	状态寄存器比特	说明
SR7	ISM 状态 1 = 准备好 0 = 忙	当执行 WRITE 或 BLOCK ERASE 时, ISM 比特 显示状态机的工作的状态。控制逻辑轮询这个比特, 以确定何时擦除和写入状态比特是正确的。
SR6	保留	保留供将来使用。
SR5	擦除 / 非保护块的状态 1 = 块擦除或块非保护出错 0 = 成功的块擦除非保护	ES 在最大数目的 ERASE 周期由 ISM 执行 (但不带有成功验证) 后被设置为 1。如果 BLOCK UNPROTECT 操作不成功, 则这个比特也被设置为 1。ES 只由 CLEAR STATUS REGISTER 命令或由 RESET (复位) 清除。
SR4	写入 / 保护块的状态 1 = 写入或块保护出错 0 = 成功的写入或块保护	WS 在最大数目的 WRITE 周期由 ISM 执行 (但不带有成功验证) 后被设置为 1。如果 BLOCK 或 DEVICE PROTECT 操作不成功, 则这个比特也被设置为 1。WS 只由 CLEAR STATUS REGISTER 命令或由 RESET (复位) 清除。
SR2 SR1	组 A1 ISM 状态 组 A0 ISM 状态	当 SRO = 0 时, 在 ISM 控制下的组可以从 BA0, BA1:[0,0] 组 0; [0,1] 组 1; [1,0] 组 2; [1,1] 组 3 中译码。
SR3	器件保护状态 1 = 被保护的、企图进行不正确的操作的器件 0 = 非保护的或满足 RP# 条件的器件	如果试图进行不正确的 WRITE, ERASE, PROTECT BLOCK, PROTECT DEVICE 或 UNPROTECT ALL BLOCKS, 则 DPS 被设置为 1。在这些命令之一被发出后, 把 RP#, 块保护比特和器件保护比特的条件进行比较, 以确定是否允许想要的操作。必须由 CLEAR STATUS REGISTER 命令或由 RESET 清除。
SRO	器件 / 组 ISM 状态 1 = 器件级 ISM 操作 0 = 组级 ISM 操作	如果 ISM 操作是器件级操作, 则 DPS 被设置为 1。对于阵列的任何的组的正确的 READ 可紧接着在器件级的 ISM WRITE 操作后面。当 DPS 被设置为 0 时, ISM 操作是组级操作。在 ISM 控制下对组的 READ 可能导致不正确的数据。SR2 和 SR3 可被译码, 以确定哪个组是在 ISM 控制下。

[0128] 器件 ID、制造商兼容性 ID、器件保护状态和块保护状态都可通过发出 READ DEVICE CONFIGURATION(90H) 命令而被读出。为了读想要的寄存器,必须坚持特定的地址。对于各种器件配置寄存器 136 的更多的细节,可参阅表 3。

[0129] 表 3 器件配置

[0130]

器件配置	地址	数据	条件
制造商兼容性	000000H	2CH	读出制造商兼容性
器件 ID	000001H	D3H	读出器件 ID
块保护比特	xx0002H	DQ0 = 1 DQ0 = 0	块被保护 块未被保护
器件保护比特	000003H	DQ0 = 1 DQ0 = 0	阻止进行块保护修正 允许进行块保护修正

[0131] 可发出命令,以使器件进入不同的操作模式。每个模式具有特定的操作,这些操作可在该模式下被执行。几种模式要求一系列命令在它们到达之前被写入。以下的节描述每个模式的性质,真值表 1 和 2 列出对于执行想要的操作所需要的所有的命令序列。读 - 写

功能允许对于任何组执行基础操作写或擦除,而同时读出任何其他组。对于写操作,在真值表 2 中的 LCR-ACTIVE-WRITE 命令序列必须在接连的时钟周期内完成。然而,为了简化同步快闪控制器操作,在命令序列中可以发出没有限制的数目的 NOP 或 COMMAND INHIBIT。为了附加保护,在三个周期内,这些命令序列必须具有相同的组地址。如果在 LCR-ACTIVE-WRITE 命令序列期间组地址改变,或如果命令序列不是接续的(除了 NOP 和 COMMAND INHIBIT 以外,它们是许可的),写和擦除状态比特(SR4 和 SR5)将被设置,以及操作被禁止。

[0132] 在功率接通以后和在对器件发出任何工作命令以前,同步快闪存储器被初始化。在功率被加到 VCC、VCCQ 和 VCCP(同时地)以及时钟是稳定以后,RP# 从 LOW 转移到 HIGH。在 RP# 转移到 HIGH 以后,需要一个延时(在一个实施例中,100 μ s 的延时),以便完成内部器件初始化。在完成器件初始化时器件处在阵列读模式,以及可执行的命令可被发出到器件。

[0133] 为了读出器件 ID、制造商兼容性 ID、器件保护比特和每个块保护比特,发出 READ DEVICE CONFIGURATION(90H) 命令。在这个模式下,特定的地址被发出来读出想要的信息。制造商兼容性 ID 在 000000H 处被读出;器件 ID 在 000001H 处被读出。制造商兼容性 ID 和器件 ID 在 DQ0-DQ7 处输出。器件保护比特在 000003H 处被读出;以及每个块保护比特在每个块内的第三地址位置(xx0002H)处被读出。器件和块保护比特在 DQ0 处输出。

[0134] 需要在接连的时钟边沿处的三个接连的命令来把数据输入到阵列(NOP 和 COMMAND INHIBIT 许可处在周期之间)。在第一周期,利用在 A0-A7 处的 WRITE SETUP(40H) 给出 LOAD COMMAND REGISTER 命令,以及组地址在 BA0、BA1 上被发出。下一个命令是 ACTIVE,它激活行地址和确认组地址。第三周期是 WRITE,在此期间发布开始的列、组地址、和数据。ISM 状态比特在接着的时钟边沿(经过 CAS 等待时间)处被设置。在 ISM 执行 WRITE 时,ISM 状态比特(SR7)将是 0。在 ISM 控制下对组的 READ 操作可能产生不正确的数据。当 ISM 状态比特(SR7)被设置为逻辑 1 时,WRITE 被完成,以及组将处在阵列读模式,并准备好可执行的命令。写入到硬件保护的块,也要求 RP# 管脚在第三周期(WRITE)之前被设置为 VHH,以及 RP# 必须保持在 VHH,直至 ISM WRITE 操作完成为止。如果 LCR-ACTIVE-WRITE 命令序列没有在接连的周期完成或组地址在三个周期的任一个内改变,则写和擦除状态比特(SR4 和 SR5)被设置。在 ISM 起动 WRITE 后,它不能被中断,除了由 RESET 或通过功率降低这个部件以外。在 WRITE 期间这样做,可能打乱被写入的数据。

[0135] 执行 ERASE 序列将把块内的所有的比特设置为逻辑 1。对于执行 ERASE 所必须的命令序列类似于 WRITE 的命令序列。为了提供对于偶然块擦除的附加安全性,需要在接连的时钟边沿处的三个接连的命令来起动块的 ERASE。在第一周期,把 A0-A7 处的 ERASE SETUP(20H) 发布给 LOAD COMMAND REGISTER,以及要被擦除的块的组地址在 BA0, BA1 上被发出。下一个命令是 ACTIVE,其中 A10, A11, BA0, BA1 提供要被擦除的块的地址。第三周期是 WRITE,在此期间在 DQ0-DQ7 处给出 ERASE CONFIRM(D0H),以及重新发布组地址。ISM 状态比特在接着的时钟边沿(经过 CAS 等待时间)处被设置。在 ERASE CONFIRM(D0H) 被发布后,ISM 将开始所寻址的块的 ERASE。对于其中存在所寻址的块的组的任何的 READ 操作可能输出不正确的数据。当 ERASE 操作完成时,该组处在阵列读模式,并准备好可执行的命令。擦除被硬件保护的块,也要求 RP# 管脚在第三周期(WRITE)之前被设置为 VHH,以及 RP# 必须保持在 VHH,直至 ERASE 完成(SR7 = 1)为止。如果 LCR-ACTIVE-WRITE 命令序列

没有在接连的周期完成 (NOP 和 COMMAND INHIBIT 许可处在周期之间) 或组地址在一个或多个命令周期内改变, 则写和擦除状态比特 (SR4 和 SR5) 将被设置, 以及操作被禁止。

[0136] 模式寄存器 148 的内容可以用 WRITE NVMODE REGISTER 命令被复制到 NV 模式寄存器 147 中。在写入到 NV 模式寄存器之前, ERASE NVMODEREGISTER 命令序列必须被完成, 以便把 NV 模式寄存器中所有的比特设置为逻辑 1。为了执行 ERASE NVMODE REGISTER 和 WRITE NVMODEREGISTER 所必须的命令序列类似于 WRITE 的命令序列。对于完成 ERASE NVMODE REGISTER 和 WRITE NVMODE REGISTER 所必须的 LCR-ACTIVE-WRITE 命令的更多的信息可参阅真值表 2。在 ERASE NVMODEREGISTER 或 WRITE NVMODE REGISTER 命令序列的 WRITE 周期被登记后, READ 命令可被发出到阵列。在当前的 ISM 操作被完成和 SR7 = 1 之前, 将不允许新的 WRITE 操作。

[0137] 执行 BLOCK PROTECT 序列, 对于给定的块允许进行第一级的软件 / 硬件保护。存储器包括 16 比特寄存器, 它具有相应于 16 个可保护的块的一个比特。存储器也具有寄存器, 以提供被使用来保护整个器件不受到写入和擦除操作的器件比特。为了执行 BLOCK PROTECT 所必须的命令序列类似于 WRITE 的命令。为了提供对于偶然块保护的附加安全性, 需要三个接连的命令周期来起动 BLOCK PROTECT。在第一周期, 把 A0-A7 处的 PROTECT SETUP (60H) 命令发布给 LOAD COMMAND REGISTER, 以及要被保护的块的组地址在 BA0、BA1 上被发出。下一个命令是 ACTIVE, 它起动在要保护的块中的行, 以及确认组地址。第三周期是 WRITE, 在此期间在 DQ0-DQ7 处发布 BLOCK PROTECT CONFIRM (01H), 以及重新发布组地址。ISM 状态比特在接着的时钟边沿 (经过 CAS 等待时间) 处被设置。ISM 将开始保护操作。如果 LCR-ACTIVE-WRITE 没有在接连的周期完成 (NOP 和 COMMAND INHIBIT 许可处在周期之间) 或组地址改变, 则写和擦除状态比特 (SR4 和 SR5) 将被设置, 以及操作被禁止。当 ISM 状态比特 (SR7) 被设置为逻辑 1 时, PROTECT 已完成, 以及该组将处在阵列读模式, 并准备好可执行的命令。一旦块保护比特被设置为 1 (保护的), 它只能被复位为 0, 如果 UNPROTECT ALL BLOCKS 命令的话。UNPROTECT ALL BLOCKS 命令序列类似于 BLOCK PROTECT 命令; 然而, 在第三周期, 利用 UNPROTECT ALL BLOCKS CONFIRM (D0H) 命令发出 WRITE, 以及地址是“不用管它”。对于附加信息, 参阅真值表 2。在位置 0 和 15 处的块具有附加安全性。一旦在位置 0 和 15 处的块保护比特被设置为 1 (保护的), 每个比特就只能被复位为 0, 如果 RP# 在 UNPROTECT 操作的第三周期之前被加到 VHH, 以及保持在 VHH, 直至操作完成 (SR7 = 1) 为止。另外, 如果器件保护比特被设置, 则 RP# 必须在第三周期之前被加到 VHH, 及保持在 VHH, 直至 BLOCK PROTECT 或 UNPROTECT ALL BLOCKS 操作完成为止。为了检验块的保护状态, 可以发出 READ DEVICE CONFIGURATION (90H) 命令。

[0138] 执行 DEVICE PROTECT 序列, 把器件保护比特设置为 1, 以及阻止块保护比特修正。为了执行 DEVICE PROTECT 所必须的命令序列类似于 WRITE 的命令。需要三个接连的命令周期来起动 DEVICE PROTECT 序列。在第一周期, 把 A0-A7 处的 PROTECT SETUP (60H) 命令发布给 LOADCOMMAND REGISTER, 以及组地址在 BA0、BA1 上被发出。组地址是“不用管它”, 但相同的组地址必须在所有的三个周期内被使用。下一个命令是 ACTIVE。第三周期是 WRITE, 在此期间在 DQ0-DQ7 处发布 DEVICEPROTECT (F1H), 以及 RP# 被加到 VHH。ISM 状态比特在接着的时钟边沿 (经过 CAS 等待时间) 处被设置。可执行的命令可被发布到器件。RP# 必须保持在 VHH, 直至 WRITE 完成 (SR7 = 1) 为止。在当前的 ISM 操作完成前, 将不允许新的

WRITE 操作。一旦器件保护比特被设置,它就不能被复位到 0。对于器件保护比特被设置为 1, BLOCK PROTECT 或 BLOCKUNPROTECT 被阻止,除非 RP# 在任一个操作期间处在 VHH。器件保护比特不影响 WRITE 或 ERASE 操作。对于块和器件保护操作的更多的信息,可参阅表 4。

[0139] 表 4 保护操作真值表

[0140]

功能	RP#	CS#	DQM	WE#	地址	VccP	DQ0-DQ7
非保护的器件							
PROTECT SETUP	H	L	H	L	60H	X	X
PROTECT BLOCK	H	L	H	L	BA	H	01H
PROTECT DEVICE	V _{HH}	L	H	L	X	X	F1H
UNPROTECT ALL BLOCKS	H/V _{HH}	L	H	L	X	H	D0H
被保护的器件							
PROTECT SETUP	H 或 V _{HH}	L	H	L	60H	X	X
PROTECT BLOCK	V _{HH}	L	H	L	BA	H	01H
UNPROTECT ALL BLOCKS	V _{HH}	L	H	L	X	H	D0H

[0141] 在 ISM 状态比特 (SR7) 被设置后,器件 / 组 (SR0)、器件保护 (SR3)、组 A0 (SR1)、组 A1 (SR2)、写 / 保护块 (SR4) 和擦除 / 非保护 (SR5) 状态比特可被检验。如果 SR3、SR4、SR5 状态比特之一或组合被设置,则在操作期间出现错误。ISM 不能复位 SR3、SR4 或 SR5 比特。为了清除这些比特,必须给出 CLEAR STATUS REGISTER(50H) 命令。表 5 列出错误的组合。

[0142] 表 5 状态寄存器错误译码

[0143]

状态比特			错误说明
SR5	SR4	SR3	
0	0	0	无错误
0	1	0	WRITE, BLOCK PROTECT 或 DEVICE PROTECT 错误
0	1	1	不正确的 BLOCK PROTECT 或 DEVICE PROTECT, RP#不正确 (V _{HH})
0	1	1	不正确的 BLOCK 或 DEVICE PROTECT, RP#不正确 (V _{HH})
1	0	0	ERASE 或 ALL BLOCK UNPROTECT 错误
1	0	1	不正确的 ALL BLOCK UNPROTECT, RP#不正确 (V _{HH})
1	1	0	命令序列错误

[0144] 同步快闪存储器被设计和被制造,满足先进的代码和数据存储要求。为了确保这个可靠性级, VCCP 在 WRITE 或 ERASE 期间必须保持为 Vcc。在这些限制以外的操作可以减小可对器件执行的 WRITE 和 ERASE 周期的数目。每个块是对于 100,000WRITE/ERASE 周期

的耐久性的最小值被设计和被处理的。

[0145] 同步快闪存储器提供可以在阵列读模式下被利用来节省功率的几个功率节省特性。深度功率降低模式可以通过把 RP# 加到 $VSS \pm 0.2V$ 而被允许。在这个模式下的电流抽取 (ICC) 是低的, 诸如 $50 \mu A$ 的最大值。当 CS# 是 HIGH 时, 器件将进入工作等待模式。在这个模式下, 电流也是低的, 诸如 30mA 的最大 ICC 电流。如果在写、擦除、或保护操作期间 CS# 被加到 HIGH, 则 ISM 将继续 WRITE 操作, 以及器件消耗工作的 Iccp 功率, 直至操作完成为止。

[0146] 参照图 16, 图上表示按照本发明的一个实施例的自定时写序列的流程图。序列包括装载命令寄存器 (代码 40H)、接受工作的命令和行地址、以及接受写命令和列地址。然后, 序列提供状态寄存器轮询, 以确定写入是否完成。该轮询监视状态寄存器比特 7 (SR7), 以确定它是否被设置为 1。可以包括可任选的状态检验。当写入完成时, 阵列被置于阵列读模式。

[0147] 参照图 17, 图上提供了按照本发明的一个实施例的完全的写状态 - 检验序列的流程图。该序列查看状态寄存器比特 4 (SR4), 以确定它是否被设置为 0。如果 SR4 是 1, 则在写操作中有错误。序列也查看状态寄存器比特 3 (SR3), 以确定它是否被设置为 0。如果 SR3 是 1, 则在写操作期间有不正确的写错误。

[0148] 参照图 18, 图上提供按照本发明的一个实施例的自定时块擦除序列的流程图。该序列包括装载命令寄存器 (代码 20H), 以及接受工作的命令和行地址。存储器然后确定块是否被保护。如果它不被保护, 则存储器对该块执行写操作 (D0H), 以及对于完成情形监视状态寄存器。可以执行任选的状态检验, 以及存储器被置于阵列读模式。如果块是保护的, 则不允许擦除, 除非 RP# 信号处在提高了的电压 (VHH)。

[0149] 图 19 表示按照本发明的一个实施例的完全的块擦除状态 - 检验序列的流程图。该序列监视状态寄存器, 以确定是否出现命令序列错误 (SR4 或 SR5 = 1)。如果 SR3 被设置为 1, 则出现不正确的擦除或非保护的错误。最后, 如果 SR5 被置为 1, 则发生块擦除或非保护错误。

[0150] 图 20 是按照本发明的一个实施例的块保护序列的流程图。该序列包括装载命令寄存器 (代码 60H), 以及接受工作的命令和行地址。存储器然后确定块是否被保护。如果它不被保护, 则存储器对该块执行写操作 (01H), 以及对于完成情形监视状态寄存器。可以执行任选的状态检验, 以及存储器被置于阵列读模式。如果块是保护的, 则不允许擦除, 除非 RP# 信号处在提高了的电压 (VHH)。

[0151] 参阅图 21, 图上提供了按照本发明的一个实施例的完全的块状态 - 检验序列的流程图。该序列监视状态寄存器比特 3、4、和 5, 以确定是否检测到错误。

[0152] 图 22 是按照本发明的一个实施例的器件保护序列的流程图。该序列包括装载命令寄存器 (代码 60H), 以及接受工作的命令和行地址。存储器然后确定 RP# 是否处在 VHH。存储器执行写操作 (F1H), 以及对于完成情形监视状态寄存器。可以执行任选的状态检验, 以及存储器被置于阵列读模式。

[0153] 图 23 是按照本发明的一个实施例的块非保护序列的流程图。该序列包括装载命令寄存器 (代码 60H), 以及接受工作的命令和行地址。存储器然后确定该存储器件是否被保护。如果它不被保护, 则存储器确定引导位置 (块 0 和 15) 是否被保护。如果没有一个

块被保护,则存储器对该块执行写操作 (D0H),以及对于完成情形监视状态寄存器。可以执行任选的状态检验,以及存储器被置于阵列读模式。如果器件是保护的,则不允许擦除,除非 RP# 信号处在提高了的电压 (VHH)。同样地,如果引导位置是保护的,则存储器确定是否所有的块应当是不保护的。

[0154] 图 24 显示初始化和装载模式寄存器运行的时序。模式寄存器通过接受装载模式寄存器命令和在地址线上接受工作代码 (操作码) 而被编程。把运行码装入该模式寄存器中。如上所述,在功率接通后,非易失性模式寄存器的内容被自动地装载到模式寄存器,以及不一定需要模式寄存器操作。

[0155] 图 25 表示时钟中止模式操作的时序,以及图 26 表示另一个脉冲串读操作的时序。图 27 表示交替的组读出访问的时序。这里,需要工作的命令来改变组地址。图 28 上表示全页脉冲串读操作。应当指出,完全页脉冲串不是自终结的,而需要终结的命令。

[0156] 图 29 通过使用数据屏蔽信号表示读操作的时序。DQM 信号被使用来屏蔽数据输出,这样,在 DQ 连接上不提供 Dout m+1。

[0157] 参照图 30,图上表示写操作的时序,后面跟随对不同的组的读出。在这个操作中,对组 a 执行写操作,以及对组 b 执行随后的读出。在每个组中访问同一个行。

[0158] 参照图 31,图上表示写操作的时序,后面跟随对同一个组的读出。在这个操作中,对组 a 执行写操作,以及对组 a 执行随后的读出。对于读操作访问不同的行,以及存储器必须等待先前的写操作被完成。这是与图 30 的读操作不同的,其中读操作并不由于写操作而被延时。

[0159] 同步快闪存储器提供没有等待时间的写操作。这是与 SDRAM 不同的,SDRAM 要求系统提供对于写操作的等待时间,就像读操作那样。所以写操作并不从系统总线中取走与 SDRAM 花费的一样多的周期。因此,可改进系统读通过量,见图 12,其中写入的数据 Din 在与写命令和列地址同一个时钟周期上被提供。图 12 的时钟周期 T1 不需要是 NOP 命令 (见图 30)。读命令可以在跟随在写数据后面的下一个时钟周期上被提供。因此,虽然读操作需要 DQ 连接在读命令后的预定的数目的时钟周期内 (等待时间) 保持为可以得到的,但 DQ 连接可以紧接在写命令被提供后 (没有等待时间) 被使用。这样,本发明允许零总线换向能力。这是大大地不同于 SDRAM 的,在 SDRAM 中当在读和写操作之间交替时在系统总线上需要大量等待。同步快闪存储器提供这两个特性,以及可提高总线通过量。

[0160] 参照图 32,本发明的系统 32 包括同步存储器 302,它具有内部的写锁存器 304,被使用来存储在 DQ 输入端 306 上接受的写数据。写锁存器被耦合到存储器阵列 310。另外,存储器阵列可被安排成多个可寻址的块。数据可被写入到一个块,而同时可对其他的块执行读操作。阵列的存储单元可以是非易失性存储单元。数据通信连接 306 被使用于与外部设备 (诸如处理器 320 或其他存储器控制器) 的双向通信。

[0161] 数据缓冲器 330 可被耦合到数据通信连接,以便管理双向数据通信。这个缓冲器可以是传统的 FIFO 或流水线的输入 / 输出缓冲器电路。写锁存器被耦合在缓冲器与存储器阵列之间,以便锁存在数据通信连接上提供的数据。最后,控制电路被提供来管理在阵列上执行的读和写操作。

[0162] 通过锁存输入写数据,数据总线 306 (DQ) 可被释放以及通过使用锁存的数据执行写操作。在执行第一写操作的同时,随后的、对存储器的写操作可被禁止。然而,可以得到

总线,以便在存储器上立刻执行读操作。本发明不应当与传统的输入/输出缓冲器结构混淆。也就是,虽然现有的存储器器件使用在 DQ 输入路径上的输入缓冲器和在 DQ 输出路径上的输出缓冲器,但是用于读和写操作的时钟等待时间被保持为相同的。本发明可包括输入/输出缓冲器电路来提供与 DQ 路径和外部处理器的接口。附加的写锁存器允许存储器来隔离对存储器的一个区域的写路径/操作,而同时允许在其他存储器区域上的数据读操作。

[0163] 现有的快闪存储器器件具有非常有限的同时操作能力。也就是,现有的快闪存储器典型地在执行写操作的同时,阻碍从存储器的读出。某些存储器器件通过中止正在进行的写操作、然后允许对阵列的读出,而允许写入的同时进行读出。另外的快闪存储器通过提供有限的扇区、这些扇区可被写入而存储器的其余部分可供读出使用,而允许写入的同时进行读出。这样的快闪存储器的目的是消除系统中对于分开的 EEPROM 的需要。有限的扇区空间提供在快闪存储器中的 EEPROM 单元,以及留下存储器的其余部分用于快闪操作。

[0164] 本发明提供被安排成类似于 SDRAM 的组结构的快闪阵列。在一个实施例中,64M 的同步快闪存储器被划分成四个组,它们具有与 64MSDRAM 相同的寻址。这些组被进一步分成更小的可寻址的扇区,它们可被擦除或被编程。存储器允许基于组的的同时的读和写。因此,一个组可被写入,而同时可以对任何其他的组执行同时读出。

[0165] 正如本领域技术人员已知的,SDRAM 可以在每个组打开一个公共的行。读和写操作可以在打开的行和跨过阵列的组上顺序地执行。

[0166] 本同步快闪存储器具有类似于 SDRAM 的组结构,允许在一个组被写入的同时在每个组中打开一行。参照图 33,图上表示本发明的处理系统 400 的一个实施例。同步快闪存储器 410 通过双向数据总线 435 被耦合到多个处理器 440,442,444 和 446。存储器包括被排列成多个组 412,414,416 和 418 的非易失性存储单元的阵列。一般地示出读/写电路 430 以便管理与阵列的数据通信。在操作时,处理器,诸如处理器 440,可起动对于阵列组 412 的行 420 的写操作。在执行写操作时,第二处理器,诸如处理器 442,可以从第二阵列组的行 420 中读出数据。这允许四个处理器在同步快闪存储器上独立地工作。本发明并不限于四个组或四个处理器。单个或多个处理器可对于一个组执行写操作,而同时从其余的存储器阵列组读出数据,例如见图 32。因此,可以同时进行写操作和多个同时的读操作。

[0167] 如上所述,本发明的同步快闪存储器可以执行组特定的读操作,而同时对另一个组执行写操作。当第一和第二个外部处理器试图对存储器读和写时,出现一个问题。也就是,如果处理器试图对同一个存储器块执行操作,则这两个处理器需要知道,正在对存储器执行哪些操作,以避免竞争。在现有的系统中,总线主机被使用来跟踪由多个处理器执行的同时操作。为了减小总线主机的开销,可以提供两个任选项。

[0168] 一个任选项是在存储器中包括自动读状态模式,它在试图对存储器执行第二操作时输出状态寄存器的内容。也就是,当对存储器执行写操作时,在存储器上执行的任何读操作会输出状态寄存器的内容。这个任选项通过输出状态数据而中断读操作,即使是试图对不同的阵列块执行读操作。第二个任选项是要求处理器读出状态寄存器来确定存储器的写状态。利用这个任选项再次中断读操作。

[0169] 本发明可提供两个状态读出模式,来避免在对于不同的阵列组的同时读操作时的干扰。在本同步快闪存储器中提供的第一状态模式提供整个存储器器件的状态。也就是,

状态寄存器表示是否正在对存储器阵列或非阵列寄存器执行写操作。这个状态模式，在这里被称为存储器状态模式，由来自外部处理器的请求而被选择地激活。响应于存储器状态命令，存储器的控制电路在 DQ 连接上提供状态寄存器数据。

[0170] 第二状态模式是存储器阵列组特定的模式。在这个模式下，状态数据被自动地提供在 DQ 连接上。也就是，如果处理器对于阵列组起动写操作，则该组被置于组状态模式。在写操作正在被执行的同时，任何以后的、从该组读出的企图导致在 DQ 连接上输出状态寄存器数据。因此，存储器控制电路允许多处理器系统互相不干扰地一起工作。

[0171] 存储器包括组寄存器 450 (图 33)，它可被设置来识别哪个阵列组正在被写入。在操作时，存储器对组寄存器进行编程，以及使用组寄存器作为指针。在读操作期间，把读访问与组寄存器进行比较，以及如果在该组地址处有正在进行的写操作，则存储器自动输出状态寄存器数据。所以，本发明可以减小总线主机的开销，因此，可以允许更好的多处理器能力。

[0172] 单个脉冲串状态

[0173] 如上所述，现有的快闪存储器包括单个状态读操作。在这些存储器中，一旦存储器进入写模式，从存储器读出的任何企图都提供状态寄存器的内容。这允许用户监视写操作如何在进行。

[0174] 本发明具有状态读模式，它用状态读命令输出来自状态寄存器的数据。而且，同步快闪存储器可以通过使用装载命令寄存器操作被编程，以建立脉冲串长度和时钟等待时间。这些设置值主要在读操作期间被使用来控制输出数据的时序。现有的快闪存储器间的问题是，输出的寄存器数据没有为受控的脉冲串长度和等待时间的输出作好准备。

[0175] 本存储器可被置于寄存器读模式，以及在一系列时钟周期（脉冲串）内输出寄存器数据。脉冲串的长度通过对模式寄存器进行编程而被预先规定，正如以上说明的。例如，模式寄存器可被设置为具有 4 个字的脉冲串长度和 3 的时钟等待时间。在接受到状态寄存器读命令后，由于时钟等待时间，本存储器在 3 个时钟周期后输出状态寄存器内容，以及由于脉冲串长度设置值，在 4 个周期内在总线（DQ 连接）上继续输出状态寄存器内容。寄存器读操作不限于状态寄存器读数，也可应用于读出器件识别寄存器、制造商识别寄存器、或用来存储操作数据的任何多个工作寄存器。

[0176] 通过使用脉冲串长度和时钟等待时间设置值来控制寄存器数据输出可减小在存储器器件中进行写操作的同时读出时的混淆。例如，如果用户需要知道在存储器器件中操作的状态，则可以起动读状态命令，以及状态寄存器数据在全脉冲串长度内被输出。

[0177] 结论

[0178] 同步快闪存储器包括非易失性存储单元的阵列。存储器阵列被排列成行和列，以及可被进一步排列成可寻址的块。数据通信连接被使用来与外部设备（诸如，处理器或其他存储器控制器）进行双向数据通信。存储器可以在一系列时钟周期期间在数据通信连接上输出来自存储寄存器的数据，以提供寄存器数据的脉冲串。存储器还可以按照规定的时钟等待时间值提供寄存器数据。寄存器数据可包括状态数据、操作设置值数据、制造识别、和存储器器件识别。

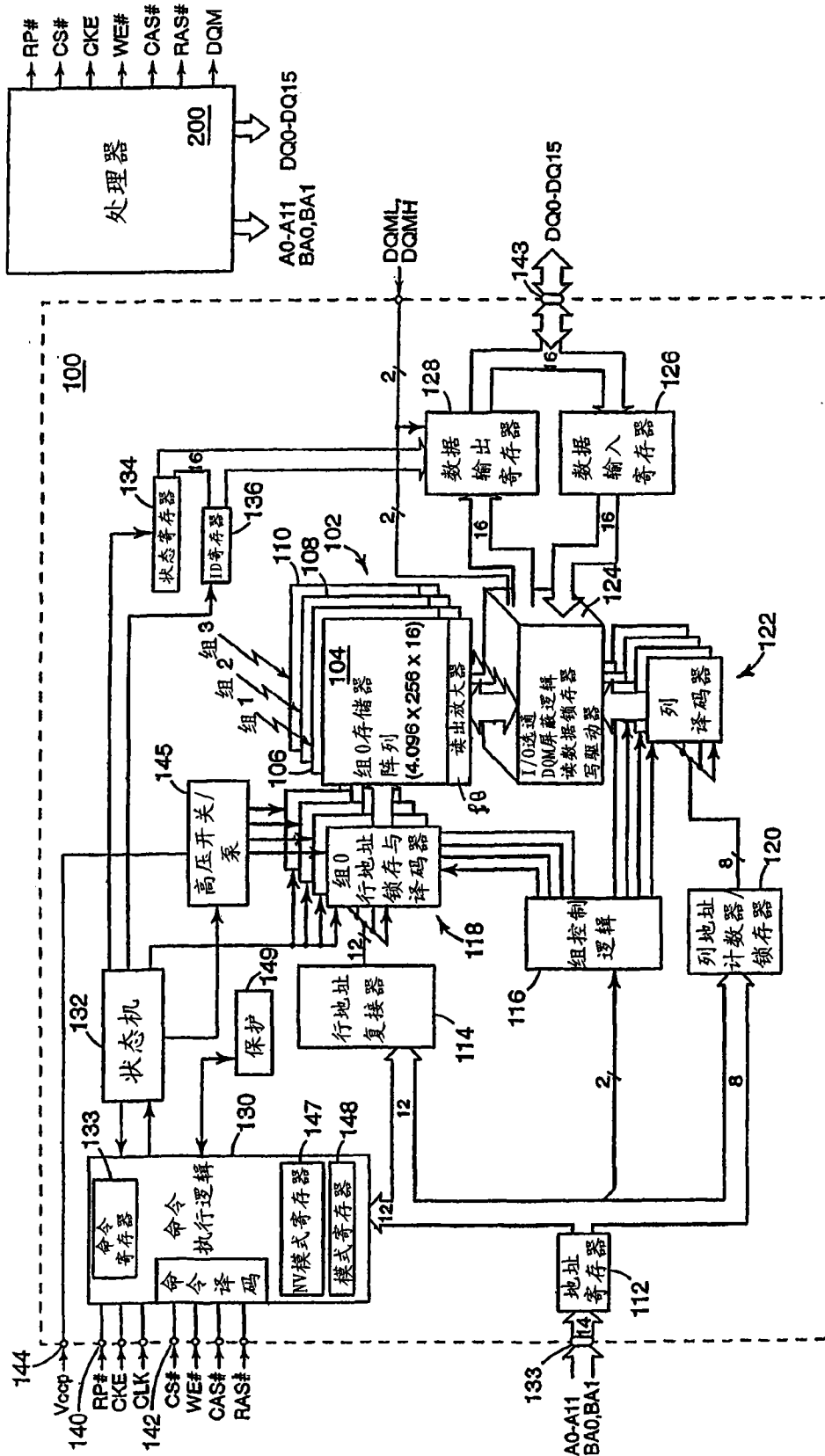


图 1A

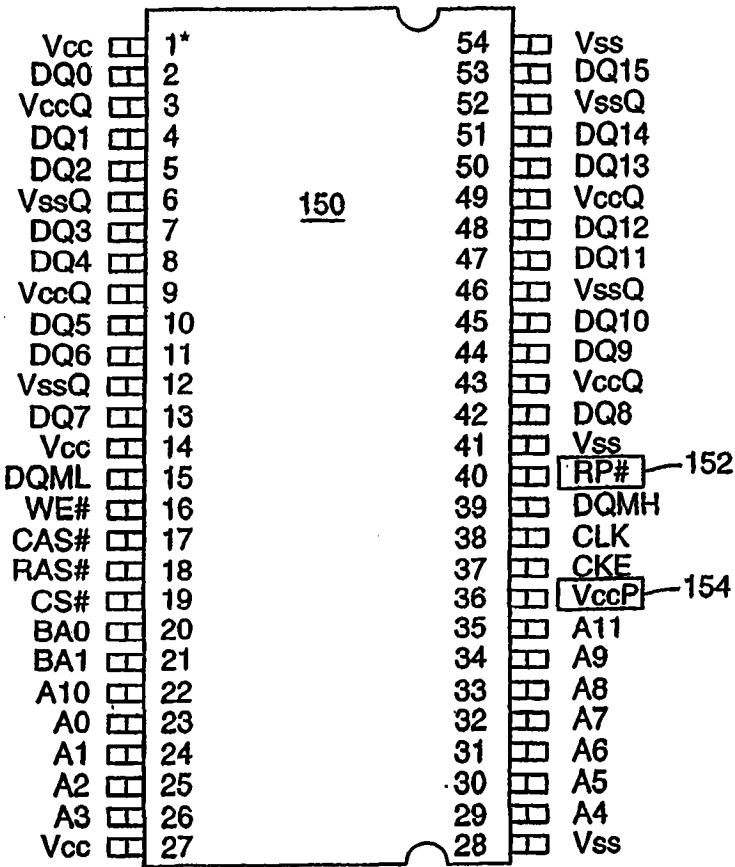


图 1B

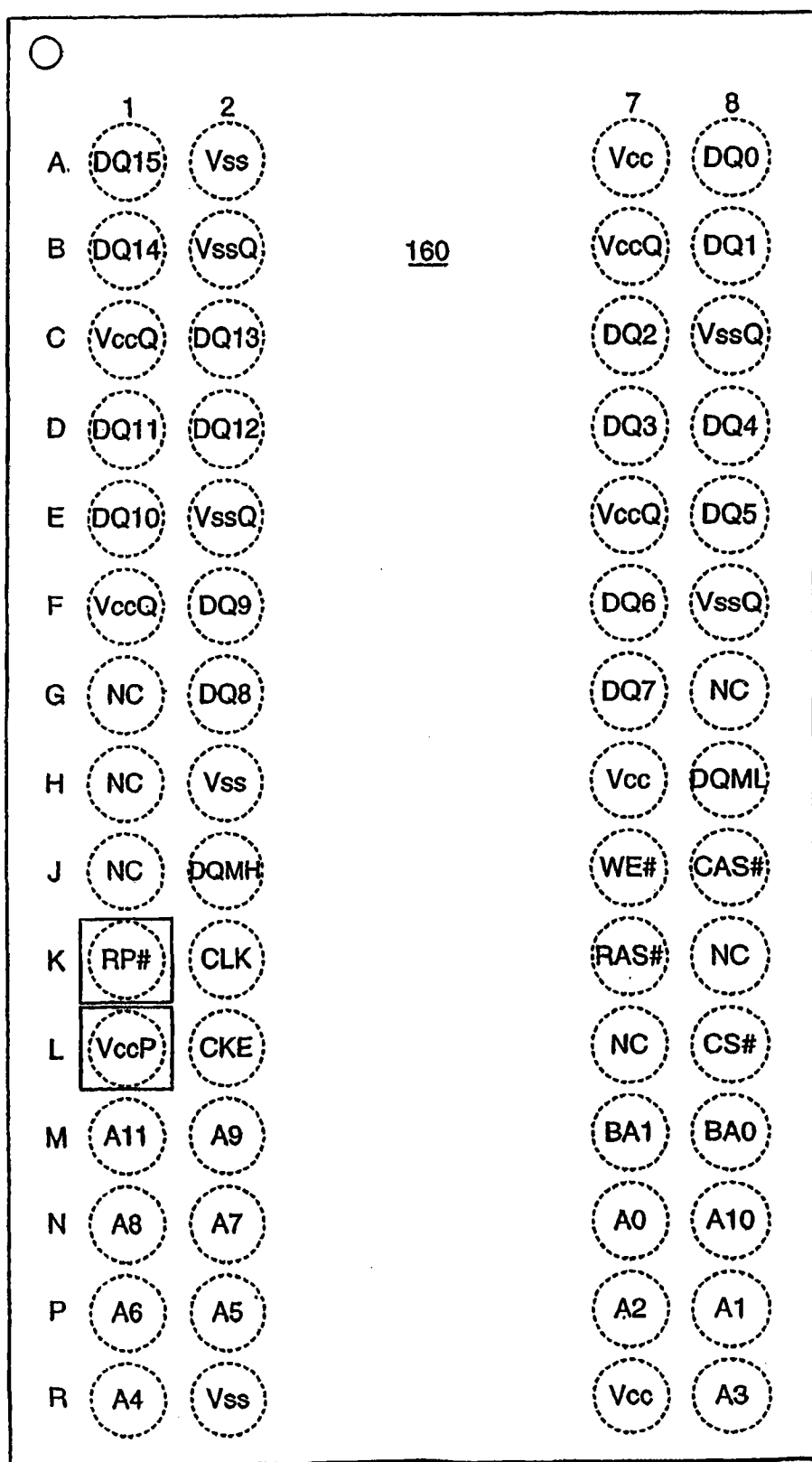


图 1C

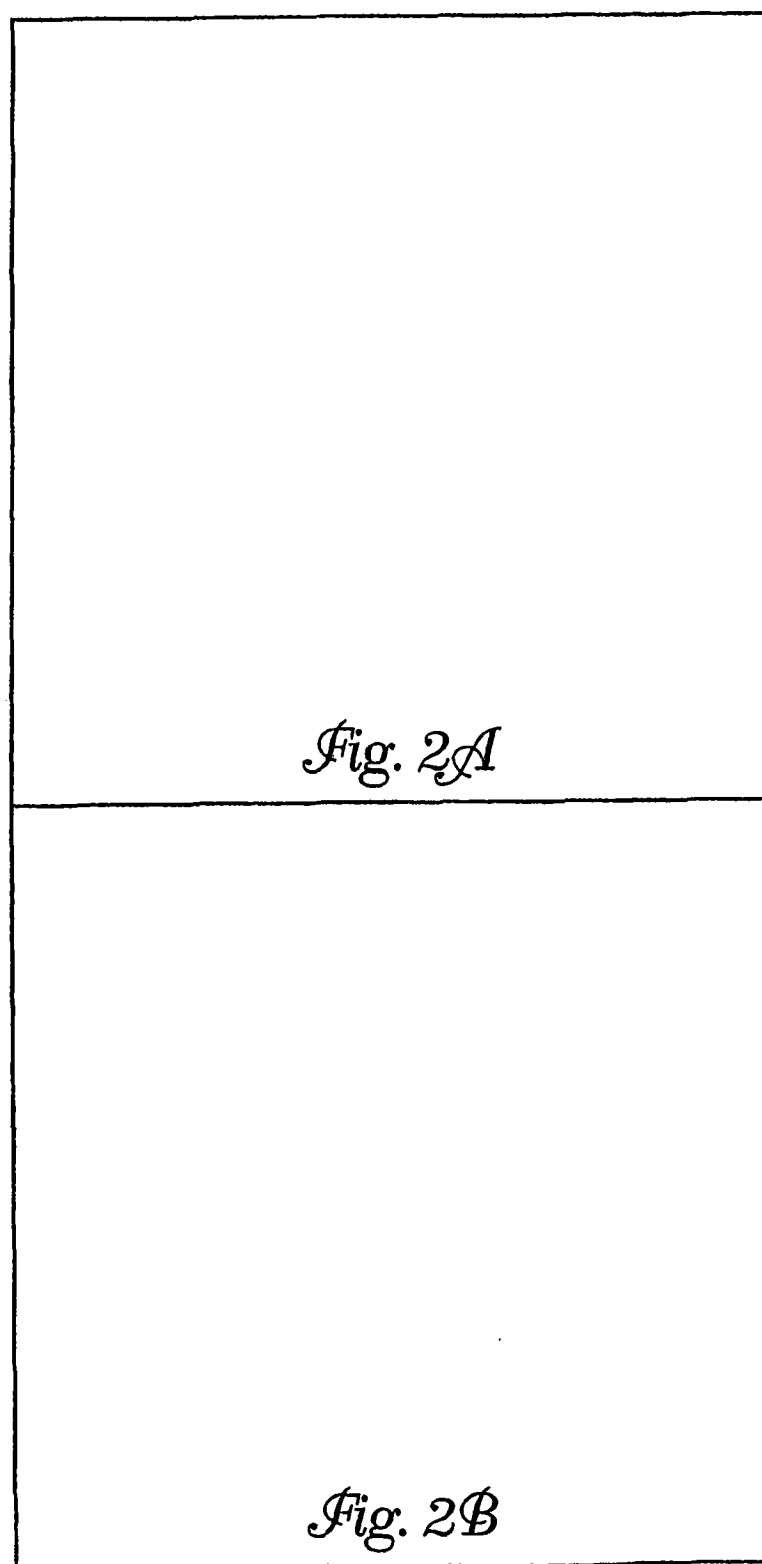


图 2

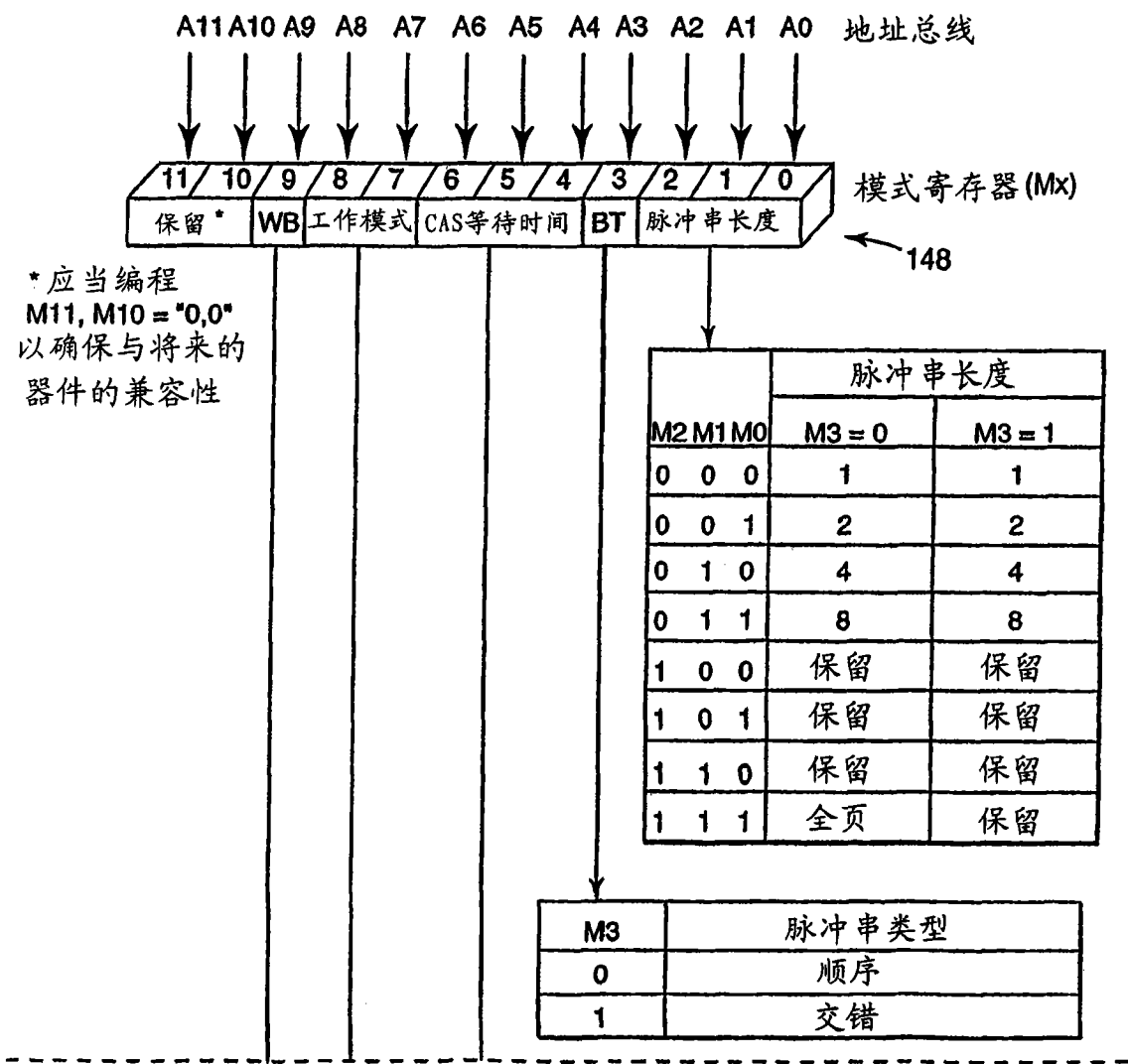


图 2A

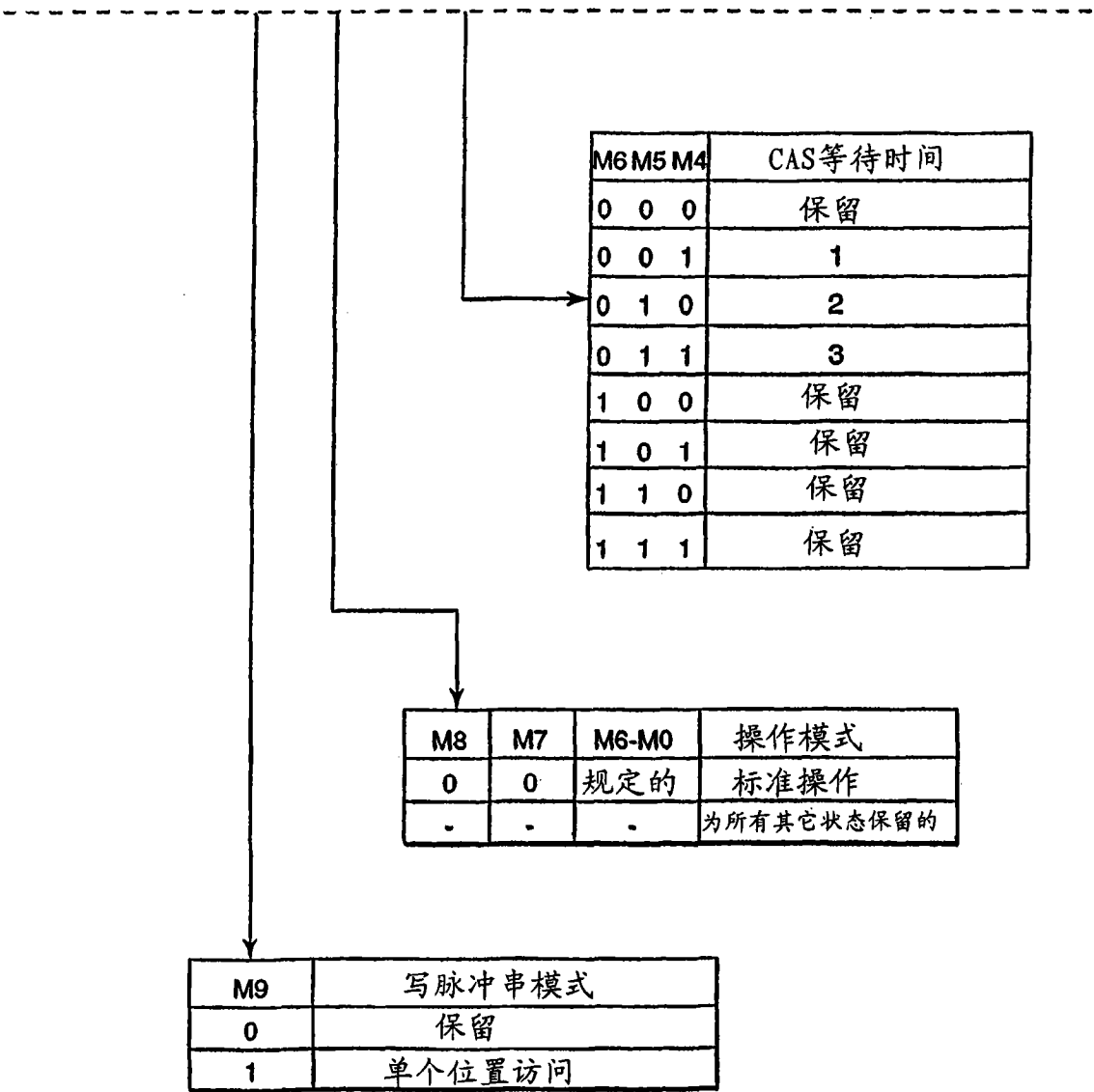


图 2B

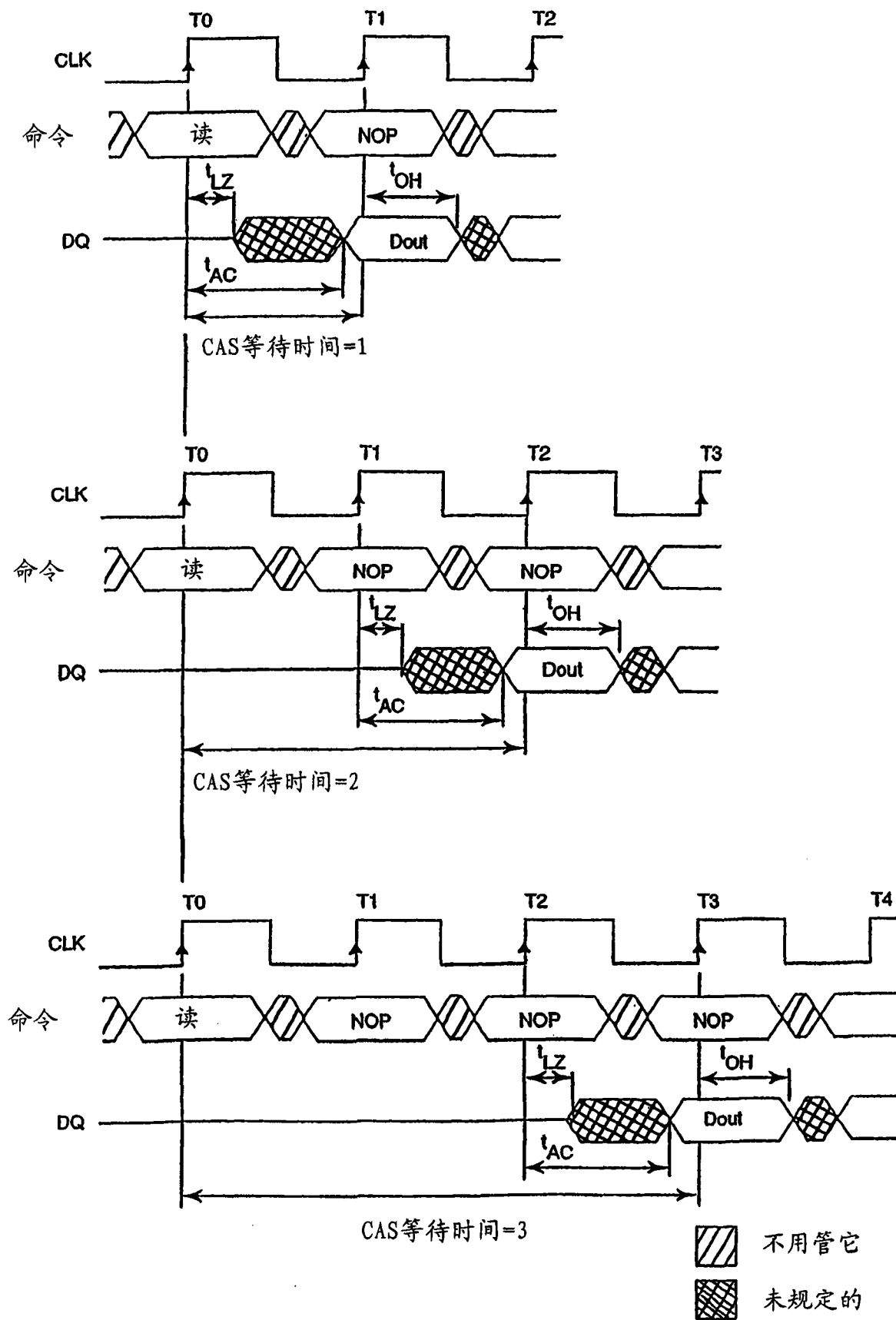


图 3

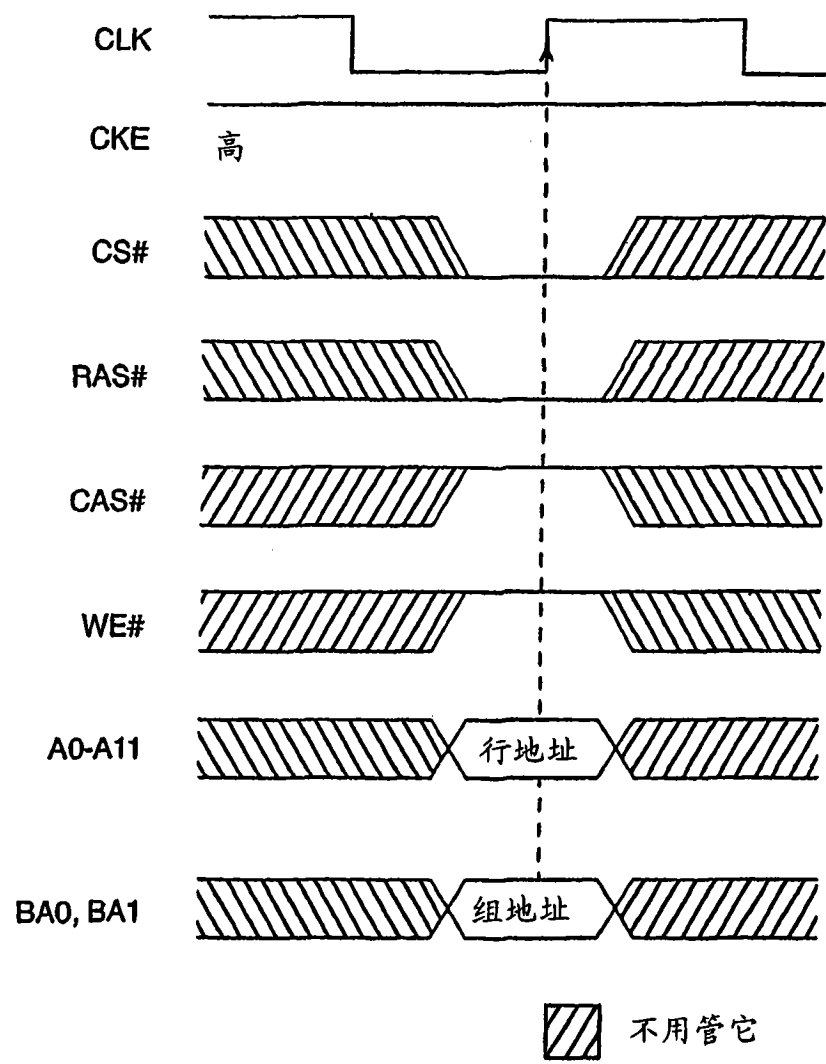


图 4

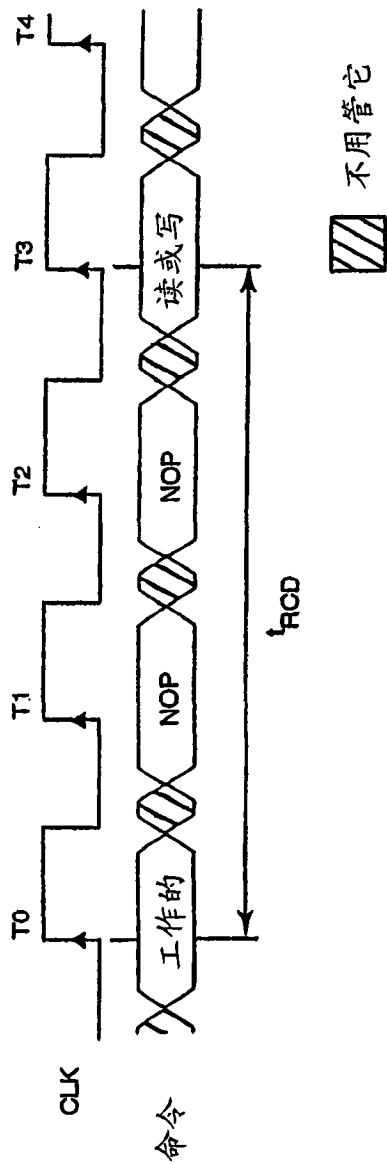


图 5

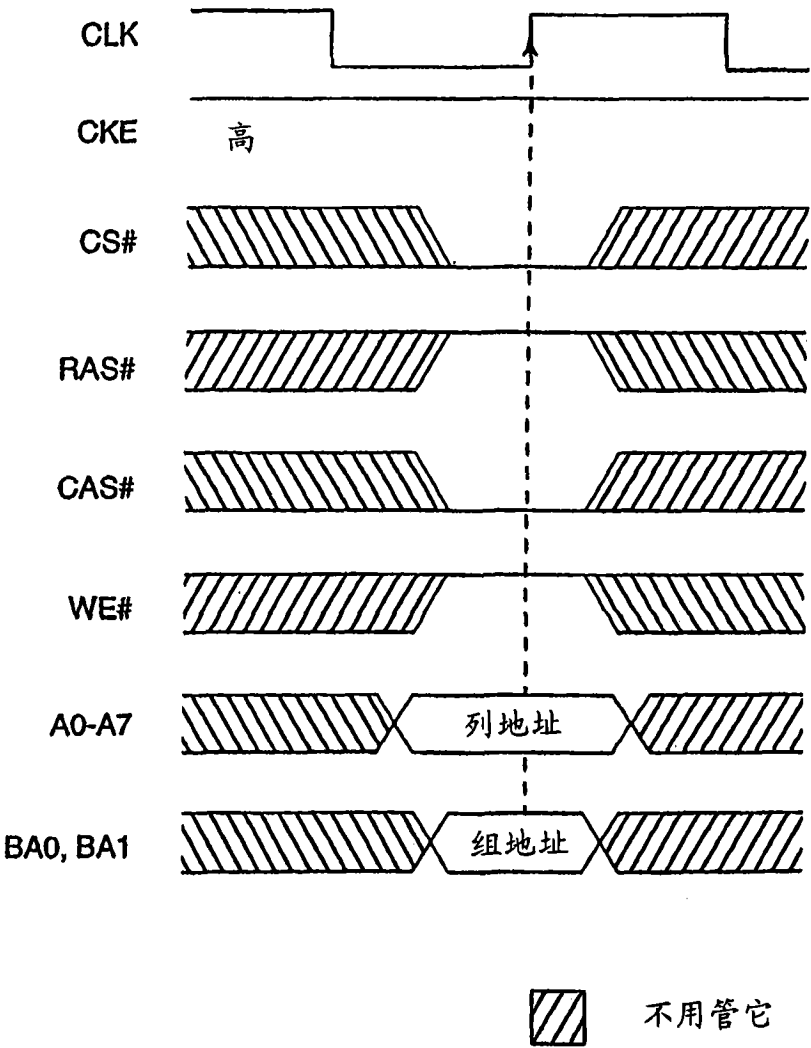
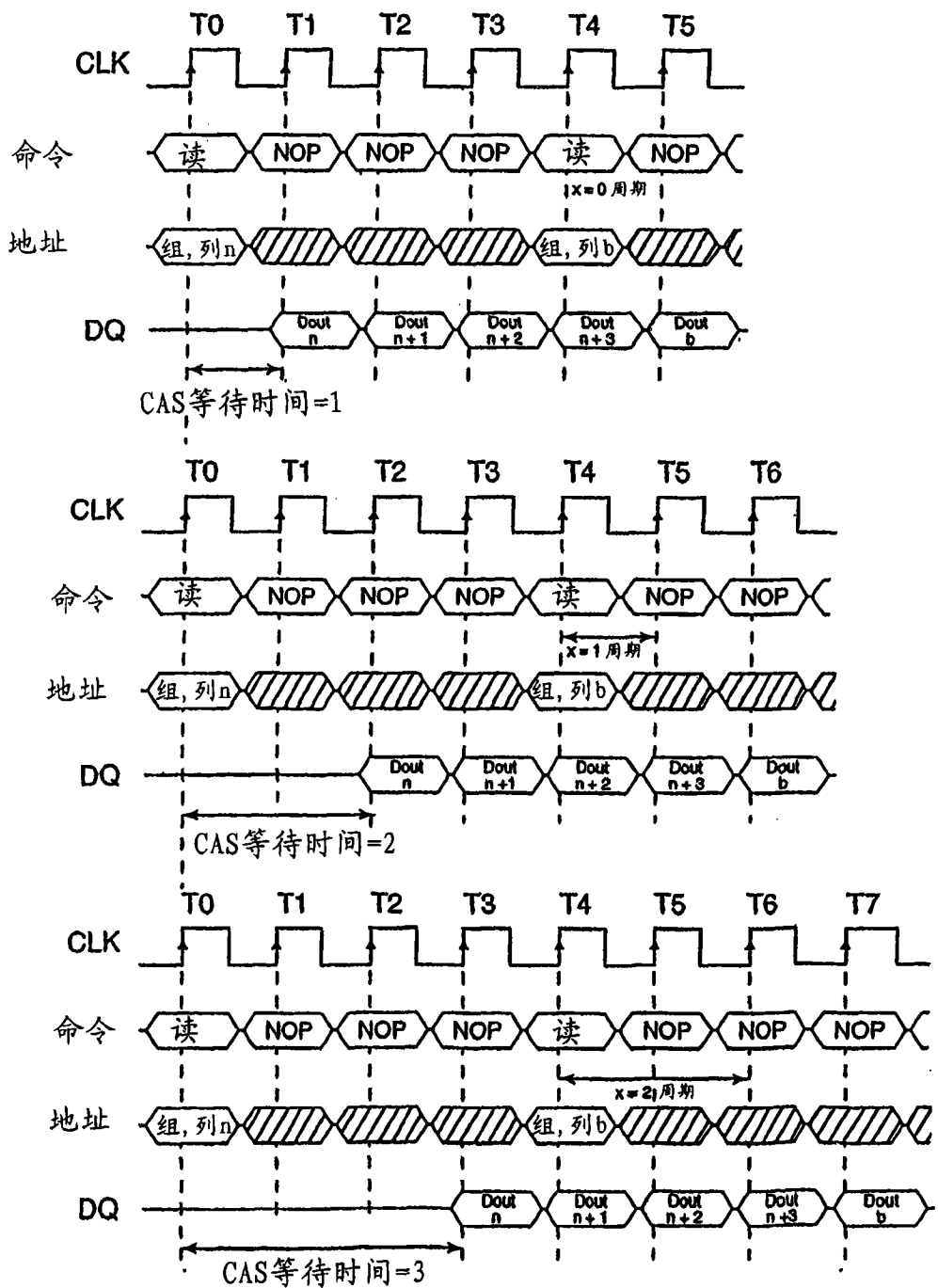


图 6



注: 每个READ命令可以对任一个组。DQM是低

图 7

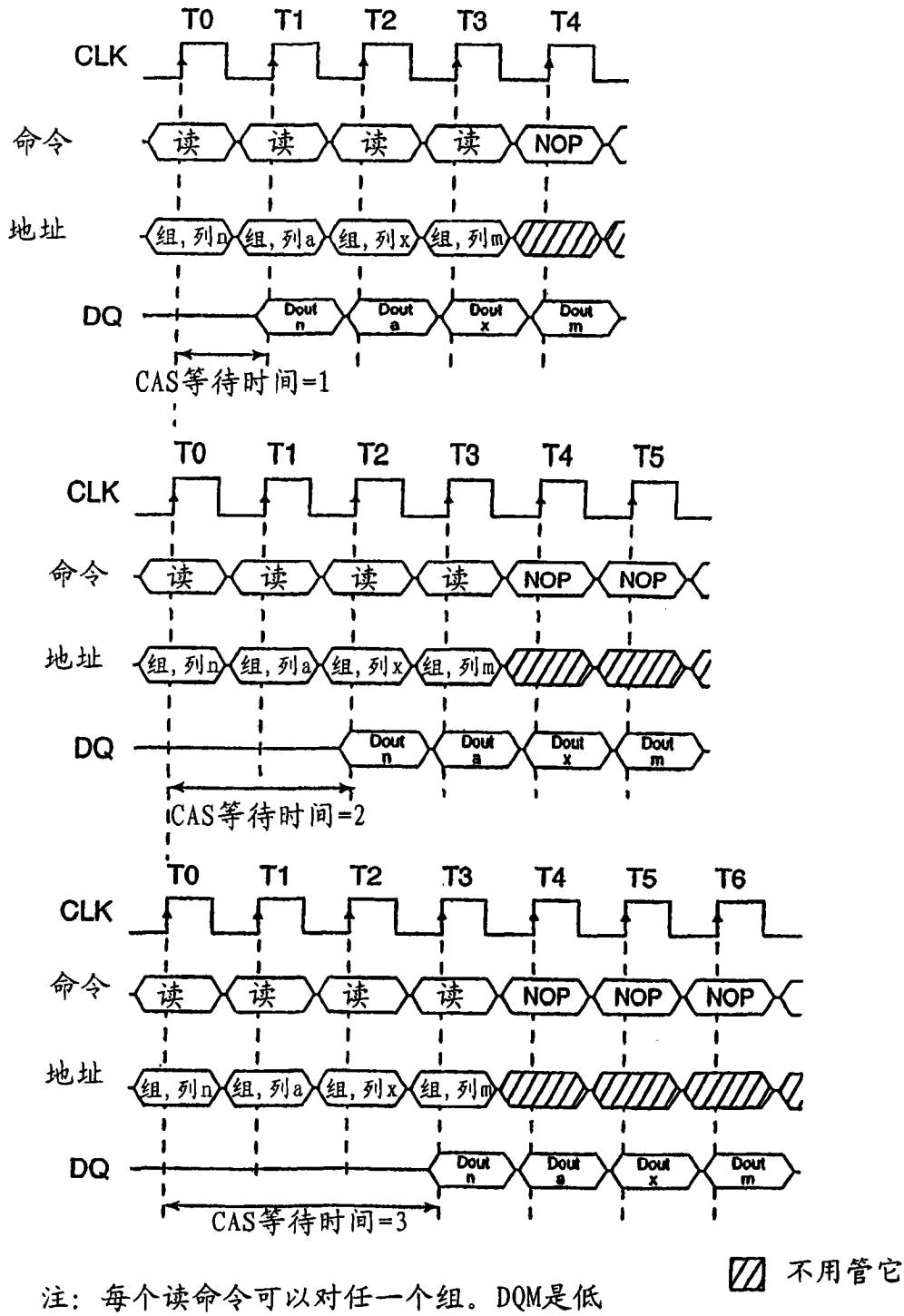


图 8

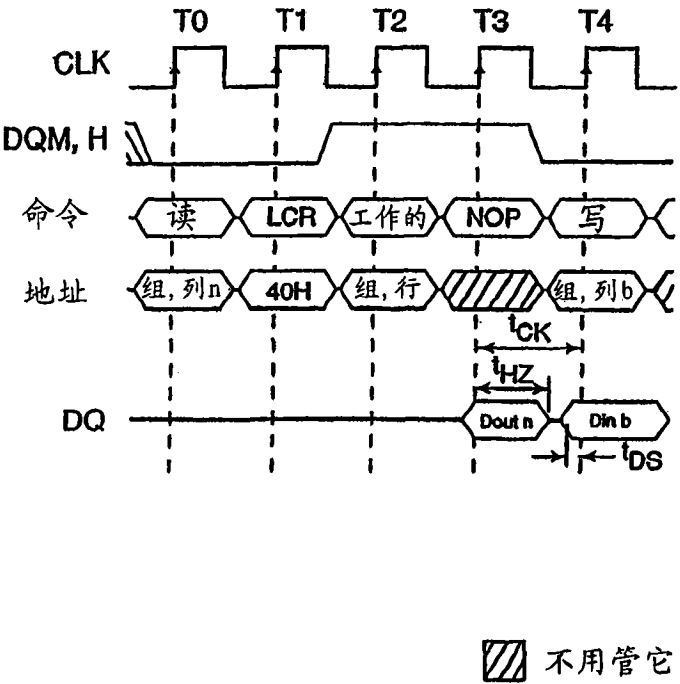
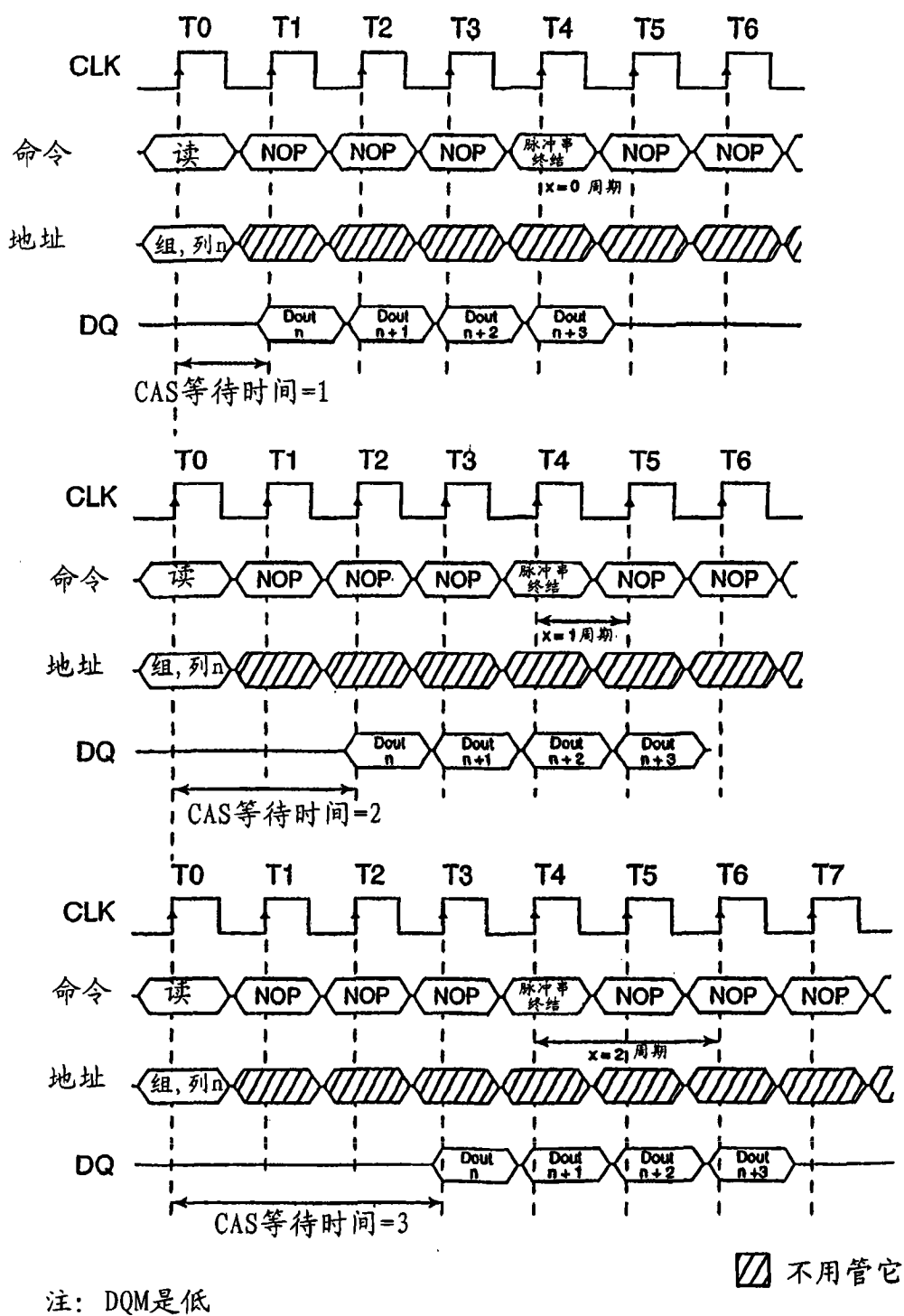


图 9



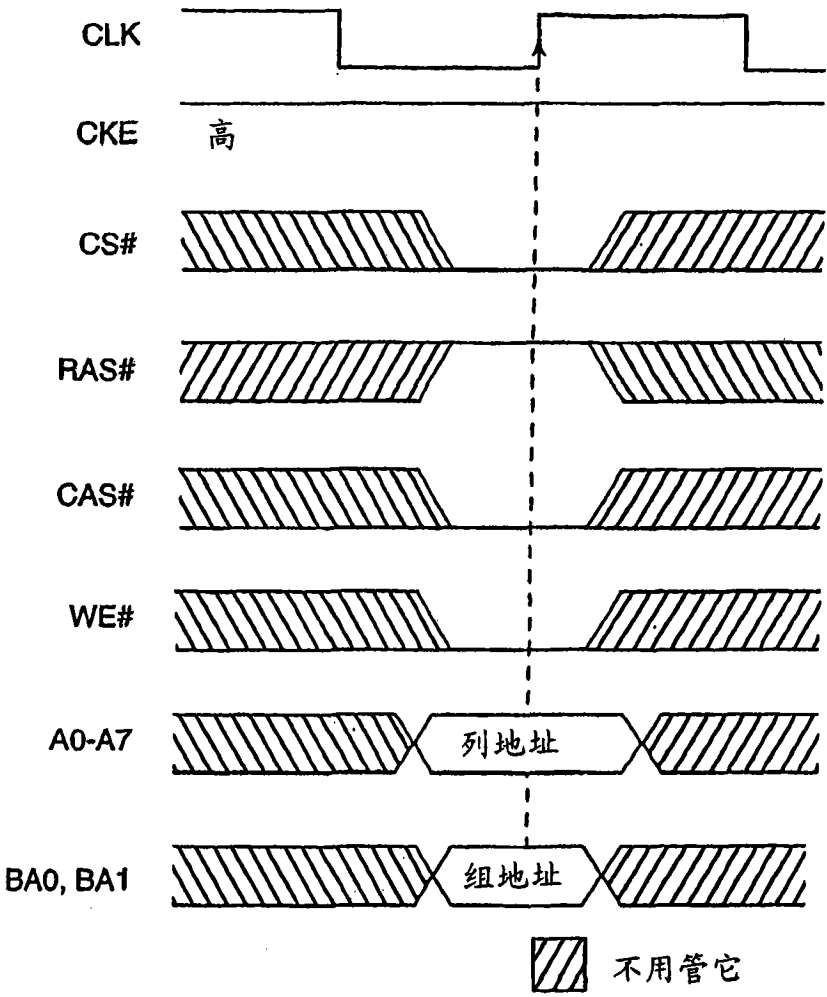


图 11

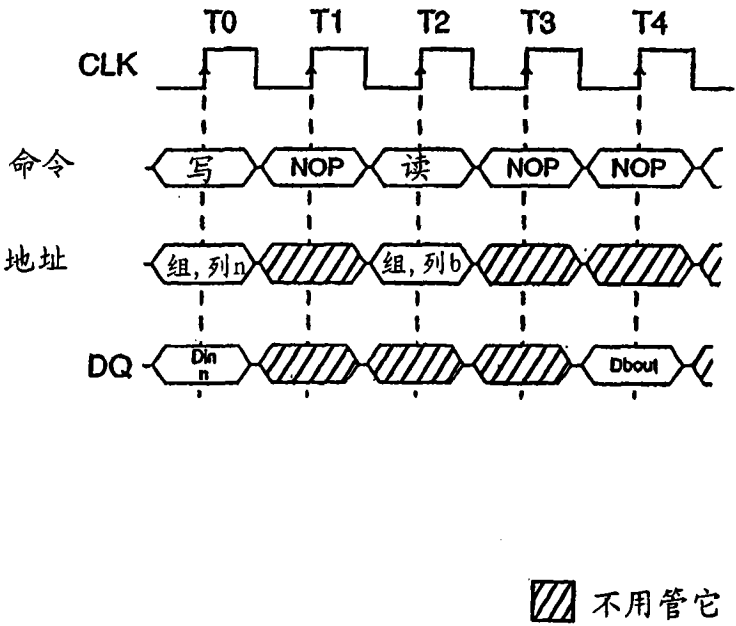


图 12

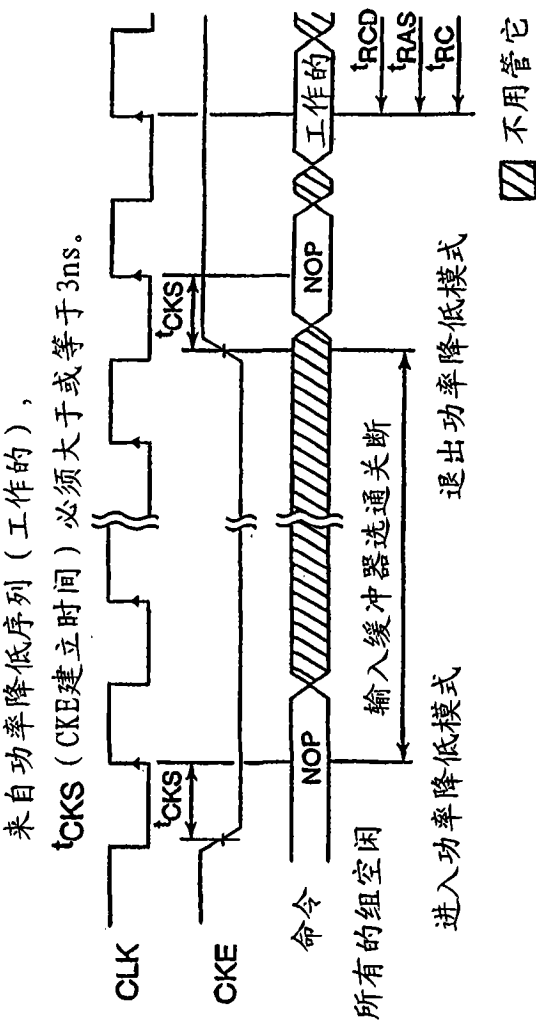


图 13

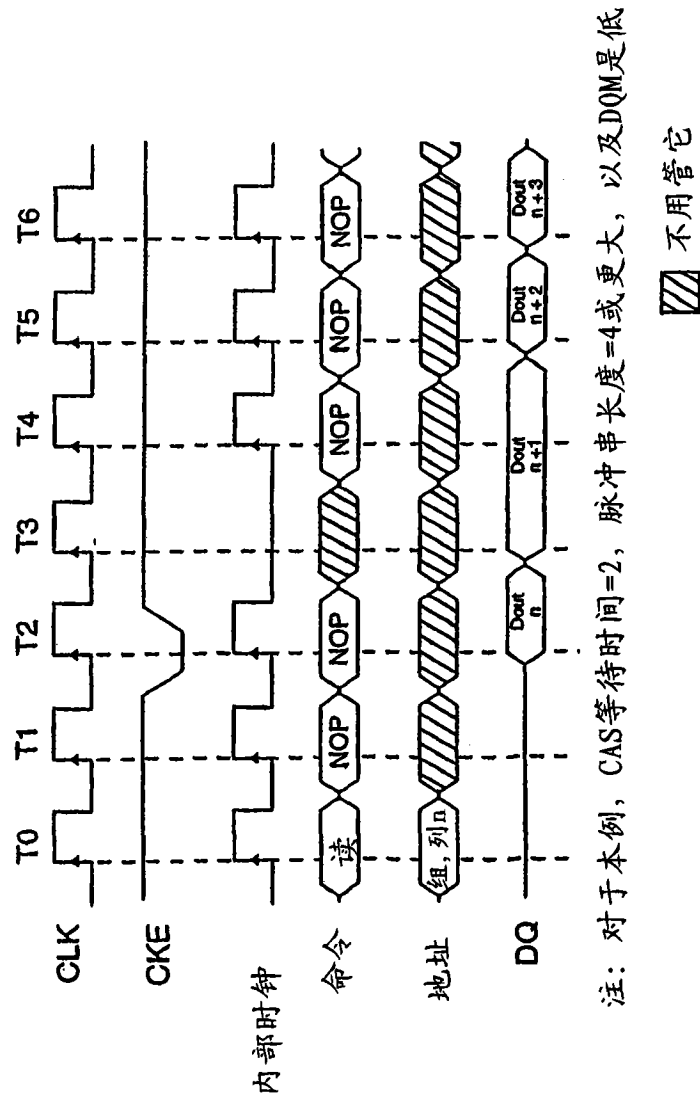


图 14

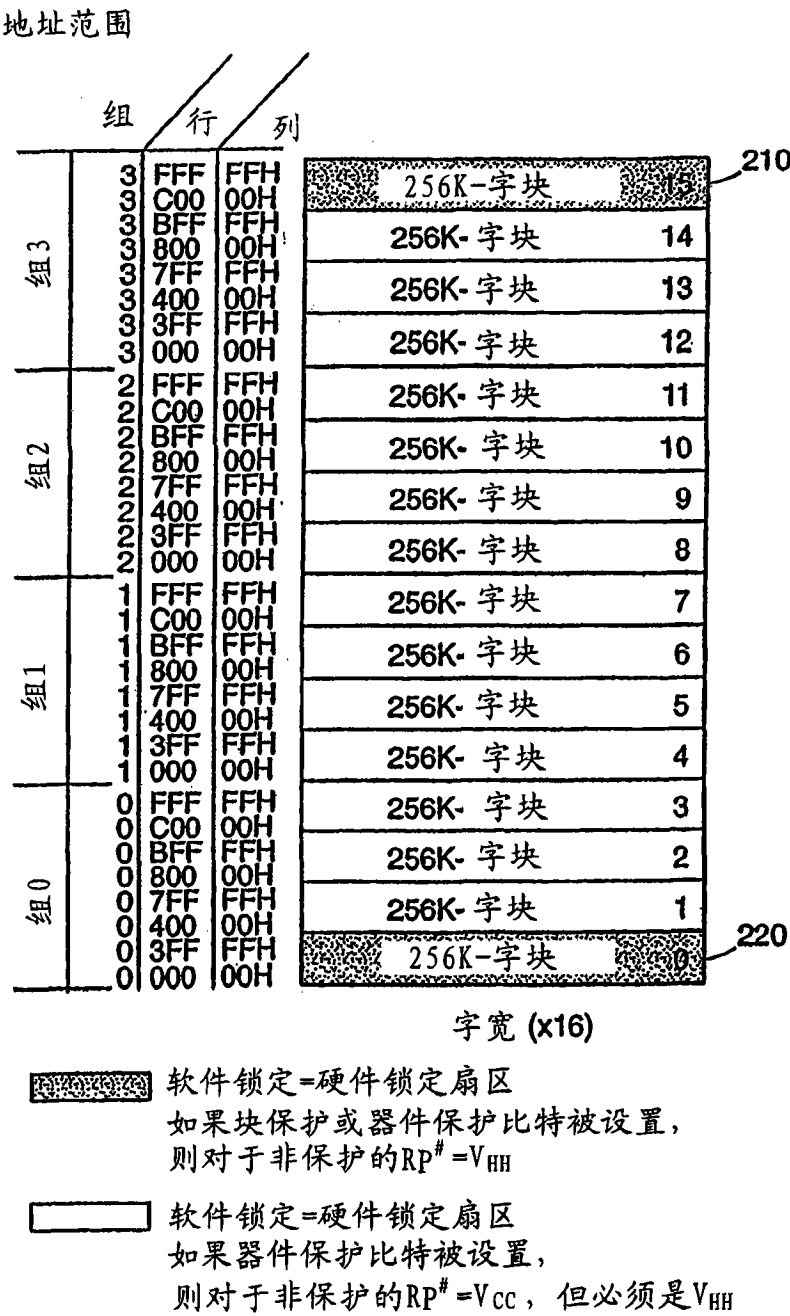


图 15

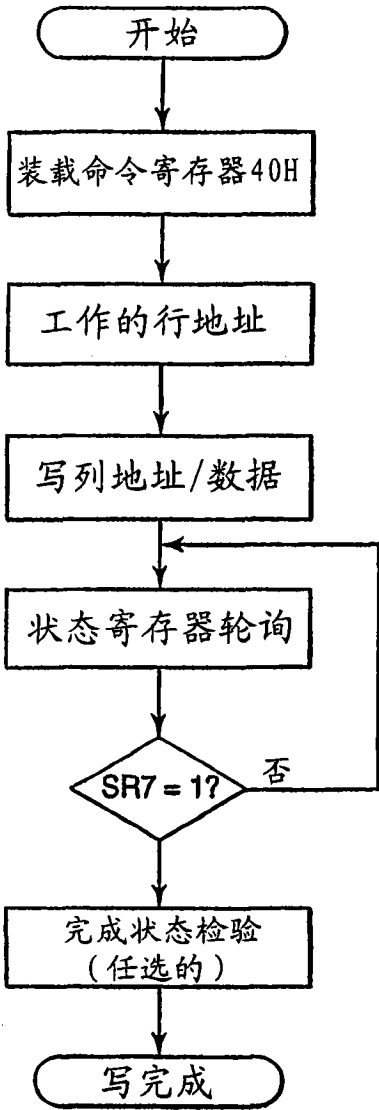


图 16

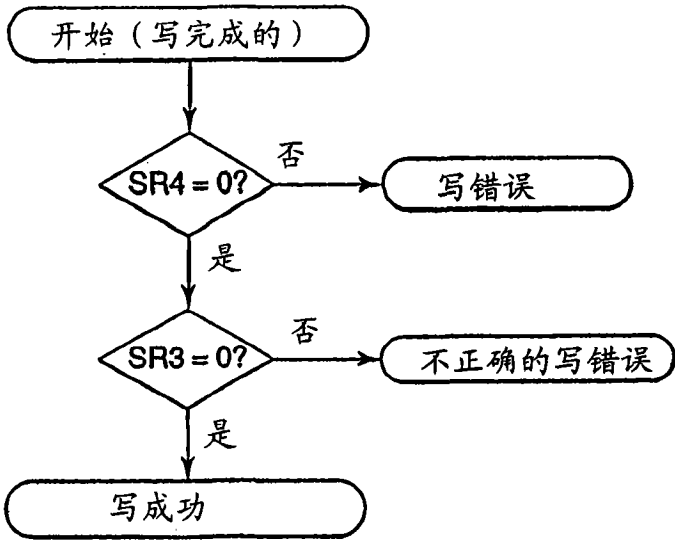


图 17

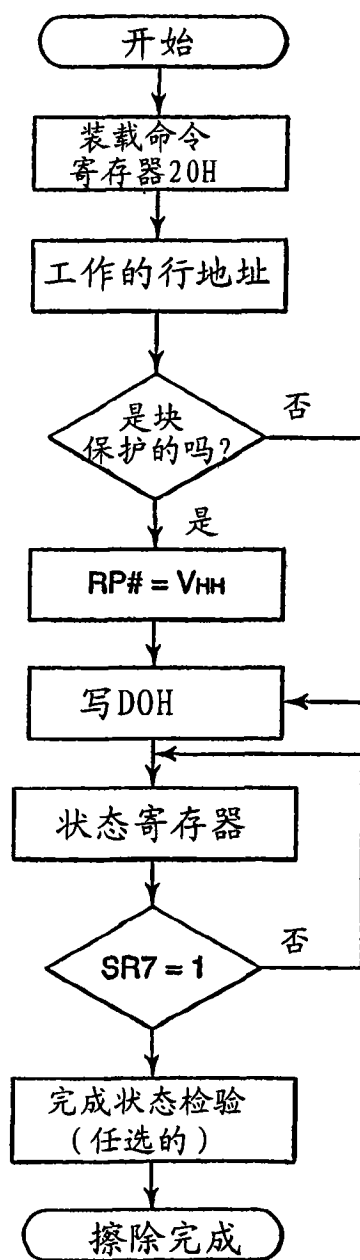


图 18

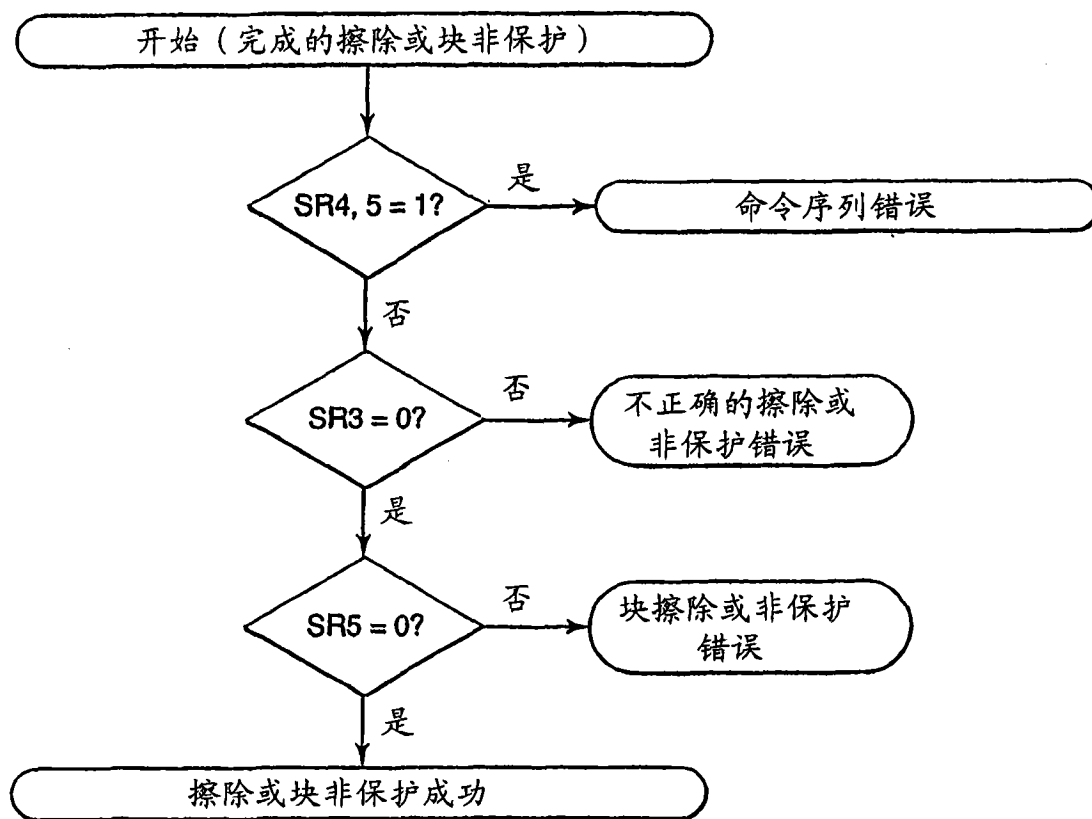


图 19

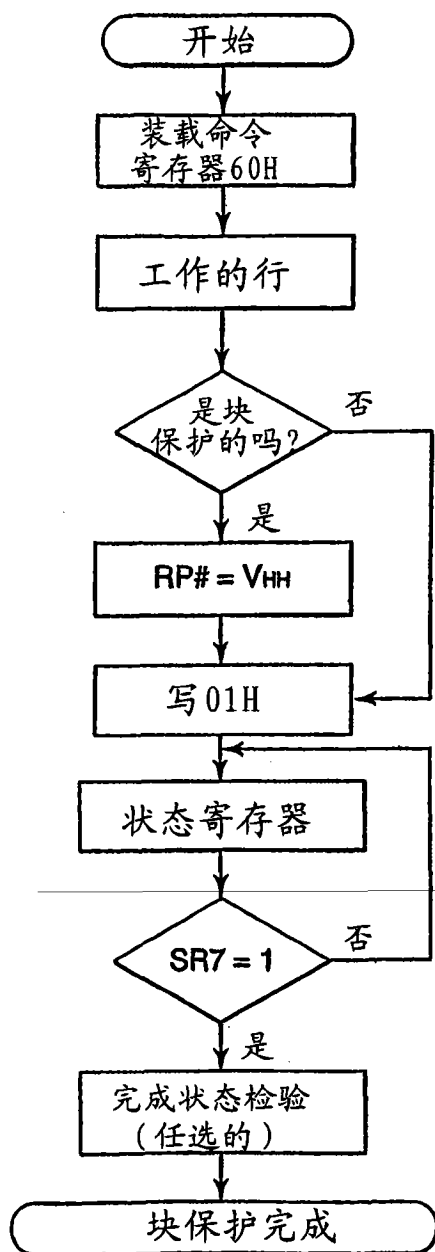


图 20

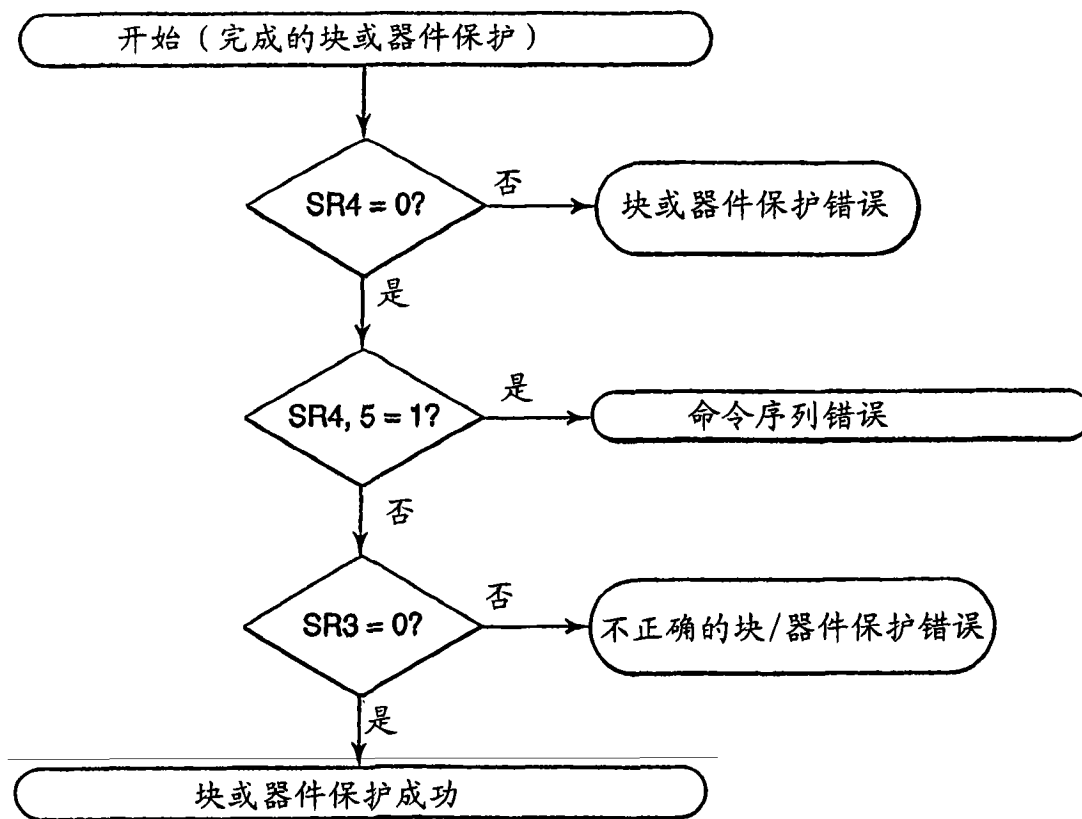


图 21

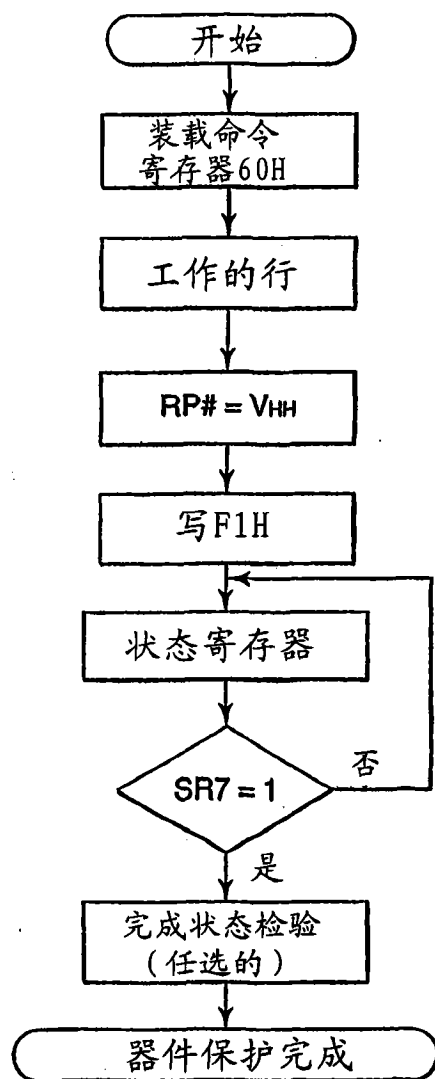


图 22

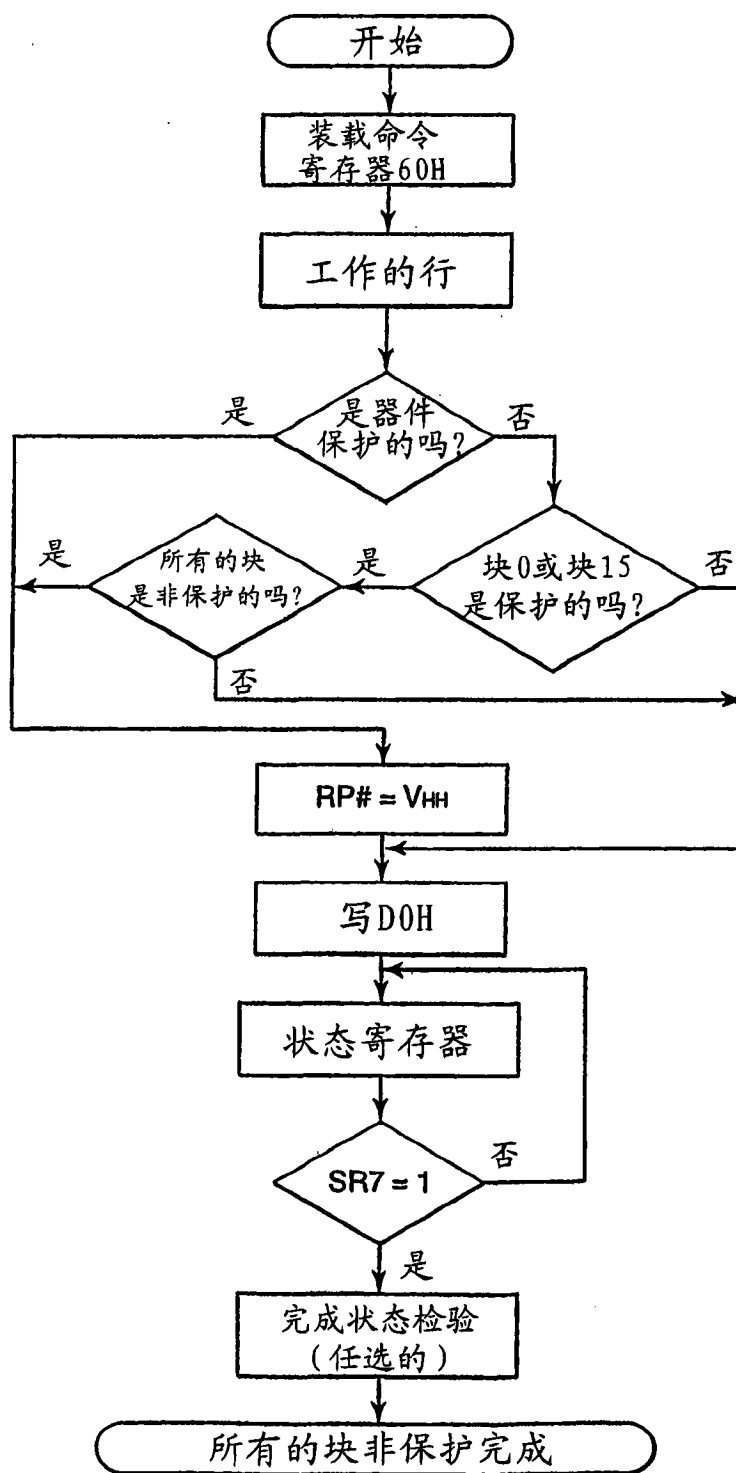


图 23

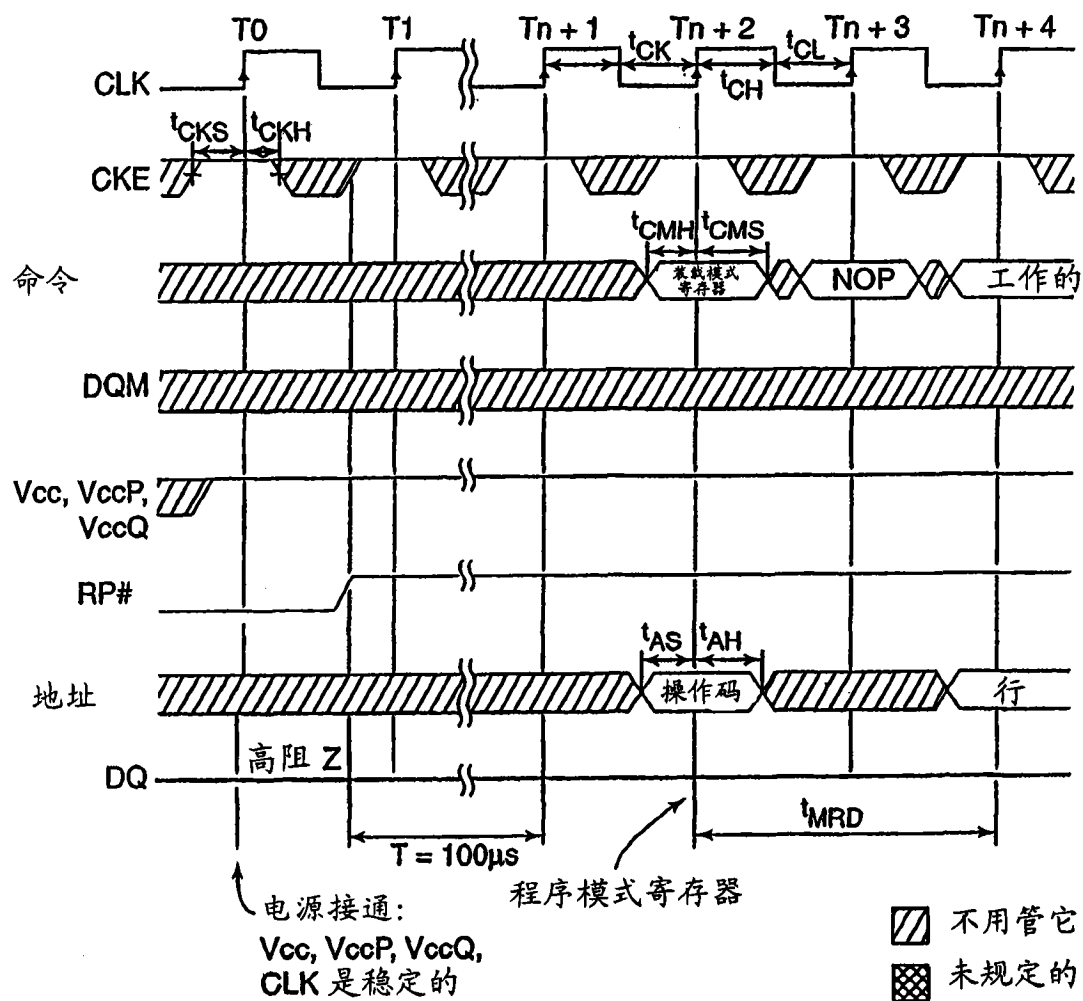


图 24

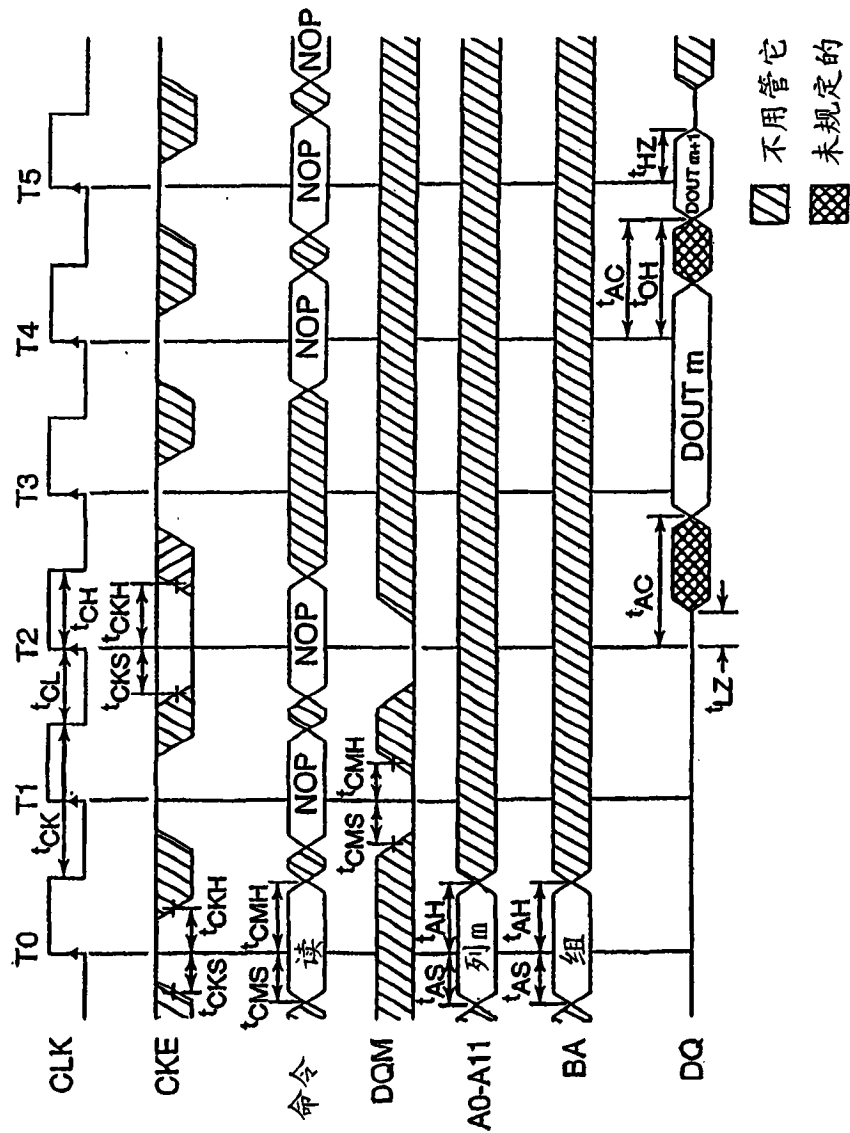


图 25

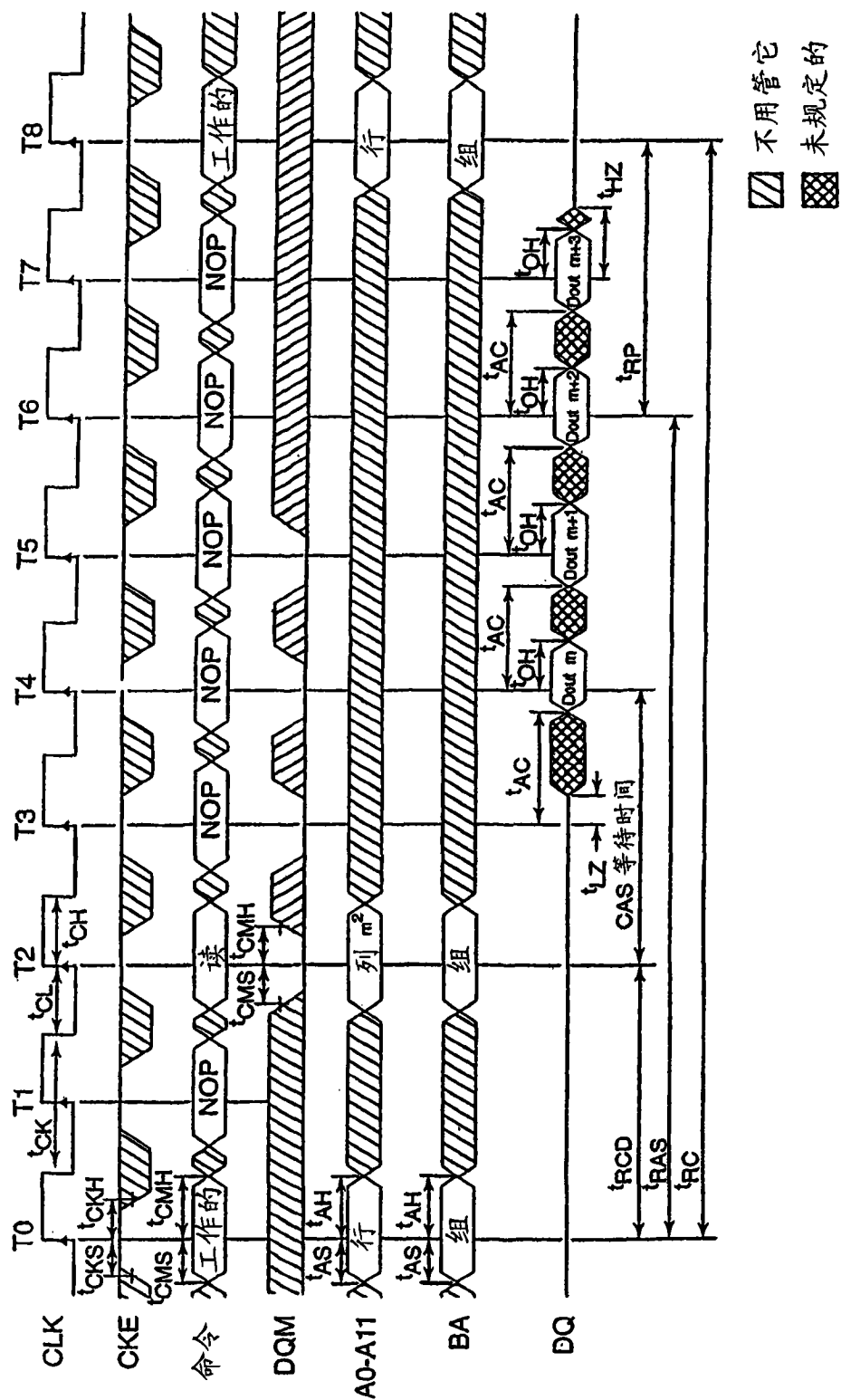


图 26

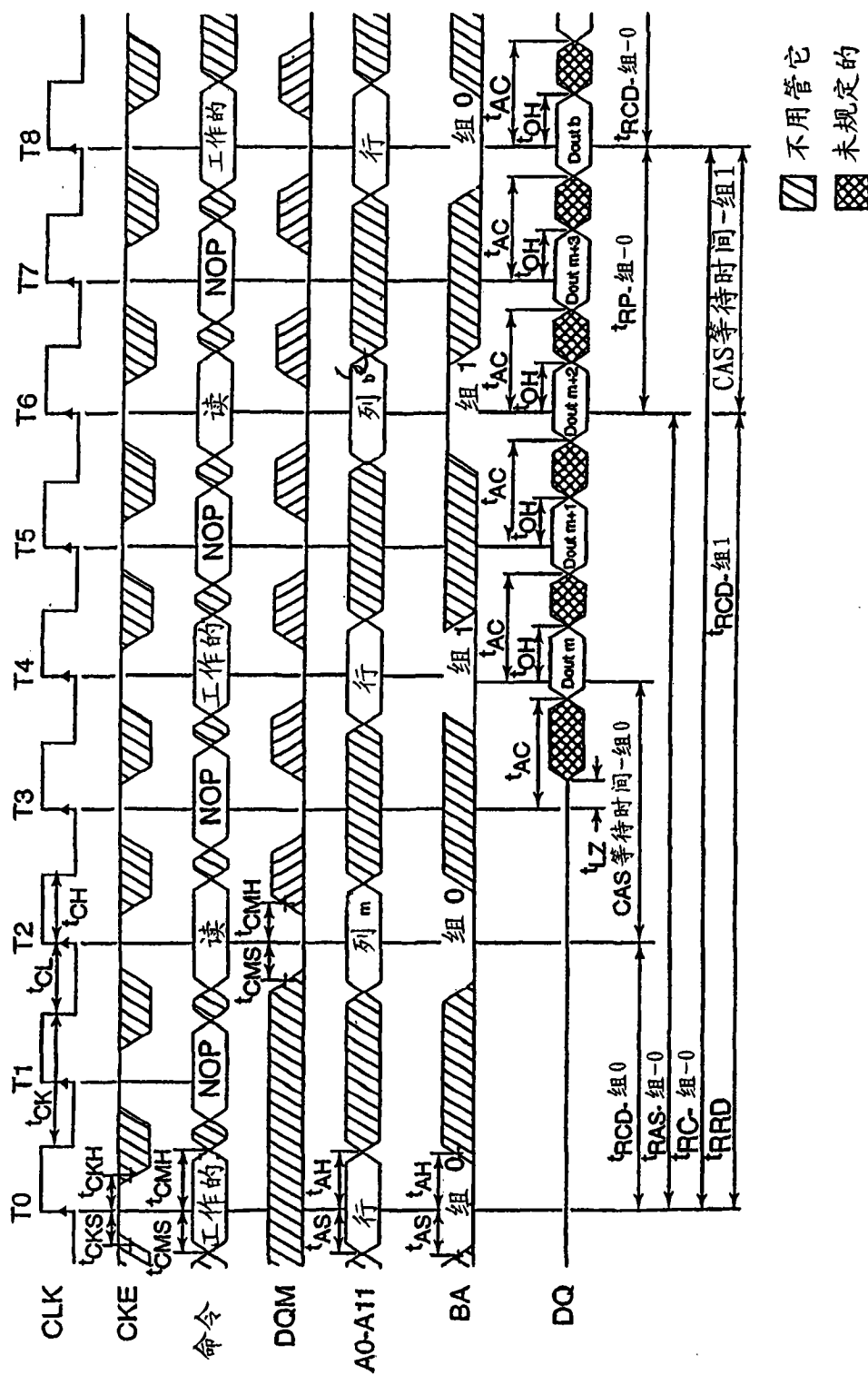


图 27

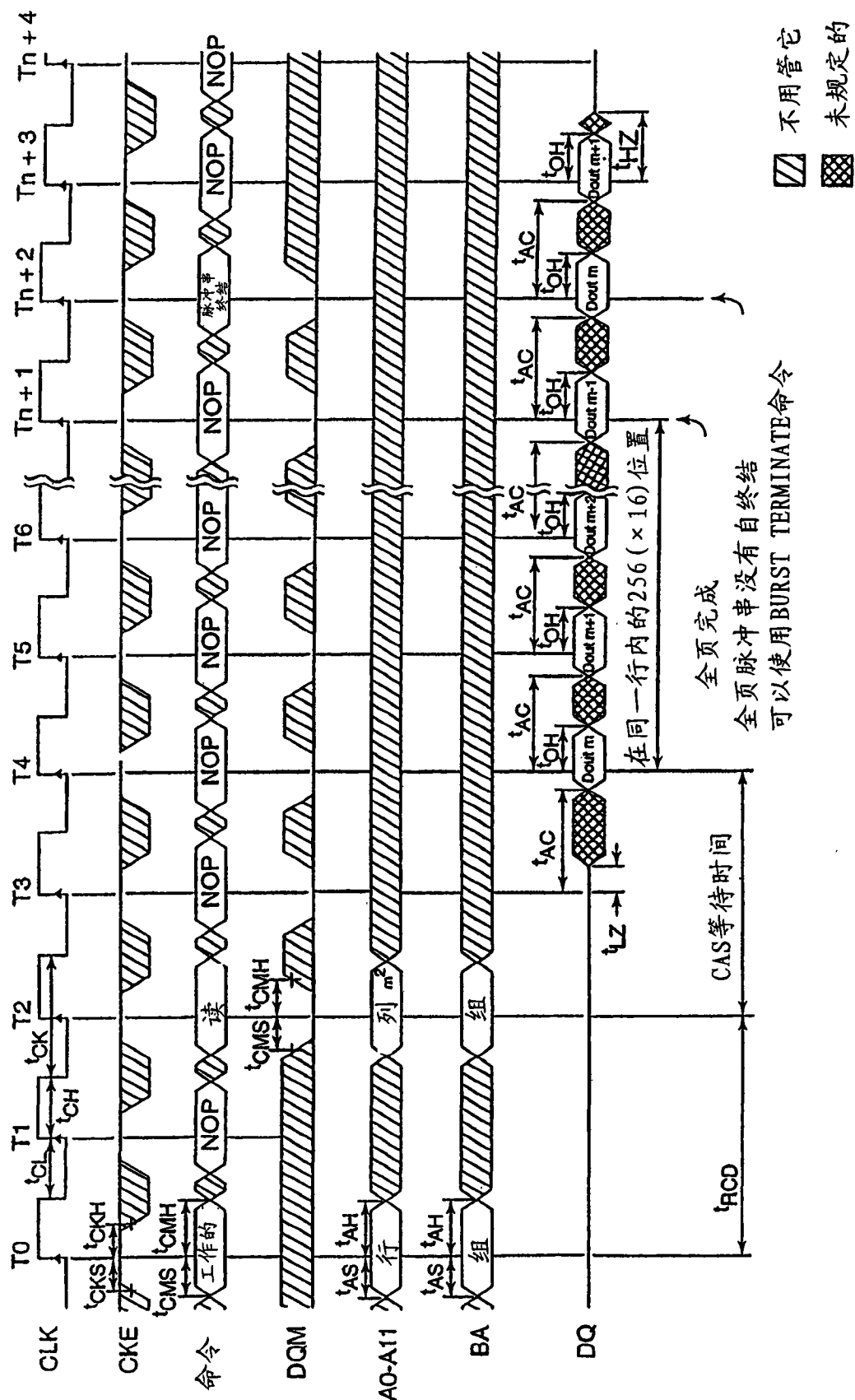


图 28

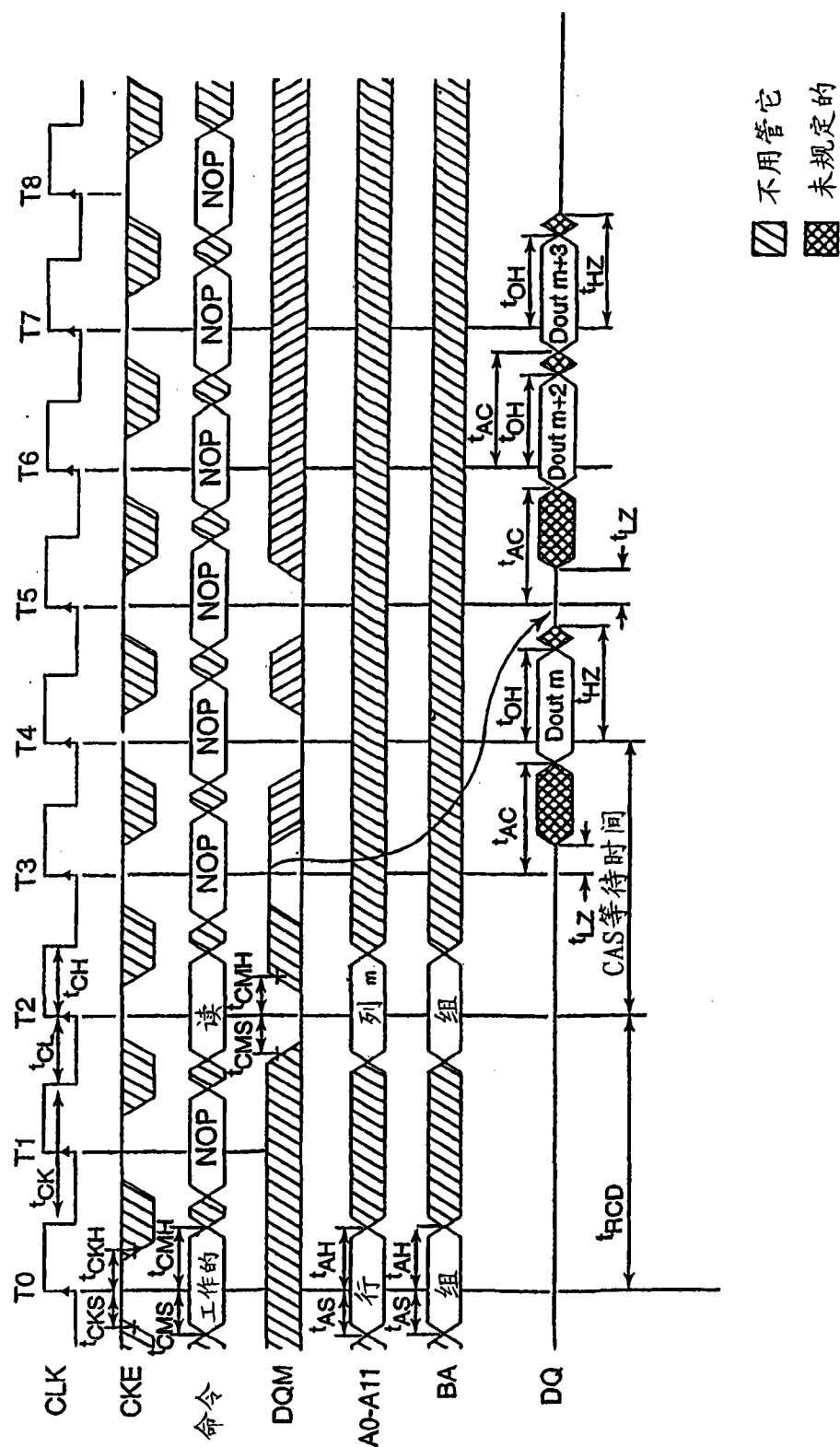


图 29

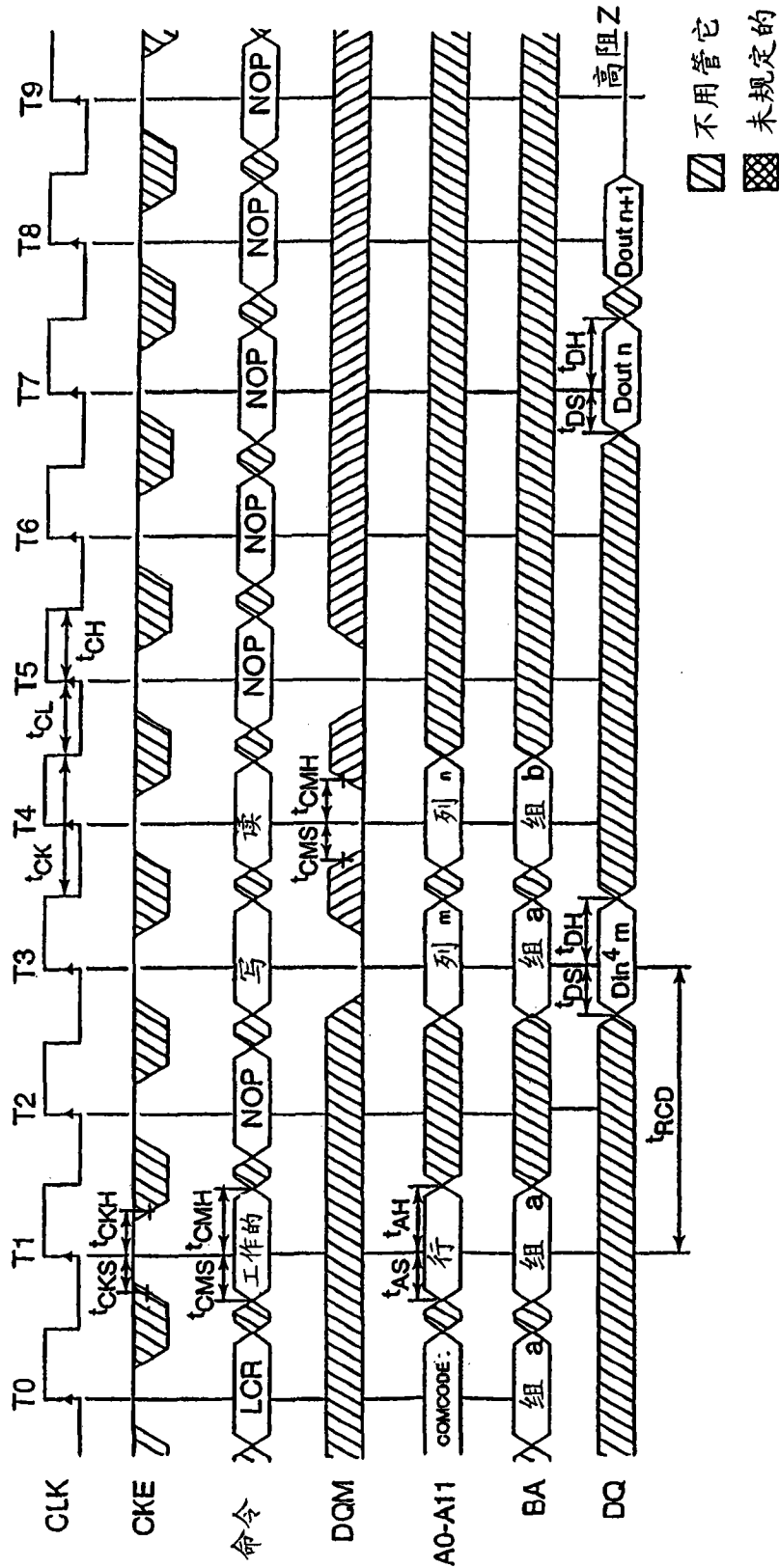


图 30

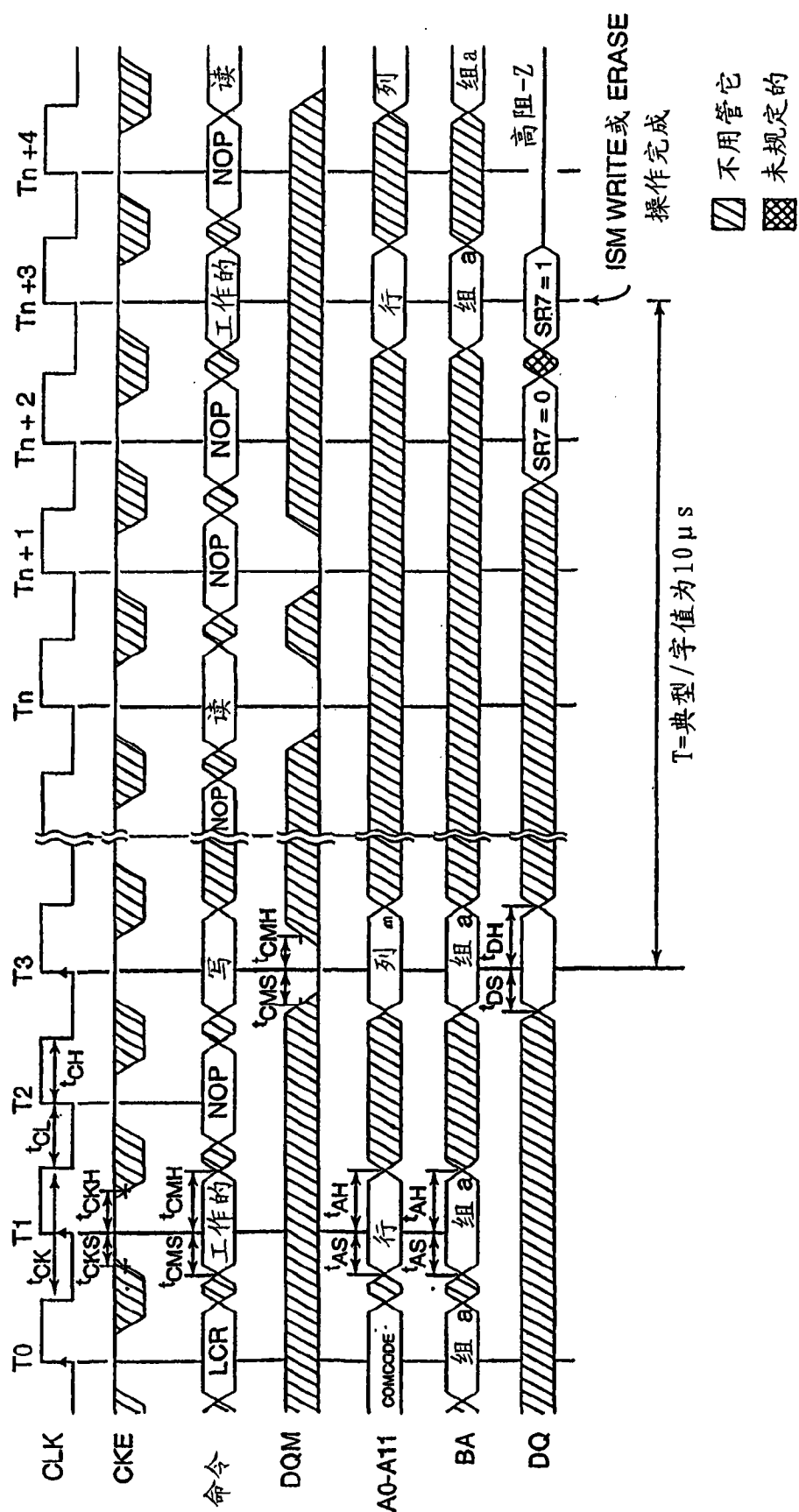


图 31

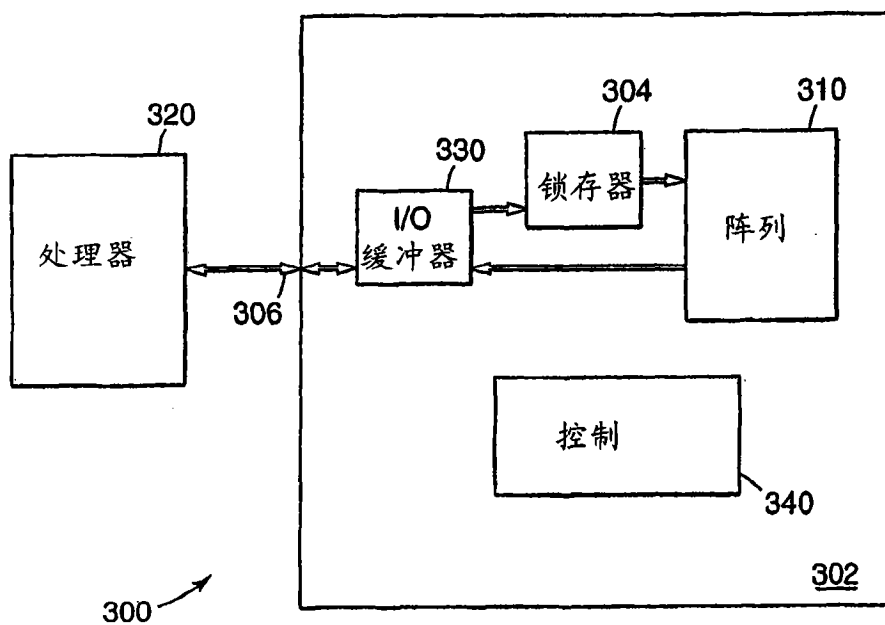


图 32

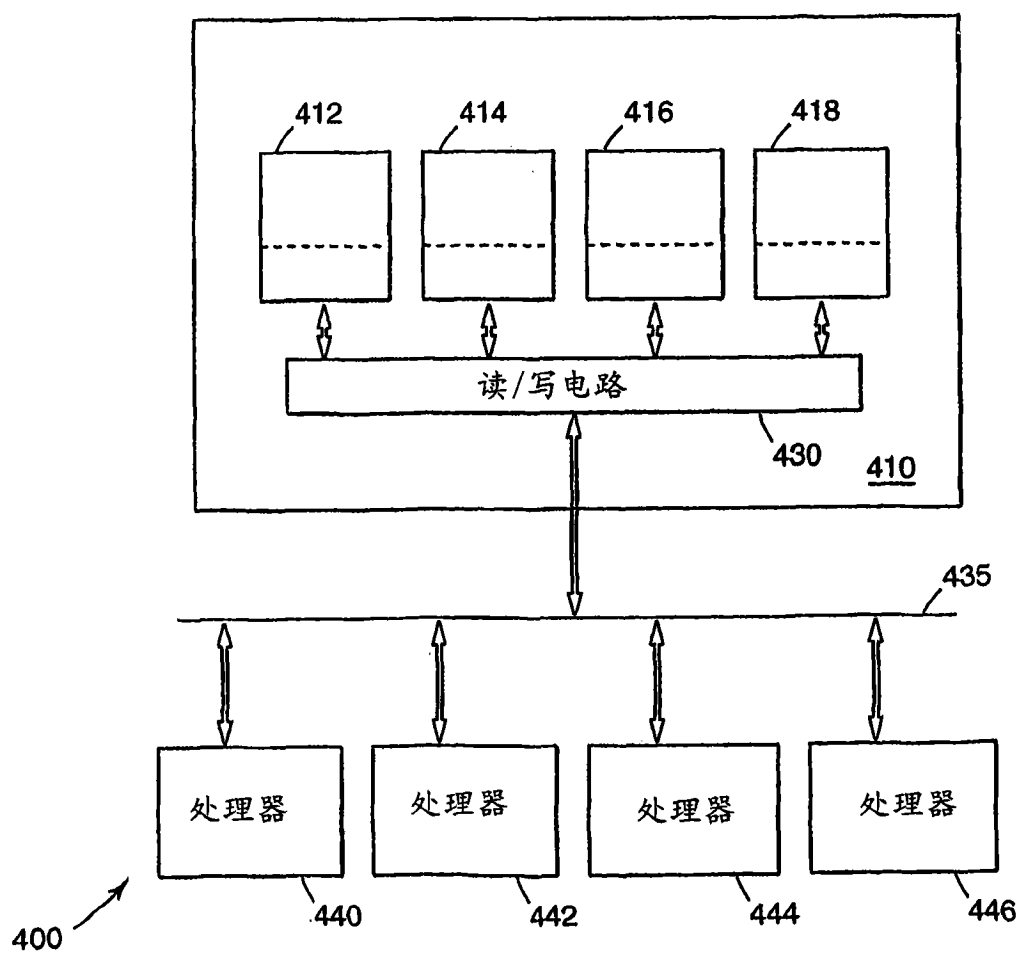


图 33