



[12] 发明专利申请公开说明书

[21]申请号 95104511.3

[51]Int.Cl⁶

G06F 13/00

[43]公开日 1996年3月13日

[22]申请日 95.3.24

[30]优先权

[32]94.3.24 [33]GB[31]9405914.4

[32]94.7.29 [33]GB[31]9415365.7

[32]94.7.29 [33]GB[31]9415391.3

[32]94.7.29 [33]GB[31]9415387.1

[32]95.2.28 [33]GB[31]9503964.0

[71]申请人 DVA公司

地址 美国加利福尼亚州

[72]发明人 安东尼·M·琼斯

唐纳德·W·W·帕特森

威廉·P·罗宾斯

阿德里安·P·怀斯 海伦·R·芬奇

马丁·W·萨瑟安

[74]专利代理机构 柳沈知识产权律师事务所

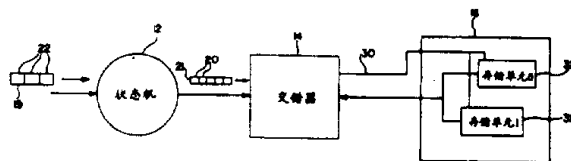
代理人 孙履平

权利要求书 8 页 说明书 41 页 附图页数 19 页

[54]发明名称 与RAM接口的方法和设备

[57]摘要

为寻址和存取存储器的方法，包括从 RAM 存取小于 RAM 的预定固定二进制位组长度 N 的 M 个字。和对 DRAM 存取方法，提供具有固定宽度的字的过程，该固定宽度字具有用来寻址可变宽度数据的固定位数，并具有宽度定义字段和地址字段。对被组织为帧或场的编码视频数据的缓冲进行控制。包括确定每一输入译码帧的图像编号、确定任何时候期待的扫描编号、以及当其图像编号被显示或在扫描编号上或其后将任何缓冲器标记为准备就绪等。



权 利 要 求 书

1、一种从RAM存取小于RAM的预定固定二进制位组长度N的M个字的方法，RAM包括可选择地允许和禁止RAM的读和写的允许线，其特征在于：

对N个字排序，以使其从RAM读出或写入RAM；

确定何时已读出M个字或已将M个字写入RAM，M小于N；以及

一旦确定已从RAM读出M个字或已将M个字写入RAM，就禁止RAM。

2、一种从RAM读出小于RAM的预定固定二进制位组长度N的M个字的方法，RAM包括允许线，其可选择地允许和禁止从RAM的读出，其特征在于：

对N个字排序，以使其从RAM被读出；

确定何时已从RAM读出M个字，M小于N；以及

一旦确定已从RAM读出M个字，就禁止RAM。

3、一种将小于RAM的预定固定二进制位组长度N的M个字写入RAM的方法，RAM包括允许线，其可选择地允许和禁止写入RAM，其特征在于：

对N个字排序，以使其写入RAM；

确定何时M个字已写入RAM，M小于N；以及

一旦确定M个字已写入RAM，就禁止RAM。

4、一种存取动态随机存取存储器的方法，用于存储和检索与两维图象相关的数据字，DRAM包括两个单独的存储单元，每一存储单元能够在页面模式下运行，读出和写入数据字，两维图象按区域

的两维网格图形来组织，每一区域包括 $M \times N$ 的象素矩阵和与每一区域相关的字，其占据一个页面或占据不足一个存储单元，其特征在于：

(a) 给每一区域指定两个存储单元中特定的一个单元，以使其从特定存储单元的特定页面读出与该区域相关的全部数据字和将与该区域相关的全部数据字写入特定存储单元的特定页面，这样给区域指定存储单元，使得与每个区域相关的存储单元不同于与也在同一行或同一列的任何相邻区域相关的存储单元；

(b) 读出与由象素矩阵组成的区域相关的数据字，该区域不与两维网格图形对齐，而与在两维网格图形的区域中的象素对齐。

5、如权利要求4的方法，所述DRAM含有第一和第二存储单元，并含有读出与未对齐区域相关的数据字的步骤(b)，其特征不在于；包括以下步骤：

(c) 确定在两维网格图形中的哪一个区域包括与未对齐区域相关的数据字；

(d) 从DRAM的第一存储单元读出与网格图形中的一个区域相关的数据字，该网格图形被确定包括了与未对齐区域相关的数据字；

(e) 从DRAM的第二存储单元读出与网格图形中的另一个区域相关的数据字；该网格图形被确定包括了与未对齐区域相关的数据字；

(f) 重复步骤(d)和(e)，直到与未对齐区域相关的全部数据字都已被读出为止。

6、一种存取动态随机存取存储器的方法，用于存储和检索与两维图象相关的数据字，DRAM包括两个单独的存储单元，每一存储单元能够在页面模式下运行，读出和写入数据字，两维图象按区域

的两维网格图形来组织，每一区域包括 $M \times N$ 的象素矩阵，与每一区域相关的字占据一个页面或占据不足一个存储单元，其特征在于：

(a) 给每一区域指定两个存储单元中特定的一个单元，以使其从特定存储单元的特定页面读出与该区域相关的全部数据字和将与该区域相关的全部数据字写入特定存储单元的特定页面，这样给区域指定存储单元，使得与每个区域相关的存储单元不同于与也在同一行或同一列的任何相邻区域相关的存储单元；

(b) 读出与由 $M \times N$ 的象素矩阵组成的区域相关的数据字，该区域不与两维网格图形对齐，而与在两维网格图形的区域中的象素对齐。

7、如权利要求6的方法，所述DRAM含有第一和第二存储单元并含有读出与未对齐区域相关的数据字的步骤(b)，其特征在於；包括以下步骤：

(c) 从DRAM的第一存储单元读出与网格图形中的一个区域相关的数据字，该网格图形含有与未对齐区域相关的数据字；

(d) 从DRAM的第二存储单元读出与网格图形中的另一个区域相关的数据字，该网格图形含有与未对齐区域相关的数据字；

(e) 重复步骤(c)和(d)，直到与未对齐区域相关的全部数据字都已被读出为止。

8、如权利要求6的方法，所述DRAM含有第一和第二存储单元，并含有读出与未对齐区域相关的数据字的步骤(b)，其特征在於；包括以下步骤：

(c) 以预定的区域次序读出与网格图形中的每一区域相关的数据字，该网格图形含有与未对齐区域相关的数据字，该预定的区域

次序产生从另外的存储单元读出的随后的区域的数据字读出。

9、如权利要求8的方法，其特征在于：所述预定的次序是在包括与未对齐区域相关的数据字的网格图形中的区域的顺时针方向。

10、如权利要求8的方法，其特征在于：所述预定的次序是在包括与未对齐区域相关的数据字的网格图形中的区域的逆时针方向。

11、如权利要求6的方法，所述DRAM含有第一和第二存储单元，并含有读出与未对齐区域相关的数据字的步骤(b)，其特征在于；包括以下步骤：

(c) 确定在两维网格图形中的哪一个区域含有与未对齐区域相关的数据字；

(d) 从DRAM的第一存储单元读出与网格图形中的一个区域相关的数据字，该网格图形被确定为含有与未对齐区域相关的数据字；

(e) 从DRAM的第二存储单元读出与网格图形中的另一个区域相关的数据字，该网格图形被确定为含有与未对齐区域相关的数据字。

(f) 重复步骤(d)和(e)，直到与未对齐区域相关的全部数据字都已被读出为止。

12、如权利要求6的方法，所述DRAM含有第一和第二存储单元，并含有读出与未对齐区域相关的数据字的步骤(b)，其特征在于；包括以下步骤：

(c) 确定在两维网格图形中的哪一个区域含有与未对齐区域相关的数据字；

(d) 以预定的区域次序读出与网格图形中的每一区域相关的数据字，该网格图形被确定为含有与未对齐区域相关的数据字；该预定的区域次序产生从另外的存储单元读出的随后的区域的数据字读

出。

13、如权利要求12的方法，其特征在于：所述预定的次序是在被确定为含有与未对齐区域相关的数据字的网格图形中的区域的顺时针方向。

14、如权利要求12的方法，其特征在于：所述预定的次序是在被确定为含有与未对齐区域相关的数据字的网格图形中的区域的逆时针方向。

15、一种对存储器进行寻址的方法，其特征在于：

提供一固定宽度字，其具有用来对可变宽度数据进行寻址的预定固定位数；

定义具有宽度定义字段和地址字段的固定宽度字；

给宽度定义字段提供至少一个位来作为终止标记；

定义地址字段，该字段具有用来定义数据地址的多个位；

与可变宽度数据的大小成反比地改变在地址字段中的位的大小；

与可变宽度数据的大小成正比地改变在宽度定义字段中的位数；

在改变宽度定义字段和地址字段的宽度时保持对可变宽度数据进行寻址的固定宽度字。

16、一种对存储器进行寻址的方法，其特征在于：

提供一固定宽度字，其具有用来对数据进行寻址的预定固定位数；

定义具有地址字段和替换字段的固定宽度字；

定义具有用来定义数据地址的多个位的地址字段；

定义一个具有至少一个替换位的可变宽度替换字段；

该替换字段具有至少一个位作为地址字段和替换字段之间的终

止标记;

用替换字段来表示来自单独的寻址源的替换位; 以及在反向地改变地址字段的宽度和替换字段的宽度时保持对可变宽度数据进行寻址的固定宽度字。

17、一种对存储器中的可变宽度数据进行寻址的方法, 其特征在于:

提供具有预定宽度和由部分字组成的字的存储器;

将要被存取的部分字循环移位成为与最低有效位对齐;

扩展该字的余下部分, 使得被存取的字将被识别为部分字;

恢复该字的余下部分;

循环移位该字, 直到该部分字被恢复到其原来的位置。

18、一种对组织成为帧排列的编码视频数据的缓冲进行控制的方法, 其特征在于:

确定帧的图象编号;

确定帧的所需扫描编号; 以及

当图象编号被显示在所需的扫描编号上或其后时, 将缓冲器标记为准备就绪。

19、一种连接总线和RAM的RAM接口, 其特征在于: 包括

从总线接收多个数据字并缓冲接收的数据字的装置;

从总线接收与多个数据字相关的地址的装置;

在RAM中产生一系列将要被写入已缓冲的数据字的地址的装置, 该地址序列从接收的地址获得; 以及

根据产生的地址将被缓冲的数据字写入RAM的装置。

20、如权利要求19的RAM接口, 其特征在于: 数据字接收和缓

冲装置包括摆动缓冲器。

21、如权利要求19的RAM接口，其特征在于：

RAM按页面寻址方式运行；以及

地址产生装置包括根据接收的地址产生行地址的装置和产生列地址的装置。

22、如权利要求21的RAM接口，其特征在于：

RAM是DRAM；

总线包括两线接口；

数据字接收和缓冲装置包括两线接口；

地址接装置包括两线接口；

多个数据字为记号的形式；以及

接收的地址为记号的形式。

23、如权利要求19的RAM接口，其特征在于：还包括确定数据字接收装置是否已接收和缓冲了多个数据字的装置。

24、一种连接总线和RAM的RAM接口，其特征在于：包括：

按照预定地址存储在RAM中的多个数据字；

从总线接收与多个数据字相关的地址的装置；

产生一系列RAM地址的装置以使其对在RAM中的多个数据字进行寻址，该地址序列从接收的地址获得；

缓冲从RAM读出的数据字的装置；以及

利用由地址产生装置产生的RAM地址序列从RAM读出多个数据字，并将数据字写入缓冲器装置的装置。

25、如权利要求24的RAM接口，其特征在于：所述数据字缓冲装置包括摆动减缓器。

26、如权利要求24的RAM接口，其特征在于：
所述RAM按页面寻址方式运行；以及
地址产生装置包括根据接收的地址产生行地址的装置和产生列地址的装置。

27、如权利要求26的RAM接口，其特征在于：
所述RAM是DRAM；
总线包括两线接口；
数据字缓冲装置包括两线接口；
地址接收装置包括两线接口；以及
接收的地址为记号的形式。

28、如权利要求24的RAM接口，其特征在于：还包括确定数据字接收装置是否已接收和缓冲了多个数据字的装置。

说明书

与RAM接口的方法和设备

本发明涉及随机存取存储器 (RAM)，特别涉及与RAM接口的方法和设备。

本发明目的是提供一种新的与RAM接口的方法和设备，本发明是针对现有技术中的方法和设备作出的进一步的改进。

本发明提供的存取RAM的方法，特别是，从RAM存取小于RAM的预定固定二进制位组长度N的M个字数的方法，RAM包括一允许线可选择地允许和禁止对RAM的读出和写入。在该方法中，对前N个字排序，以使其从RAM读出或写入RAM。然后确定何时从RAM已读出M个字或已将M个字写入RAM，M小于N。最后，一旦确定已从RAM读出M个字或已将M个字写入RAM，就禁止RAM。

本发明提供的另一方法是对动态随机存取存储器 (DRAM) 进行存取的方法，以使其存储和检索与两维图象相关的数据字。DRAM包括两个单独的存储单元，第一存储单元和第二存储单元。每一存储单元能够在页面模式下运行，使其读出和写入数据字。两维图象按区域的两维网格图形来组织，每一区域包括M×N的象素矩阵。与每一区域相关的字占据一个页面或占据不足一个存储单元。为每一区域指定两个存储单元中的一个存储单元，以使其从该存储单元的某一页面读出与该区域相关的全部数据字和将与该区域相关的全部数据字写入该存储单元的某一页面。这样来给多个区域指定存储单元，使

得每个区域与任何相邻区域的不同存储单元相关；相邻区域指在同一行或同一列的区域。然后读出与一区域相关的数据字，该区域由象素矩阵组成，它不与两维网格图形对齐，而与在两维网格图形的区域中的象素对齐。

根据本发明的方法，通过首次读出，从DRAM的第一存储单元读出与未对齐区域相关的数据字，该数据字与网格图形中的一个区域有关，该网格图形被认为含有与未对齐区域相关的数据字。然后从DRAM的第二存储单元读出与网格图形中另一区域有关的数据字，该网格图形含有与未对齐区域相关的数据字。继续第一和第二存储单元之间的交替读出，直到与未对齐区域相关的全部数据字都被读出为止。

根据本发明的方法，通过首先读出，按预定的区域次序读出与未对齐区域相关的数据字，其与网格图形中的每一区域相关，网格图形含有与未对齐区域相关的数据字。这样选择预定的区域次序，使得从随后的区域的数据字读出是从另外的存储单元被读出的。

根据本发明的方法，预定次序是在网格图形中的区域的顺时针方向，其被确认为包括与未对齐区域相关的数据字。或者，预定次序是网格图形中的区域的逆时针方向。其被确认为包括与未对齐区域相关的数据字。

本发明涉及对存储器进行存取的方法和设备，特别涉及使用固定宽度字，它具有用来寻址可变宽度数据的固定数目的位和地址替换。

本发明提供了对被组成为帧的编码视频数据进行缓冲的方法，该方法包括以下步骤：确定帧的图象编号、确定帧的所需扫描编号

(presentation number) 以及当图象编号处在所需的扫描编号上或者在其之后时将缓冲器标记为准备就绪。

根据本发明的目的，本发明提供了连接总线和RAM的RAM接口，该RAM接口包括：从总线接收多个数据字的装置，从总线接收与该多个数据字相关的地址的装置，在RAM中产生一系列将要被写入被缓冲的数据字的地址（该地址序列从接收的地址获得）的装置，以及根据产生的地址将被缓冲的数据字写入RAM的装置。数据字接收和缓冲装置可以包括摆动减缓器。RAM能以页面寻址模式运行，地址产生装置可以包括根据接收的地址产生行地址的装置和产生列地址的装置。RAM可以是DRAM，总线可以包括两线接口 (two-wire interface)，数据字接收和缓冲装置可以包括两线接口，地址接收装置可以包括两线接口，多个数据字以及接收地址可以是记号 (Token) 的形式。RAM接口可以进一步包括一些装置以确定数据字接收装置是否已接收了并已缓冲了多个数据字。

本发明提供了连接总线和RAM的RAM接口，该RAM接口包括：按照预定地址存储在RAM中的多个数据字，从总线接收与在RAM中的多个数据字相关的RAM地址的装置，产生一系列RAM地址以便对在RAM中的多个数据字进行寻址的装置，该地址序列从接收地址获得，缓冲从RAM读出的数据字的装置，以及利用由地址产生装置产生的RAM地址序列从RAM读出多个数据字、并将数据字写入缓冲器装置的装置。数据字缓冲装置可以包括摆动减缓器。RAM可以工作于页面寻址模式，地址产生装置可以包括根据接收地址产生行地址的装置和产生列地址的装置。RAM可以是DRAM，总线可以包括两线接口，数据字接收和缓冲装置可以包括两线接口，地址接收装置可以包括两

线接口，多个数据字以及接收地址可以是记号(Token)的形式。RAM接口可以进一步包括确定数据字接收装置是否已接收并已缓冲了多个数据字的装置。

本发明的方法和设备将以实施例结合附图进行详细说明。

附图简要说明

图1是将本发明的方法用于存取同步DRAM的设备的系统方框图；

图2是由图1状态机接收的高级命令和由图1的状态机产生的低级命令之间的相互关系示意图；

图3是图1的DRAM的组织方式的图形表示；

图4是存储在图1的DRAM的单元中的字的组织方式的图形表示；

图5是显示在电视或监视器屏幕上的图象示意图，由在矩形网格中对齐的区域组成，一个未对齐的区域叠加在对齐的区域上；

图6表示在图5的每一区域内象素的排列，每一区域为示范性的8象素×8象素块；

图7表示图6每一象素的数据字；

图8表示未对齐的区域和位于下面的对齐区域中的象素之间的关系；

图9是将图5或图8的区域影射到图6的RAM存储单元的示意图；

图10是具有两个RAM的存储单元的RAM的系统的方框图，用于存储图7的数据字；

图11是视频译码器的图形表示，该视频译码器包括图10的RAM系统，给图5的屏幕提供数字视频；

图12表示图5或图8的区域被进一步划分为子区域，以及子区域

和图10 RAM存储单元之间的关系；

图13是微码状态机的方框图；

图14是“算术中心部分”的方框图；

图15是“寄存器文件”的功能图；

图16是在“寄存器文件”中的数据流的方框图；

图17是寄存器文件地址译码的方框图；

图18是用来寻址的固定宽度字，具有宽度定义字段和地址字段；

图19是用来寻址的固定宽度字，具有地址字段、替换字段和替换指示符；

图20是13位字的实例，用于在 64×32 RAM中寻址8位数据；

图21是具有字段的固定宽度字的实例；

图22是实施本发明的DRAM接口的方框图；

图23是具有一个写摆动减缓器和一个读摆动减缓器的DRAM接口的方框图；

图24是图象格式器的方框图；

图25是缓冲器管理状态机的图示；

图26表示图25状态机的主要环路。

根据本发明的存取同步DRAM的方法请参看图1，该图表示实施本发明存取同步DRAM的方法的系统10的方框图。系统10包括状态机12、交错器14和同步DRAM 16。DRAM 16包括两个存储单元32。交错器14控制对DRAM 16的存取（例如读和写）。交错器14包括计数器（未示出），该计数器监测哪一个存储单元最近被存取，使交错器14容易地在存储单元32之间反复。

现在参看图1、图2和图3，一旦接收到对DRAM 16的同一存储单

元32进行寻址的高级命令22的字符串19，状态机12就产生对同一存储单元32进行寻址的最优低级命令20（即图上列出的由JEDEC规定的7个命令）的字符串21。一旦接收到字符串21，交错器14就执行被请求的动作。

高级命令22是READ（读）22-1、WRITE（写）22-2、EXTEND（扩展）22-3、INTERRUPT（中断）22-4和REFRESH（刷新）22-5。（命令22全部字母大写以便容易地将它们与低级命令20区别开来）。简单地说，READ 22-1、WRITE 22-2和REFRESH 22-5命令的操作基本上与它们同名的低级命令20相同。EXTEND 22-3重复先前WRITE 22-2或READ 22-1命令的动作。如以下进一步讨论的那样，由于二进制位组长度（burst length）被固定为四个字24，所以需要INTERRUPT 22-4。总计，命令22提供了强有力的编程工具。

在讨论命令22如何简化存取DRAM 16的任务之前，应当更详细地考察命令22本身。高级命令字符串19的第一个高级命令22是READ 22-1或WRITE 22-2。相应低级命令字符串21的第一个低级命令20总是现用（active）命令20。每一个READ或WRITE命令分别要求读出或写入四个顺序字24。如图2所示，每一个READ或WRITE命令22都跟着四个顺序字24第一个字24的地址26（在被存取的特定存储单元32中）。（注意在JEDEC标准中，该第一个字24必须是偶计数）。

当DRAM 16处于“自动预充电模式”时，如果READ或WRITE命令22是字符串19中的最后的命令22（即如果到达状态机12的下一个命令是对DRAM 16的其它存储单元32进行寻址），则状态机12就将发出预充电读或写命令20，随后为恰当的四顺序字24的必要地址26。否则，状态机12就将产生读或写命令20，随后为必要地址26。例如，

如果在自动预充电模式中，在字符串19-2中的最后的命令22是具有10的地址26的READ 22-1，则相应的字符串21-2就将是预充电读命令20，随后为10、11、12和13的地址26。

在图2中，EXTEND 22-3位于字符串19-3的末尾，跟在READ (WRITE)命令22之后。类似于READ和WRITE命令22，EXTEND 22-3的格式包括在四个字24序列中的第一个字24的地址26中。在自动预充电模式中，当EXTEND 22-3出现在字符串19的末尾时，如在字符串19-3中，则给字符串21增加附加的预充电读（写）。

本发明将二进制位组长度固定为四个字24，减少了支持动态地改变二进制位组长度所需的复杂性。为了处理JEDEC标准所需的两个字的传送，提供了INTERRUPT 22-4。现在参看图1、图2和图3，INTERRUPT 22-4使状态机12开始四个字24长低级命令20的读（或写）。但是，在已读出（或写入）两个字24之后，状态机12使存储单元32的允许线30变为低电平，从而禁止存储单元32。这样一来，尽管状态机12发出四个字24读（或写）的低级命令20，但实际只从存储单元32读出（或写入到存储单元32）两个字24。

当然，在允许线30变为低电平之后，需要另外的两个时钟周期来完成现行无效的读（或写）命令20。但这一方案在将二进制位组长度固定为四个字24的时候这种处理解决了不要的数据两个字24。

现在参看图1、图2、图3和图4，图4表示字24如何被存储在存储单元32中。图4表示区域50的两维矩阵的一部分48，每一区域50包括字24的M×N矩阵，每个字用作为在区域50内的字24的地址的数字54来表示。特别是，将每一区域50表示为8行乘8列的字24。按照网格图形56来对齐区域50。

叠加在区域50上的是区域60，区域60表示由来自在下的区域50的读出字24生成的所需的新区域（存储在DRAM 16的另一部分中）。区域60与字24对齐，但不与网格图形56对齐。区域60覆盖了区域50-1、50-2、50-3和50-4。区域50-1和50-3包括存储在同一存储单元32（例如存储单元0）中的，但可能位于不同页面上的字24。同样地，区域50-2和50-4包括存储在另一存储单元32（例如存储单元1）中的，但可能位于不同页面上的字24。

为了从区域50-1读出合适的字24，应当使用以下高级命令22的字符串19：

```
READ(62, X), INTERRUPT(62, 62)。
```

相应的低级命令20的字符串21将是：

现用（存储单元0），读（62，63），读（62，63），读和预充电（62，63）。

为了从区域50-4读出合适的字24，应当使用以下高级命令22：

```
READ(6, X), INTERRUPT(14, 22)。
```

相应的低级命令20的字符串21将是：

现用（存储单元1），读（6，7），读（14，15，不考虑，不考虑），读和预充电（22，23，不考虑，不考虑）。

为了从区域50-3读合适的字24，应当使用以下高级命令22：

```
READ(0, X), EXTEND(8, X), EXTEND(16, X), EXTEND(4, X),  
INTERRUPT(12, 20)。
```

相应的低级命令20的字符串21将是：

现用（存储单元0），读（0，1，2，3），读（8，9，10，11），读（16，17，18，19），读（4，5），读（12，13），读和预充电（20，

21, 不考虑, 不考虑)。

根据本发明的存取DRAM的存储单元的方法请参看图5、图6和图11, 图11示出具有适合于显示数字视频信号204的图象208的显示屏幕206的视频监视器202。数字视频信号204的信号源是视频译码器205。视频译码器205包括合适的译码电路(未示出)。视频译码器205对编码视频信号209进行译码。编码视频信号209的惯用信号源包括CD或激光光盘播放机207, 或者包括有线电视的电路耦合(hook-up)203。

图5表示显示屏幕206的一部分, 包括图象208。如同图象208的多个图象由象素214组成。通常将象素214分为多个区域212。通过将象素214分为多个区域212, 就能够更有效地对代表图象208(以及显示屏幕206的其余部分)的数字视频信号204进行处理(例如压缩)。

虽然可以按照任何重复图形来安排区域212, 但通常按照直线网格209的图形来安排区域212。网格209的图形贯穿显示屏幕206。在每一区域212内, 象素214通常按照N行乘N列的方矩阵来排列。例如, 在图6中, 区域212由8行×8列的象素214组成。与每一象素相关的是识别位置数字213(从0至63)。或者可以将象素214分为非方形矩阵(即M行×N列, 这里M不等于N)。

现在参看图5、图6和图7, 数字视频信号204包括一些数据字215。在MPEG的数字视频标准中, 需要6个数据字215来表示4个象素214的每一个范围: 一个字215表示 C_b 、一个字215表示 C_r , 四个字215表示Y(亮度)。

现在参看图5、图7、图10和图11, 视频译码器205包括RAM系统230。RAM系统230是视频译码器205用来存储数据字215的存储器。

在产生、显示和处理屏幕206上的图象208的过程中，视频译码器205从RAM系统230读出字215。RAM系统230包括交错器234和RAM的两个存储单元，存储单元0 232-0和存储单元1 232-1。交错器234将存储单元232与视频译码器205中的被用来产生、显示和处理图象208的部分（未示出）连接。数据字215存储在存储单元232的页面233中，页面233在图10中用重叠的矩形来表示。页面233通常的容量为1024个8位字。

现在参看图5、图6和图8，视频译码器205必须能够读出未与已有的区域网格209对齐的区域222。区域222与在区域212内的象素214对齐。为了搜索图象208的特征，或者为了检测顺序图象208之间的运动，需要视频译码器205能够读出未对齐区域222。

现在参看图5和图8，图5表示从RAM系统230被读出的区域222-1。区域222-1不与网格209对齐，而是与区域212的象素214对齐。用虚线来表示区域222-1，可以看到区域222-1与四个区域212、即与区域212-5、212-6、212-7和212-8重叠。

图8更详细地表示未对齐的读出区域222-1如何与在下的区域212-5、212-6、212-7和212-8重叠。区域222-1内的数字26代表在区域222-1和每一个在下的区域212之间边界处的象素214的数字213。注意未对齐的读出区域222-1主要由从一个在下的区域212-5中得到的象素214组成。从区域212-6、212-8和212-7得到的象素214的总数分别是7、7和1。从区域212-5得到49个象素。

在图5和图8中，直线网格209，没有示意页面233如何被交错，该页面233包括代表每一区域212的数据字215。可以设想各个页面233可以位于RAM系统230的同一个存储单元232中。各个页面与在某

一未对齐的读出区域222之下的所有区域212相关。如果这样的话，在最坏的情况下，生成的未对齐读出区域222将涉及从同一存储单元232中存取4个页面233，这一过程需要停滞时间来对该存储单元232进行3次预充电。对于图8上的所有未对齐区域222而言，该最坏的情形必将出现：搜索或匹配没有规定在未对齐区域222处，因此总会是最坏的情形。

比从同一存储单元232存取4个页面233好得多的是从每一存储单元232存取2个页面233，该过程是一个最大可能的交错处理。现在的问题是如何处理类似图8实例的情形，在图8中，未对齐的读出区域222-1的大部分象素214从一个在下的区域212-5得到，因此在进行从与其它3个在下的区域212-6、212-7和212-8相关的页面233中读出数据所需的相对较短的读出操作期间就几乎没有时间进行预充电。另一困难的情形是具有与两个区域212的相当大的部分以及与另外两个区域212的相当小的部分重叠的未对齐的读出区域222。

根据本发明，通过在某一两维图形240中对与区域212相关的页面233进行交错而解决了相对较短的读出时间的问题。现在参看图9和图10，在图9中，区域212的一半有阴影239而区域212的另一半没有阴影239。有阴影的区域212表示与该特定区域212相关的页面233位于存储单元0 232-0中。无阴影的区域212表示与该特定区域212相关的页面233位于存储单元1 232-1中。例如，区域212-6有阴影，因此与描述区域212-6的数据字215相关的页面233存储在存储单元0 232-0中，必须从存储单元0 232-0中读出。

根据本发明，基与页面233相关的区域212的特定的两维图形240，通过使交错器234将页面233交错到存储单元232之中，就解决了

相对较短的读出时间的问题。如图9所示，图形240类似方格盘：网格209的同一行中的两个顺序区域212在同一个存储单元232中不具有与它们相关的页面233，网格209的同一列中的两个顺序区域212在同一个存储单元232中不具有与它们相关的页面233。

在方格盘图形240中当未对齐的读出区域222与4个区域212重叠时，通过确保重叠区域212中的两个页面233存储在一个存储单元232中，与另外两个重叠区域212相关的页面233存储在其它存储单元232中就使交错的可能性为最大。例如，在图9中，区域212-5和212-7具有与它们相关的页面233存储在存储单元1 232-1中，而区域212-6和212-8具有与它们相关的页面233存储在存储单元0 232-0中。

为了使交错效率最大，应当从另外的存储单元232读出页面233。通过按照顺时针顺序或按照反时针顺序从未对齐的读出区域222之下的4个区域212中读出就可以保证这一点。作为沿顺时针方向读出的方法的一实例，考虑图9中的区域222-1之下的4个区域212。首先利用交错器234从存储单元1 232-1读出与区域212-5相关的页面233。然后读出与区域212-6相关的页面233，这是因为区域212-6与区域212-5在同一行并在区域212-5的右侧的缘故。由于区域212-7与区域212-6在同一列并在区域212-6之下，所以接着读出与区域212-7相关的页面233。由于区域212-8与区域212-7在同一行并在区域212-7的左侧，所以最后读出与区域212-8相关的页面233。

模拟已表现出交错存储单元232的方格盘图形240减少了停滞时间。可以选择方格盘图形240的高宽比和大小，以便最适应于特定应用。唯一的要求就是用少于一个页面233的字215来表示未对齐的读出区域222之下的至少一个区域212。这样一来，在下的区域212

的四种可能读出中的每一种都包括自身，将交错限制为所描述的机制。任何其它的分段存储都将涉及到更复杂的交错算法并将造成更长的停滞时间。

应当指出，有时候在下的区域212的每对可能的读出会包括同一页面233中的字215（例如，在未对齐的读出区域222-1之下的区域212-6和212-8的字215可以被包括在同一页面233上）。因此从页面233读出字215将更加有利。但是，最坏的情形仍然存在。

本发明的方法还可以没有任何性能损失地应用于需要具有比能够存储在一个页面233中的数据更多的数据字的面积(M×N)的区域212的情况。现在参看图10和图12，这种区域212被子网格252划分为子区域250，这样选择子区域250的面积，使得与每一子区域250相关的一些数据字215能够存储在一个页面233中。如图12所示，同样的方格盘图形240可应用于子区域250（例如，存储在同一存储单元232中的子区域250-1和250-3，以及存储在同一存储单元232中的子区域250-2和250-4），使得能够以与管理的区域212本身相同的效率方式来管理与每一“过大”区域212相关的字215的存取。

根据本发明提供的对存储器进行寻址的方法和设备。特别是，需要使用一固定宽字来寻址可变数据的过程。在各种形式的实施例中，固定宽字可以包括宽度定义字段、地址字段或替换字段。该固定宽字的长度由被编址的存储器的数量预先确定。构成本发明的设备包括具有算术中心部分的微码状态机。

微码状态机将被用来解决需要进行多变的和/或复杂的计算的设计问题。这种设计一些例子是：地址产生、数据流分析和译码，或滤波器抽头系数计算。寻址必须适应两种不同的特征，存取字的

可变宽部分的可变长度地址和地址替换。在本发明中，可以用具有 64×32 、 128×16 位、 256×8 位、 512×4 位、 1024×2 位或 2048×1 位格式的部分字来编址具有 64×32 位配置的RAM。

在许多应用中，用诸如替换、可变宽数据寻址或字的其它部分的压缩之类的动作定义该字的可变部分（将称为字段）是很有用的。这样做的惯用方法是用一附加字（或一些附加字）来规定在字内的字段（或一些字段）的宽度。以下将描述用字本身中的这一信息进行译码的方法。这一方法具有在字的全部定义中节省位的优点，简化了对编码字的译码并提供了已进行了什么译码的更直观的观察。如果可变宽字段是在字内对齐的最高或最低有效位，则这一编码方法就能是应用的。

表1-1表示在8位字内定义的、对齐的可变宽字段（标为“F”）最低有效位的两个例子，“W”标记这些字的其它可能的字段。

表1-1

位数(十六进制)	7	6	5	4	3	2	1	0
固定字	W	W	W	F	F	F	F	F
	W	W	W	W	W	W	F	F

表1-2表示通过增加足够的位来规定二进制中的字段的最大宽度而对表1-1所示字段进行编码的惯用方法。（标为“X”的位为“不考虑” - 本领域的术语）。这一方法占用过多的空间。

表1-2

位数(十六进制)	7 6 5 4 3 2 1 0 字段定义
固定字	W W W X X X X X 1 0 1
	W W W W W W X X 0 1 0

表1-3表示利用新的方法对表1-1所示字段进行编码。这一方法通过使用连续标记和终止标记来定义字段。在这一情形中，连续标记是“1”而终止标记是“0”。字段被定义为自该字段的对齐端

(在这一情形中为最低位)起的全部连续标记并包括终止标记。由表1-3可见，为了编码字段终止标记所占的空间必须附加到固定宽字的字段起始处，这就能够利用包括终止标记的附加空间来定义零长度字段。

表1-3

位数(十六进制)	7 6 5 4 3 2 1 0
固定字 连续标记 = 1; 终止标记 = 0。	W W W 0 1 1 1 1 1
	W W W W W W 0 1 1

可以看出，这一编码方法的优点是：

- 1、减少了编码所需的位数。
- 2、由于通常所需的对表1-2所示的“字段定义”，进行“ 2^x 的X至1”译码的译码要求，是 2^x 的1的形式的编码的固有性质，所以简化了所需的译码。
- 3、编码是更直观的方式，以便容易地识别定义的字段。

通过指出可以交换终止标记和连续标记来使表1-3的编码类似表1-4的编码就能够扩大这一编码的使用。此外，在这一应用的过程中可以互换“1”和“0”。

表1-4

位数(十六进制)	7	6	5	4	3	2	1	0
固定字	W	W	W	1	0	0	0	0
连续标记 = 1; 终止标记 = 0。	W	W	W	W	W	W	1	0

字段还可以是例如表1-5所示的对齐的最高有效位。以类似于最低有效位对齐字段的方式编码这些字段，字段从最高有效位（此后称为MSB）指向最低有效位（此后称为LSB）并包括第一个终止标记。表1-5所示字段的编码如表1-6所示。

表1-5

位数(十六进制)	7 6 5 4 3 2 1 0
固定字	F F F F F W W W
	F F W W W W W W

表1-6

位数(十六进制)	7 6 5 4 3 2 1 0
固定字	1 1 1 1 1 0 W W W
连续标记 = 1; 终止标记 = 0。	1 1 0 W W W W W W

最后，可以同时从字的最低位和最高位端对字段进行编码。例如，表1-7所示的两个字段可以如在表1-8中那样来编码，由于以上所述的理由，对每一字段只增加一个位。

表1-7

位数(十六进制)	7 6 5 4 3 2 1 0
固定字	F F F F W W F F
	W W W W F F F F

表1-8

位数(十六进制)	7 6 5 4 3 2 1 0
固定字 连续标记 = 1; 终止标记 = 0。	1 1 1 1 0 W W 0 1 1
	0 W W W W 0 1 1 1 1

在图21中表示上述一般概念。不是必需用来寻址数据的地址字段具有一个带有终止标记和连续标记的字段。在这种情况下，该字段与最低有效位对齐。

有时候用另外的值来替换存储器地址的一个部分是很有用的。这样就能够构成依赖数据的地址。可以将编码方法应用于存储器的编址，以便指定地址的哪一部分要被替换。如果在这一地址中使用了最低有效位对齐的可变长度字段，就能够定义替换字段。例如，将被编码的12位地址0baaaaaaaaaaaaaa的5个最低有效位由12位值

0bcccccccccccc替换将是0baaaaaaa011111并将产生地址
 0baaaaaaaccccc。表1-9表示替换为12位地址的编码。

表1-9: 地址替换

替换位号	B	A	9	8	7	6	5	4	3	2	1	0	
0	a	a	a	a	a	a	a	a	a	a	a	a	1
1	a	a	a	a	a	a	a	a	a	a	a	0	1
2	a	a	a	a	a	a	a	a	a	a	0	1	1
3	a	a	a	a	a	a	a	a	a	0	1	1	1
4	a	a	a	a	a	a	a	a	0	1	1	1	1
5	a	a	a	a	a	a	a	0	1	1	1	1	1
6	a	a	a	a	a	a	0	1	1	1	1	1	1
7	a	a	a	a	a	0	1	1	1	1	1	1	1
8	a	a	a	a	0	1	1	1	1	1	1	1	1

表1-9: 地址替换(续)

替换位号	B	A	9	8	7	6	5	4	3	2	1	0
9	a	a	a	0	1	1	1	1	1	1	1	1
10	a	a	0	1	1	1	1	1	1	1	1	1
11	a	0	1	1	1	1	1	1	1	1	1	1
12	0	1	1	1	1	1	1	1	1	1	1	1

在图19中，用于寻址的固定宽字带有任选的替换指示符的地址字段。如上所述，替换字段的大小可变并且起到用外部寻址源来替换可变量的地址位“a”的作用。替换发生以便代替终止标记位“Y”和连续标记位“X”。

终止标记告诉地址译码电路替换在哪里停止。连续标记填充固定宽字。

如果总是使用替换，则无需指示符。但是，替换指示符能最有效地用于替换。

本发明的一实施例是用于寻址可被存取的存储器，其整个宽度可被存取或从 2^n 的宽度直到其整个宽度可被存取（这些较短的字称为部分字）。它将指出可变字段编码如何可被用来寻址这一存储器以及将那些地址变址(index)到存储器中。

为了存取宽度为32、16、8、4、2和1位的64×32位寄存器文件需要不同长度的地址。有和16位存储单元的两倍一样多的32位存储单元和比1位存储单元多32倍的32位存储单元。此外多达8位的这一地址可被变址寄存器来替换。因此，可变的信息量必须被编码为固定数目的微码位。一种方法是将3位字段用于宽度和用于被替换的LSB的个数，而将一个12位字段用于地址，产生一18位的微码(microcode)字。但是，较好的方法是使用可变宽度字段对齐的最高有效位来压缩其宽度可被定义的地址，因此就能够定义存取的宽度。例如，6位地址指示32位的存取而12位地址指示1位的存取。这表示在表1-10，表中连续标记是“0”，终止标记是“1”。可以看出可变宽度字段如何压缩了如此定义其宽度的地址“a.....a”和如此存取宽度。用于寻址的固定宽度字的一般情形如图18所示。

表1-10: 可变宽寻址

数据宽度	A	9	8	7	6	5	4	3	2	1	0
1	1	a	a	a	a	a	a	a	a	a	a
2	0	1	a	a	a	a	a	a	a	a	a
4	0	0	1	a	a	a	a	a	a	a	a
8	0	0	0	1	a	a	a	a	a	a	a
16	0	0	0	0	1	a	a	a	a	a	a
32	0	0	0	0	0	1	a	a	a	a	a

为了允许地址存储单元的变址,可用其它值来替换地址“a...a”的一部分,该地址的替换部分(或字段)可用叠加在表1-10所示可变宽字段的顶部的最低有效位对齐的可变宽字段(连续标记“1”;终止标记“0”)来定义。表1-11作为例子使用8位字的地址来表示如何定义要被替换的最低有效位的数。附加的最低有效位是替换指示字(标记为“W”)。用于替换的固定宽字的一般情形如图20所

示。

表1-11: 地址替换

替换位	A	9	8	7	6	5	4	3	2	1	0	W
0	0	0	0	1	a	a	a	a	a	a	a	0
1	0	0	0	1	a	a	a	a	a	a	0	1
2	0	0	0	1	a	a	a	a	a	0	1	1
3	0	0	0	1	a	a	a	a	0	1	1	1
4	0	0	0	1	a	a	a	0	1	1	1	1
5	0	0	0	1	a	a	a	0	1	1	1	1
6	0	0	0	1	a	a	0	1	1	1	1	1
7	0	0	0	1	a	0	1	1	1	1	1	1
8	0	0	0	1	0	1	1	1	1	1	1	1

事实上, 替换代码叠加在已被编码的地址的顶部。

从这一编码可以看出存在非法地址，最明显的是 0×0000 和 $0 \times 3fff$ ，此时“0”必须在低部的9个位中，以便防止替换多于8个的位，“1”必须在顶部的6个位中，以便指定允许的存取宽度。如果这些错误之一被检测，存取就未被定义，但寄存器文件内容将不受影响。

以下将讨论用于寻址的设备和在寄存器文件中存取部分字的方法。惯用的存储器电路限定必须总是对存储器的整个宽度进行存取。为了实现可变宽度存取，读出全（32位）宽度字。对该全字进行循环移位，直到被存取的部分字在LSB中被对齐。该字的高位部分扩展到整个宽度，然后输出。扩展可以包括零或1的填充、符号扩展，使用符号数值的符号位作为新的MSB或任何类似的惯用方法。扩展依赖于操作的模式。当部分字被输入以便回写到存储器中时，它被多路转换回到被循环的全字，被循环的全字然后回转并写入阵列之中。图15表示在32字的第四个4位字中存取一四位部分字的这些步骤。

为了存取或读出部分字，例如在图15的行“1”中的增强亮度的四位字，就必须如行“2”所示将全宽字循环以便将部分字放置在LSB。如行“3”所示，4位字被扩展以便产生全32位字。该字现在能够被存取了。

已被选择回写的全宽字被截断为原来部分字的宽度，该部分字被多路转换为行“2”所示的位于LSB位置处的字，它示意于行“4”。得到的字被转回到其在读出字中的原来的有效位，它示意于行“5”。现在该全字可以被回写到寄存器文件中去。

以下概括了在图15中编号的各步骤：

- 1、从存储器读出全字
- 2、12位右循环移位将部分字放入LSB
- 3、扩展为全字，然后输出
- 4、输入的部分字被多路转换为来自(2)的被循环移位的全字
- 5、12位左循环移位使全字回复到原样，以使其被写入。

上述存取设定了图16所示存储器数据流结构。结构中的数字即图15及以上说明中所指数字。

必须对存储器地址译码，以使其控制上述结构。应当知道任何宽度地址的MSB相对于存储器都位于相同的有效位。译码地址的顶部6个位是32位字地址，余下的位是位地址。因此译码级（与替换平行）将通过检测最高有效终止标记的位置来译码地址宽度定义的可变字段。这就使地址成为对齐的MSB（在LSB处移位为零）。顶部6个位可被直接用作32位字的行地址。低部5个位可被用来直接控制两个柱形(barrel)移位器（如图16所示），例如，因为原来32位地址总是具有0b00000的移位（当地址是MSB对齐时已被移了位），所以类似地，16位地址可以具有0bX0000的移位，即0或16位移位，而1位地址可以具有0bXXXXX的移位，即0至31位移位。扩展器和输入多路转换器由存取宽度译码来控制，以便分别表征出输出字和将输入字多路转换为合适的有效位。译码的方框图如图17所示。可以看出，可以并行和独立地进行宽度和替换的两个可变宽字段的译码。

图20表示用于寻址可变宽数据和寻址如底部两行所示的替换的13位长度的固定宽字的一实例。对于这一例子，一8位字已在位置0b1101SSSS处被编址，这里“SSSS”用另一地址源来替换。

在具有图13所示结构的微码状态机中已将对存储器地址的替换

和对存储器的可变宽度存取合并起来。这种结构是一种利用称为微码指令的宽度字控制信号对算术中心部分进行控制的状态机。算术中心部分依次又将状态标志和某些数据送给状态机。

状态机由包括一系列微码指令的存储器组成。如同惯用的微码状态机一样，它能够顺序地通过一系列微码指令进行处理，或者能够将任何指令转移为任何其它指令。转换(jump)地址具有图19的形式。如图13和图14所示，被替换值来自算术中心部分。这使其能够在微码程序内构成“转移表”。因此，例如，如果用3个被替换的位来进行转移，则根据来自算术中心部分的值，就可能有可被转移的8个相邻的位置，这样就成为可编程的转移。

如图14所示，算术中心部分由称为寄存器文件的存储器、算术和逻辑单元(ALU)、输入端口和输出端口组成。这些部件利用总线和多路转换器来连接。如上所述，这些部件和规定它们连接关系的多路转换器完全由状态机发出的微码指令来控制。ALU和端口都是惯用的，但寄存器文件是允许对其进行可变宽变址存取的存储器。寄存器文件的地址被直接编码成为微码指令。

利用对寄存器文件进行寻址的这一方法的优点首先在于应用中的许多存储单元不必是存储器的全字宽度(在这一情形中为32位)。虽然使用全宽的存储单元对设备的操作不造成任何影响，但这造成存储器存储单元的极大浪费。减少所用存储器存储单元的数目将减少存储器所用的空间，因此就减少了在寄存器文件中的装入容量，并由此增大了寄存器文件的速度。其次，与可变宽度存储器存取结合在一起的变址允许通过可变宽度的存储单元进行操作(stepping)。在1个位的情形时，这就可以精确地进行长度的除法和乘法运算。

总而言之，上述公开了对存储器进行寻址的具有以下步骤的过程：提供具有预定固定位数的固定宽字来对可变宽数据进行寻址，用宽度定义字段和给该宽度定义字段提供了至少一个位，用作终止标记的地址字段来定义固定宽字，用确定数据的地址的多个位来定义地址字段，以反比于可变宽数据的大小改变地址字段中的位的大小，以正比于可变宽数据的大小改变宽度定义字段中的位数，以及在改变宽度定义字段和地址字段的宽度时保持对可变宽数据进行寻址的固定宽字。此外，还公开了对存储器进行寻址的具有以下步骤的过程：提供具有预定固定位数的固定宽度字来对数据进行寻址，用地址字段和替换字段来定义固定宽度字，用确定数据的地址的多个位来定义地址字段，用至少一个替换位来定义可变宽度替换字段，该替换字段具有至少一个位作为地址字段和替换字段之间的终止标记，使用该替换字段来表示来自各个寻址源的被替换位，以及在逆向地改变地址字段的宽度和替换字段的宽度时保持对可变宽度数据进行寻址的固定宽度字。此外，对存储器中的可变宽度数据进行寻址的过程具有以下步骤：提供具有预定宽度并由部分字组成的字的存储器，将被存取的部分字循环移位成为最低有效位对齐，扩展字的剩余部分使被存取的字识别为部分字，恢复字的剩余部分并循环移位该字直到部分字被恢复到其原来位置为止。

根据本发明的缓冲器管理器，一种图象格式器如图24所示，有两个地址产生器，一个为写地址产生器410，另一个为读地址产生器420，控制这两个地址产生器410和420并提供帧频率转换的缓冲器管理器430，包括垂直和水平过取样器(upsampler)、色空间变换和伽玛校正的数据处理流水线，以及调整处理流水线的输出的最后控制部件。

到达图象格式器输入端的记号在FIFO 440中被缓冲并被传送给缓冲器管理器430。这一部件检测新图象的到达并确定用来存储每一新图象的缓冲器的可用性。如果有可用的缓冲器，就将该缓冲器指定给到达的图象并将其参见号传送给写地址产生器410。如果没有可用的缓冲器，就暂缓处理输入图象，直到有可用的缓冲器为止。全部记号都输入到写地址产生器410。

每当读地址产生器420从显示系统接收了VSYNC信号时，就向缓冲器管理器430请求新的显示缓冲器的参见号(index)。如果有包含完整图象数据的缓冲器，并且该图象被认为可以显示，则该缓冲器的参见号就传送给显示地址产生器。否则，缓冲器管理器就传送被显示的最后缓冲器的参见号。在启动时，传送零作为参见号，直到第一个缓冲器满为止。如果图象的编号（在每一图象输入时被计算）大于或等于给出在编码帧频率的显示屏上期待的图象编号（扫描编号），则认为该图象可被显示。该期待的图象编号通过计数图象时钟脉冲来确定，图象时钟可以由时钟分频器在内部产生，也可以从外部产生。这一技术使帧频率变换（例如2-3下变换pull-down）。

外部DRAM用作缓冲器，在数目上可以是两个或三个。如果要进行帧频率(frame-rate)变换，就需要三个。

缓冲器管理器430的目的是给地址产生器提供表示任何两个或三个用于图象数据的写入和读出的外部缓冲器的标记。这些标记的分配受到三种主要因素的影响，每一因素代表在操作中的一种定时方式的作用：图象数据到达图象格式器输入端的速率（编码数据速率），数据被显示的速率（显示数据速率），以及编码视频序列的帧频率（扫描速率）。

一个三缓冲器系统使扫描速率和显示速率不同(例如2-3下变换),因此,如果需要的话,一些帧可被重复或跳过,以便得到给出了系统的定时约束的最好的可能帧序列。也可以用类似的方法来调节在译码方面有某些困难的图象,因此,如果图象读出的时间长于可利用的译码显示时间,则在其他一切“准备就绪”时重复以前的帧。在一个两缓冲器系统中,这三种定时方式必须被锁定,正是第三个缓冲器提供了作为备用部分的灵活性。

缓冲器管理器通过保留与每一外部缓冲器相关的某些状态信息来运行-这些状态信息包括一些标记,用以表示缓冲器是正在使用、装满数据,或者是作好显示的准备,以及在当前存储在缓冲器中的图象序列内的图象编号。扫描编号也被记录,这是每当接收了图象时钟脉冲时就被递增的数字,代表了在编码序列的帧频的基础上当前可期待被显示的图象编号。

每当在输入端检测到 PICTURE_START 记号就指定到达缓冲器(要被写入输入数据的缓冲器),然后将这一缓冲器标记为IN_USE;在PICTURE_END状态时,到达缓冲器将要被去除指定,根据图象编号和扫描编号之间的关系将该缓冲器标记为FULL或 READY。

显示地址产生器一旦通过两线接口对每一个VSYNC请求新的显示缓冲器。如果有标记为READY的缓冲器,则缓冲器管理器就将该缓冲器指定作显示用。如果没有READY的缓冲器,就重复以前显示的缓冲器。

每当扫描编号改变,就对其进行检测,通过检查每一缓冲器的图象编号和扫描编号之间的关系来检测包括完整图象的每一缓冲器

的READY状态。按顺序考察缓冲器，当认为任何一个为READY时，以前被标记为READY的任一缓冲器的READY状态就自动地被取消，然后被标记为EMPTY。因为较后的图象编号被存储在被较后考虑的缓冲器中，所以借助于指定机制，这是可行的。

如果在输入流中指示出跳跃图象，则在H261中的TEMPORAL_REFERENCE记号就使缓冲器的图象编号更改。TEMPORAL_REFEREAENCE记号在MPEG中无作用。

FLUSH记号使输入暂缓处理，直到每一个缓冲器为EMPTY或已被指定作为显示缓冲器为止；扫描编号和图象编号然后被复位，并可以开始新的序列。

所有数据都从先进先出输入端bm_front输入到缓冲器管理器。这一传送通过两线接口来进行，数据为8位宽加一扩展位。到达缓冲器管理器的所有数据都被保证是完整的记号，这是对扫描编号进行连续处理所必需的，在数据上游(upstream)中有大的间隙时，显示缓冲器就进行请求。

记号(8位数据，1位扩展)通过两线接口传送给写地址产生器。到达缓冲器的参见号也根据同一接口传送，所以可得到正确的参见号，用于地址产生的时刻与PICTURE_START记号到达写地址产生器的时刻相同。

读地址产生器的接口包括两个单独的两线接口，它们可被认为分别作为“请求”和“应答”信号，但是，由于在两端的两个基于两线的状态机的缘故，单线是不合适的。

与显示地址接口相关的正常事件序列如下：显示地址响应显示设备的vsync，通过对缓冲器管理器确认drg_Valid输入而产生一请

求；当缓冲器管理器到达在其状态机中的合适状态时，它将接受该请求并着手指定要被显示的缓冲器；然后确认disp-valid线，传送缓冲器参见号，这通常将立即被显示地址接受。存在与这一最后的两线接口相关的附加导线(rst-fld)，该附加导线指出：不考虑以前的段号而必须复位与当前参见号相关的段号。

缓冲器管理器部件使用四位的微处理器地址空间，连同8位的数据总线以及读和写选通脉冲。有两个选择信号，一个表示用户可存取的存储单元，另一个表示在通常的操作条件下不需要存取的测试存储单元。

缓冲器管理器能够产生两个不同的事件：参见号发现和后到达。当其PICTURE_START扩展字节（图象索引）与在准备期间写入到BU_BM_TARGET_IX寄存器的值一致的图象到达时，就确认这两个事件的第一个事件。当指定了其图象编号小于当前扫描编号的显示缓冲器时，即没有对在到达缓冲器管理器的系统流水线中的处理进行管理以便跟上扫描需求时，第二事件出现。图象时钟是扫描编号计数器的时钟信号，或者在芯片上产生、或者取自外部源（通常是显示系统）。缓冲器管理器接受这两个信号并根据pclk_ext的值（在缓冲器管理器的控制寄存器中的位）选择其一。这一信号还作为填充picoutpad的允许信号，因此，如果图象格式器正在产生其自己的图象时钟，这一信号还可从芯片中作为输出信号来得到。

在缓冲器管理器的状态机中有19个状态。这些状态如图25所示相互作用。复位状态是PRESO，标志被置为零，使得一开始就使主环路循环。

状态机的主环路包括图26所示的状态（主图即图25中的要点）。

状态PRES0和PRES1涉及利用信号presflg对图象时钟进行检测。由于它们都依赖于rdytst的值，所以对涉及的检测允许两个周期。如果检测了扫描标志，就对所有的缓冲器检查可能的“准备就绪”，否则状态机就到达状态DRQ。PRES0-PRES1环路的每个周期检查不同的缓冲器，检查满和准备就绪状况：如果所有这些都满足，就清除以前准备就绪缓冲器（如果有一个），指定新的准备就绪缓冲器并更新其状态。重复这一过程直到所有缓冲器都已被检查为止（参见号 = max buf），然后到达下一状态。当任一以下条件为真时，就认为缓冲器对显示是准备就绪的：

`(pic_num > pres_num) && ((pic_num - pres_num) > = 128)`

或者

`(pic_num < pres_num) && ((pres_num - pic_num) < = 128)`

或者

`pic_num = pres_num`

状态DRQ检查对显示缓冲器的请求 (`drq_Valid_reg && disp_acc_reg`)。如果没有请求，该状态提前（通常到达状态TOKEN - 在这之后再到达其它状态），否则就如下地发出显示缓冲器参见号：如果没有准备就绪缓冲器，就再发出以前的参见号，或者，如果没有以前的显示缓冲器，就发出空参见号（零）；如果缓冲器做好显示准备，就发出其参见号并更新其状态 - 如果需要的话，就清除以前的显示缓冲器。然后状态机就象以前一样地提前。

状态TOKEN是完成主环路的通常任选项：如果有有效输入和输出未被暂缓处理，就检查记号的关键值（以后描述），否则控制返回到状态PRES0。

当某些条件被满足时，控制才偏离主环路。这些条件在以下的部分中描述。

如果在PRES0-PRES1环路期间，缓冲器被确定为准备就绪，则因为在任何时刻只有一个缓冲器能够被指定为准备就绪，所以任何以前准备就绪的缓冲器都需要被腾空。状态VACATE_RDY通过将以前准备就绪缓冲器的状态设置为VACANT(腾空)而将其清除，并且它将缓冲器参见号复位为1，因此，当控制返回到PRES0状态时，将检查所有缓冲器的准备就绪状态。这样做的理由是参见号现在正指向以前准备就绪的缓冲器(为了清除的目的)并且没有我们想要的新的准备就绪缓冲器参见号的记录 - 因此需要重新测试所有的缓冲器。

直接根据状态DRQ (状态USE_RDY) 或利用清除以前显示缓冲器状态的状态VACATE_DISP来指定显示缓冲器参见号。选择的显示缓冲器被标记为IN_USE(在使用中)，rdy_buf的值被置为零，参见号被复位为1，以返回到状态DRQ。disp_buf被给定所需的参见号并由此控制两线接口导线(disp_valid和drq_acc)。控制只返回到状态DRQ，因在此状态USE_RDY不需要在状态TOKEN、FLUSH和ALLOC(指定)之间作出决定。

一旦接收了PICTURE_END记号，控制就从状态TOKEN转移到状态PICTURE_END，在该状态中，如果参见号还没有指向当前到达缓冲器，就将参见号设定为指向当前到达缓冲器，由此能够更新其状态。假定out_acc_reg和en_full都是真的，就可以如下地更新状态；如果都不是真的，控制就保持在状态PICTURE_END，直到它们都是真的为止。en_full信号由写地址产生器来提供，以便指出摆动减缓器已被摆动，即最后的存储块已被成功地写入。因此更新缓冲器状态是安全的。

检测刚刚写完的缓冲器的准备就绪状态并根据检测结果给出FULL或READY状态。如果它是准备就绪，rdy_buf就被给定它的参见号的值，set_lq_ev信号（最后到达事件）被置为高电平（表示期待的显示在时间上早于译码）。arr_buf的新值现在成为零，并且，如果以前准备就绪缓冲器的状态需要清除，则参见号就指向那里，控制移向状态VACATE_RDY；否则，就将参见号复位为1，并控制返回主环路的开始。

当PICTURE_START记号在状态TOKEN期间到达，就置位标志from_ps，使基本的状态机环路被改变，使得状态ALLOC而不是状态TOKEN被访问。状态ALLOC涉及指定到达缓冲器（可向其写入到达图象数据）并通过缓冲器循环，直到它发现其状态是VACANT的一个缓冲器为止。只有out_acc_reg为高电平才指定缓冲器，由于它在数据两线接口上被输出，因此环路的循环将继续到这一情形为止。一旦已发现合适的到达缓冲器，参见号就被指定为arr_buf，并将其状态标记为IN_USE。参见号被设定为1，标志from_ps被复位，将状态设定到NEW_EXP_TR。对图象的参见号（包括在跟随PICTURE_START的字中）进行检查，以便确定其是否与targ_ix（在准备期间规定的目标参见号）相同。如果是，设定set_if_ev为高电平（参见号发现事件）。

三个状态NEW_EXP_TR，SET_ARR_IX和NEW_PIC_NUM建立了新的期待暂时标准以及输入数据的图象编号 - 中间的状态就将参见号设定为arr_buf，因此正确的图象编号寄存器被更新（注意this_pnum也被更新）。然后控制到达状态OUTPUT_TAIL，该状态在出现低的扩展之前输出数据（假定有用的两线接口信号），此时主环路被重

新启动。这意味着全部数据块(64项)被输出,在这些数据块中没有扫描标志或显示请求的检测。

数据流中的FLUSH记号表示序列信息(扫描编号、图象编号、rst_flg)应当被复位。这只有当逐渐通向FLUSH的所有数据已被正确处理才能够发生,因此,在接收了FLUSH后,必须监测所有缓冲器的状态,直到确认所有帧已送交显示为止,即除一个缓冲器外的所有缓冲器具有状态EMPTY,这一个缓冲器处于IN_USE状态(作为显示缓冲器)。此时就能够安全地启动“新的序列”。

当在状态TOKEN检测了FLUSH记号时,标志from_flg被置位,使基本的状态机环路被改变,使得状态FLUSH而不是状态TOKEN被访问。状态FLUSH按序检查每一缓冲器的状态,等待每一缓冲器成为VACANT或IN_USE状态,作显示用。状态机就绕该环路循环,直到该条件为真为止,然后递增缓冲器的参见号并重复该过程,直到所有的缓冲器已被访问为止。当最后的缓冲器满足该条件时,扫描编号、图象编号和所有暂时基准寄存器都设定其复位值;rst_flg被置位为1。标志from_flg被复位,重新开始通常的主环路操作。

当遇到TEMPORAL_REFERENCE记号时,检测H261的位,如果被置位,就访问四个状态TEMP_PEF0至TEMP_REF3。这些访问执行以下操作:

TEMP_REF0: temp_ref = in_data_reg;

TEMP_REF1: delta = temp_ref - exp_tr; index = arr_buf;

TEMP_REF2: exp_tr = delta + exp_tr;

TEMP_REF3: pic_num[i] = this_pnum + delta; index = 1;

在除上述列出的情形外的所有情形中,状态TOKEN通过控制到

达状态OUTPUT_TAIL。控制保持在这一状态，直到遇到该记号的最后字(in_extn_reg为低电平)，然后重新进入主环路。

需要不断地检测图象时钟的“非同步”定时事件和显示缓冲器请求，以及在這些检测期间需要暂缓处理缓冲器管理器的输入就意味着当给缓冲器管理器的输入端连续提供数据时，对通过缓冲器管理器的数据速率就有约束。通常的状态序列可以是PRES0、PRES1、DRQ、TOKEN、OUTPUT_TAIL，除OUTPUT_TAIL外，每一个都持续一个周期。这意味着对每组64个数据项，将有3个周期的额外量，在此期间，输入被暂缓处理（在状态PRES0、PRES1和DRQ期间），由此将写速率降低 $3/64$ 或约5%。当状态机的辅助分支在最坏的情况下被执行时，这一数值有时往往可增大到13个周期的额外量。注意这样大的额外量(开销)只有在每帧一次的基础上来提供。

在upi存取期间扫描编号自由地运行；如果要求在放弃存取时的扫描编号与获得存取时的相同，就可以通过在存取被允许之后读出扫描编号并且就在存取被放弃之前将扫描编号回写而实现上述要求。注意这是同步的，因此可能需要将存取重复几次以确保存取为有效。

写地址产生器410从缓冲器管理器430接收记号并检测每一新DATA记号的到达。对于每次到达，它计算DRAM接口450的新地址，在该地址将存储该到达数据块。原始数据然后传送给DRAM接口450，利用该接口它被写入到摆动缓冲器。注意DRAM地址是块地址，而且DRAM中的图象被组织成光栅块。但是，输入的图象数据被组织成宏块序列，因此地址产生算法必须考虑这一点。

根据本发明的RAM接口，其中一高性能、可配置的DRAM接口500

如图22所示。这一接口是不依赖于标准的组件，用来直接驱动例如空间译码器、时间译码器和视频格式器所需的DRAM。不需要外部逻辑、缓冲器或部件来将该DRAM接口与在那些系统中的DRAM连接。

该接口在两个方面是可配置的。首先，可将接口的具体时序进行配置以便适应各种不同类型的DRAM。其次，可将与DRAM的数据接口的宽度进行配置以便为各种应用提供成本/性能的折衷。

在每一芯片上的DRAM接口将该芯片与外部DRAM连接。使用外部DRAM是由于目前在芯片上加工所需较大的DRAM是不实际的，但是，在芯片上加工所需大的DRAM是可能的。

虽然该DRAM接口是不依赖于标准的，但它仍必须被配置成满足多个标准，H.261、JPEG和MPEG中的每一个标准。如何再配置DRAM接口进行多标准操作在此将进一步讨论。

理解DRAM接口500的操作的一重要方面就是理解DRAM接口500和地址产生器510之间的关系，以及这两者如何利用两线接口进行通信。有两个地址产生器，一个为写地址产生器520，另一个为读地址产生器530。缓冲器管理器540控制这两个地址产生器520和530。该缓冲器管理器540在本说明书的其它处有较详细的说明。

简单地说，地址产生器顾名思义为需要寻址DRAM产生DRAM接口地址（例如，从DRAM中的某一地址读出或写入）。由于两线接口，只有当DRAM接口具有数据（取自流水线中的前级）和有效地址（取自地址产生器）时才进行读和写。如以下进一步所述的，单独的地址产生器的使用简化了地址产生器和DRAM接口的结构。

DRAM接口能够根据与地址产生器和与传送和接收数据的组件(block)的时钟都不同步的时钟进行工作。已用专门的技术来处理

操作的这一非同步性质。

数据通常以64个字节一组的形式在DRAM接口和芯片的其余部分之间传送。利用称为“摆动减缓器”的设备进行传送。这主要是一对以双缓冲结构的方式运行的RAM，DRAM接口填充或腾空一个RAM，而芯片的其它部分腾空或填充另一RAM。携带来自地址产生器的地址的单独总线与每一摆动减缓器相关。

每一芯片具有四个摆动减缓器，但这些摆动减缓器的功能在每一情形中是不同的。在空间译码器中，一个摆动减缓器用来将编码数据传送给DRAM，另一个从DRAM读出编码数据，第三个摆动减缓器将记号化数据传关给DRAM，第四个从DRAM读出记号化数据。在时间译码器中，一个摆动减缓器用来给DRAM写入内图象数据或预测图象数据，第二个从DRAM读出内数据或预测数据，其余两个读出前向和后向预测数据。在视频格式器中，一个摆动减缓器用来给DRAM传送数据，其余三个用来从DRAM读出数据，分别用于亮度(Y)以及红和蓝色差数据(C_r 和 C_b)。

以下描述具有一个写摆动减缓器502和一个读摆动减缓器504的DRAM接口的操作。

控制506作为地址产生器510和DRAM接口500的接口，提供和接收数据的芯片的其余部件都是两线接口。地址产生器510可以由于接收了控制记号而产生地址，或者就产生固定的地址序列。DRAM接口500按照特定的方式用地址产生器510来处理两线接口。当其做好接收和寻址准备时，它不是保持接受线为高电平，而是等待地址产生器提供有效地址，处理该地址，然后设定接受线高电平一个时钟周期。因此，它就实施了请求/应答(REQ/ACK)协议。

DRAM接口500的特有特征是其独立地与地址产生器510和与提供或接受数据的组件通信的能力。例如，地址产生器510可以产生与在写摆动减缓器502中的数据相关的地址，但在该写摆动减缓器502发出有一组数据已可被写入外部DRAM的信号之前不采取行动。类似地，写摆动减缓器502可以包括一组做好了被写入外部DRAM准备的数据，但在地址产生器510在合适的总线上提供地址之前不采取行动。此外，一旦在写摆动减缓器502中的一个RAM已被填充了数据，则在数据输入被暂缓之前（两线接口接受信号设定为低电平），另一个RAM可被填满并“摆动”到DRAM接口一侧。

为了理解DRAM接口500的操作，提出这一点是很重要的，即在适当配置的系统，DRAM接口500至少能够和在摆动减缓器502和504以及芯片的其余部分之间的所有平均数据率之和一样快地在摆动减缓器502和504以及外部DRAM之间传送数据。

每一DRAM接口500包含确定它下一次将为哪一个摆动减缓器服务的方法。一般来说，这将或者是“循环调度”（即被服务的摆动减缓器就是下一个可用的已刚刚转变的摆动减缓器）、或者是优先级译码器（即在该译码器中，某些摆动减缓器具有高于其它缓冲器的优先级）。在这两种情形中，一额外请求将出自刷新请求产生器，该额外请求具有比所有其它请求都高的优先级。该刷新请求由刷新计数器来产生，刷新计数器可通过微处理器接口来编程。

写摆动减缓器与两组RAM，即RAM1和RAM2接口。正如以下进一步讨论的那样，在写地址和控制的控制下，数据从在前的组件或级写入RAM1和RAM2。数据再从RAM1和RAM2写入DRAM。当将数据写入DRAM时，如在此进一步描述的那样，DRAM的行地址由地址产生器来

提供，而列地址产生器由写地址和控制来提供。在运行时，有效数据在输入端（数据输入）被提供。该数据从在前的级接收。由于每一段数据由DRAM接口500来接受，所以它被写入RAM1并且写地址控制递增RAM1的地址，以便允许下一段数据被写入RAM1。继续将数据写入RAM1，直到没有了数据或者RAM1满了为止。当RAM1满了，输入侧就放弃控制并给读出侧发送信号，指出RAM1现在已可被读出。该信号在两异步时钟状态之间传送，因此通过了三个同步触发器。

假定RAM2是空的，到达输入侧的下一项数据就写入RAM2，否则，当RAM2已被腾空时才这样做。当循环调度或优先级译码器（取决于特定的芯片使用哪一种）指出轮到这一摆动减缓器被读出时，DRAM接口就读出RAM1的内容并将它们写入外部DRAM。然后通过异步接口往回发送信号，指出RAM1现在又可被填充了。

如果DRAM接口在输入侧已填充RAM2之前腾空RAM1并“摆动”了它，则数据就可以不断地被摆动减缓器接受。否则，当RAM2被填充时，摆动减缓器将设定其接受信号为低电平，直到RAM1已被“摆动”回到被输入侧使用为止。读摆动减缓器的操作是类似的，但输入和输出数据总线被反向。

设计DRAM接口500以使其可用的存储器带宽为最大。每一 8×8 的数据块被存储在同一DRAM页面中。这样一来，就能够充分利用DRAM的快速页面存取模式，在这一模式中，一行地址之后是许多列地址。特别是，如以下进一步讨论的那样，行地址由地址产生器510来提供，而列地址由DRAM接口500来提供。

此外，提供了允许到外部DRAM的数据总线为8、16或32位宽的便利条件，因此所用DRAM的大小量能够满足特定应用的大小和带宽

需求。

在这一例子中，地址产生器510给DRAM接口500提供了每一读和写摆动减缓器505和502的组件地址。该地址被用作DRAM的行地址。6位列地址由DRAM接口本身提供，并且这些位还被用作摆动减缓器RAM的地址。到摆动减缓器的数据总线是32位宽，因此，如果到外部DRAM的总线宽度小于32位，则在从写摆动减缓器读出下一个字或将下一个字写入读摆动减缓器（读和写指相对于外部DRAM的传送方向）之前，必须存取两或四个外部DRAM。

应当理解DRAM接口500不限于两个摆动减缓器。

应当懂得某些结构的变化已呈现在本领域技术人员的面前。应当理解上述详细描述是说明性的，本发明的构思和范围依所附权利要求的内容为准。

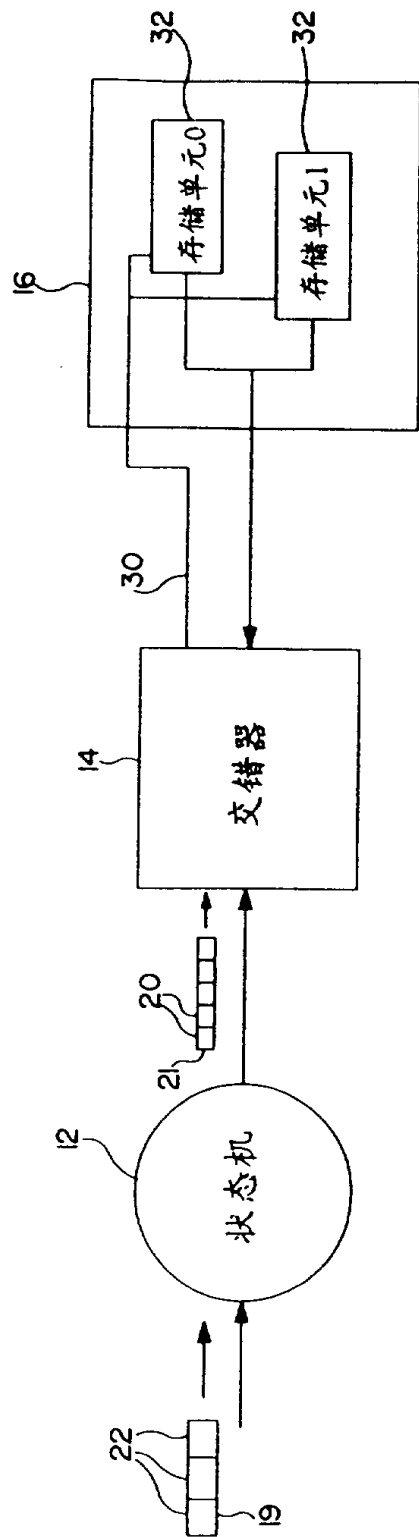


图1

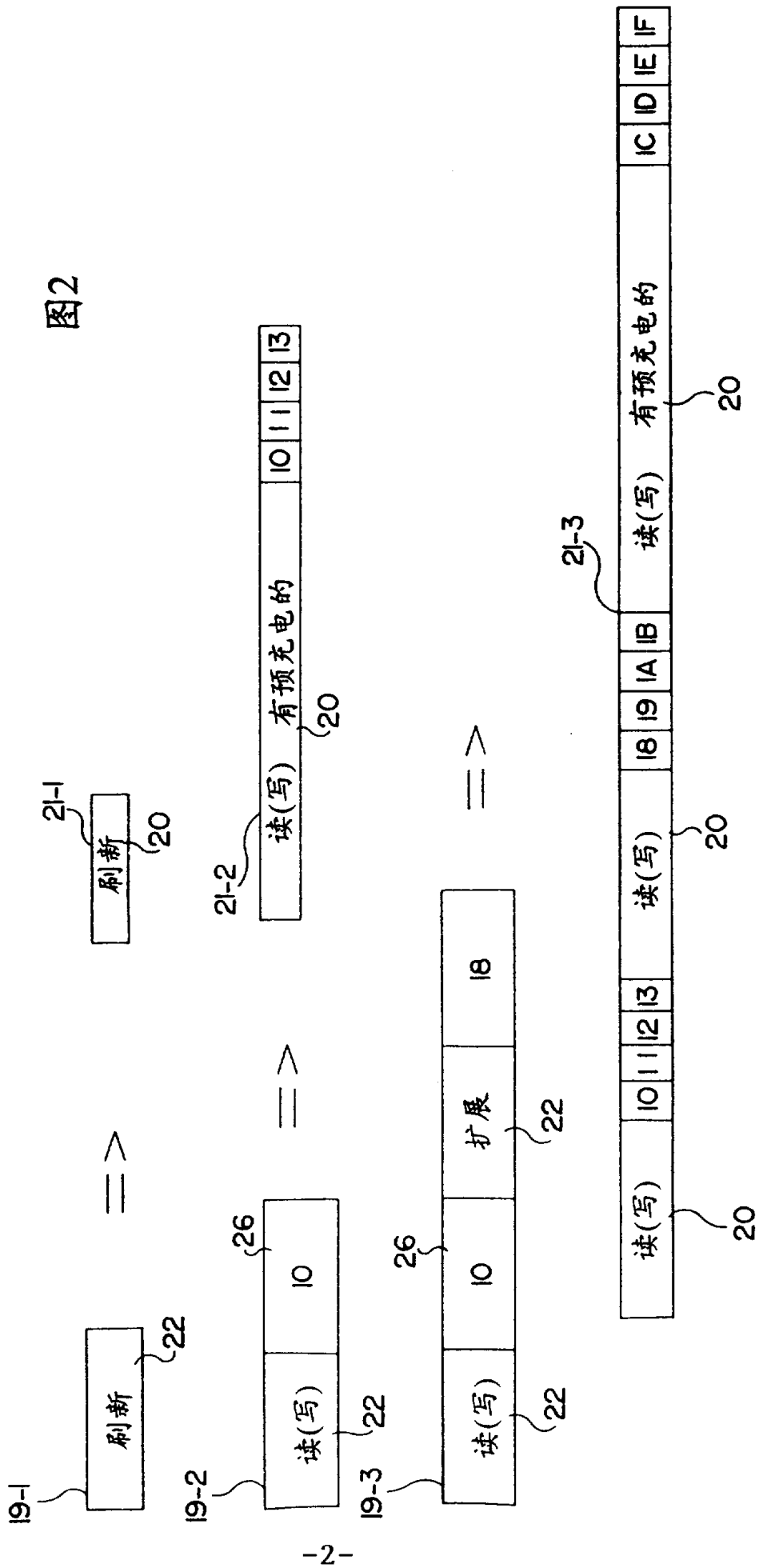


图2

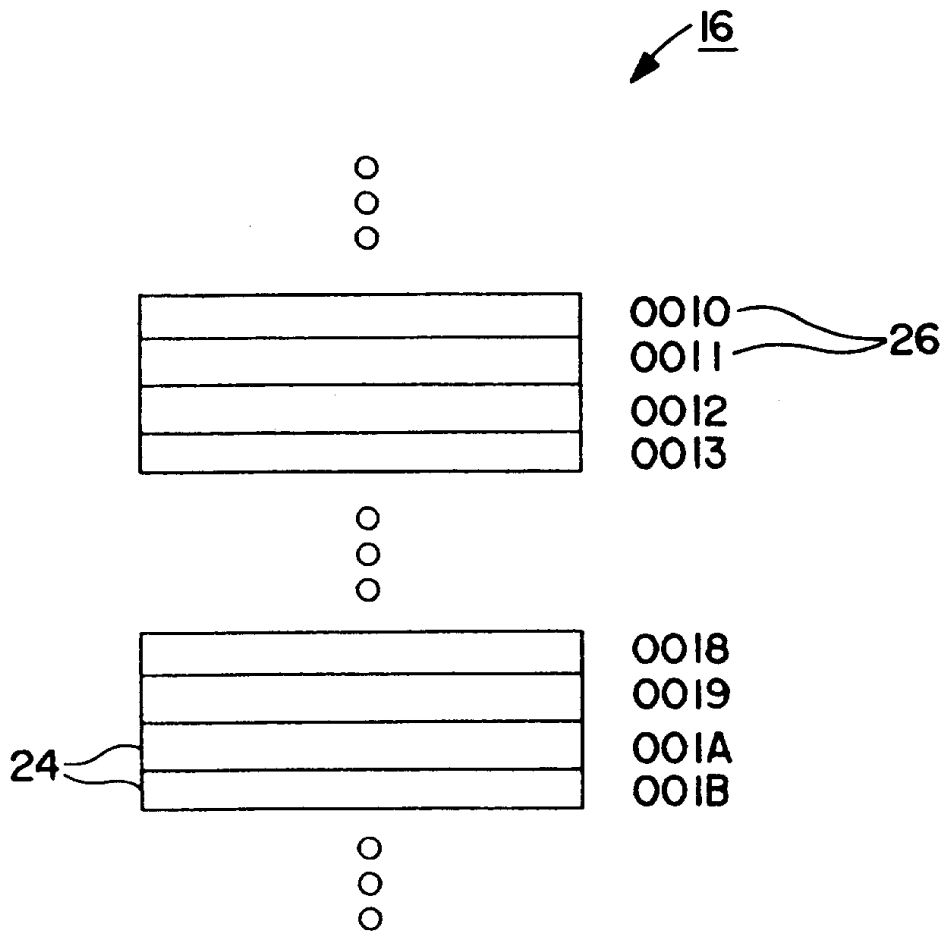


图3

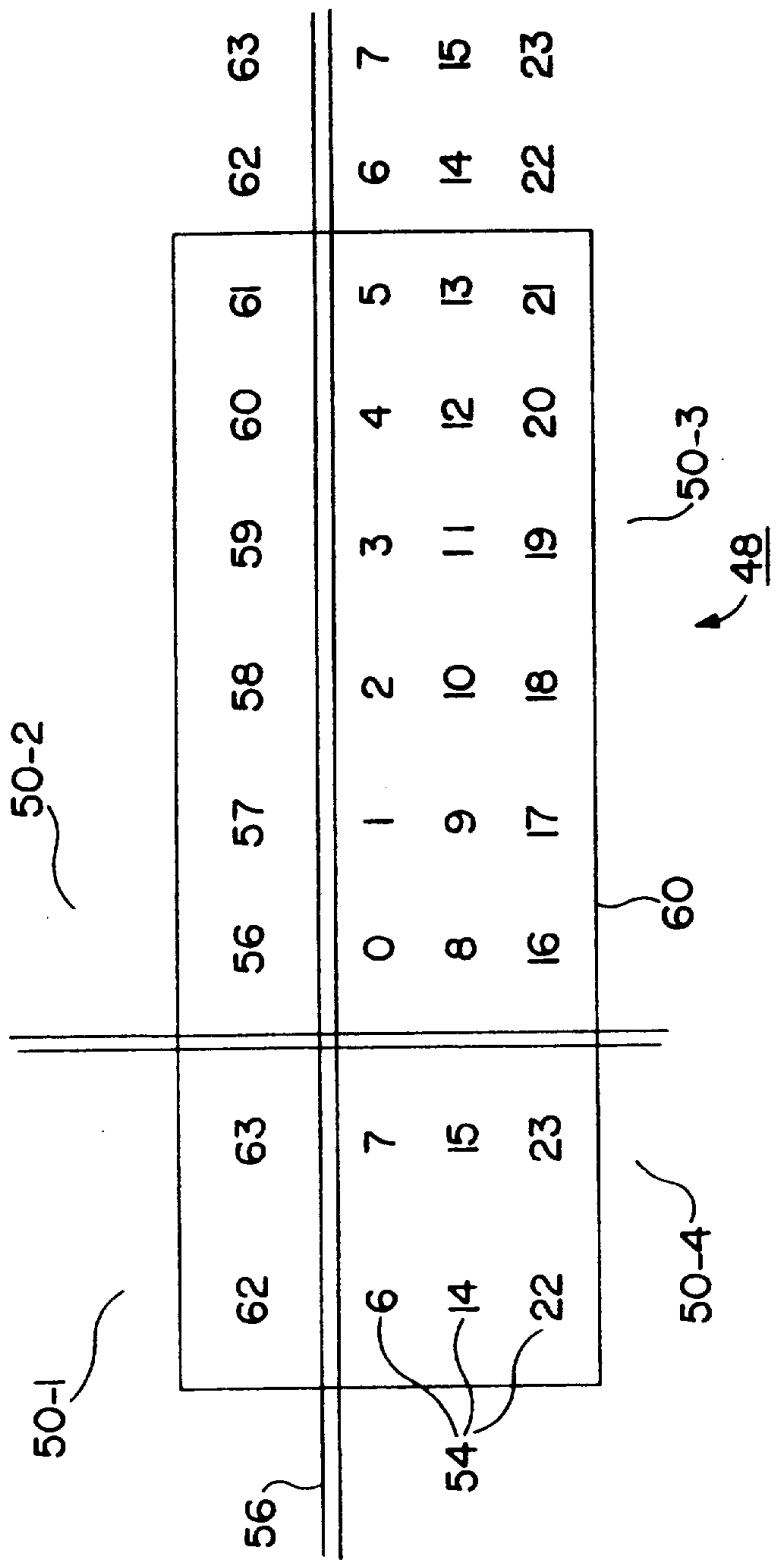


图4

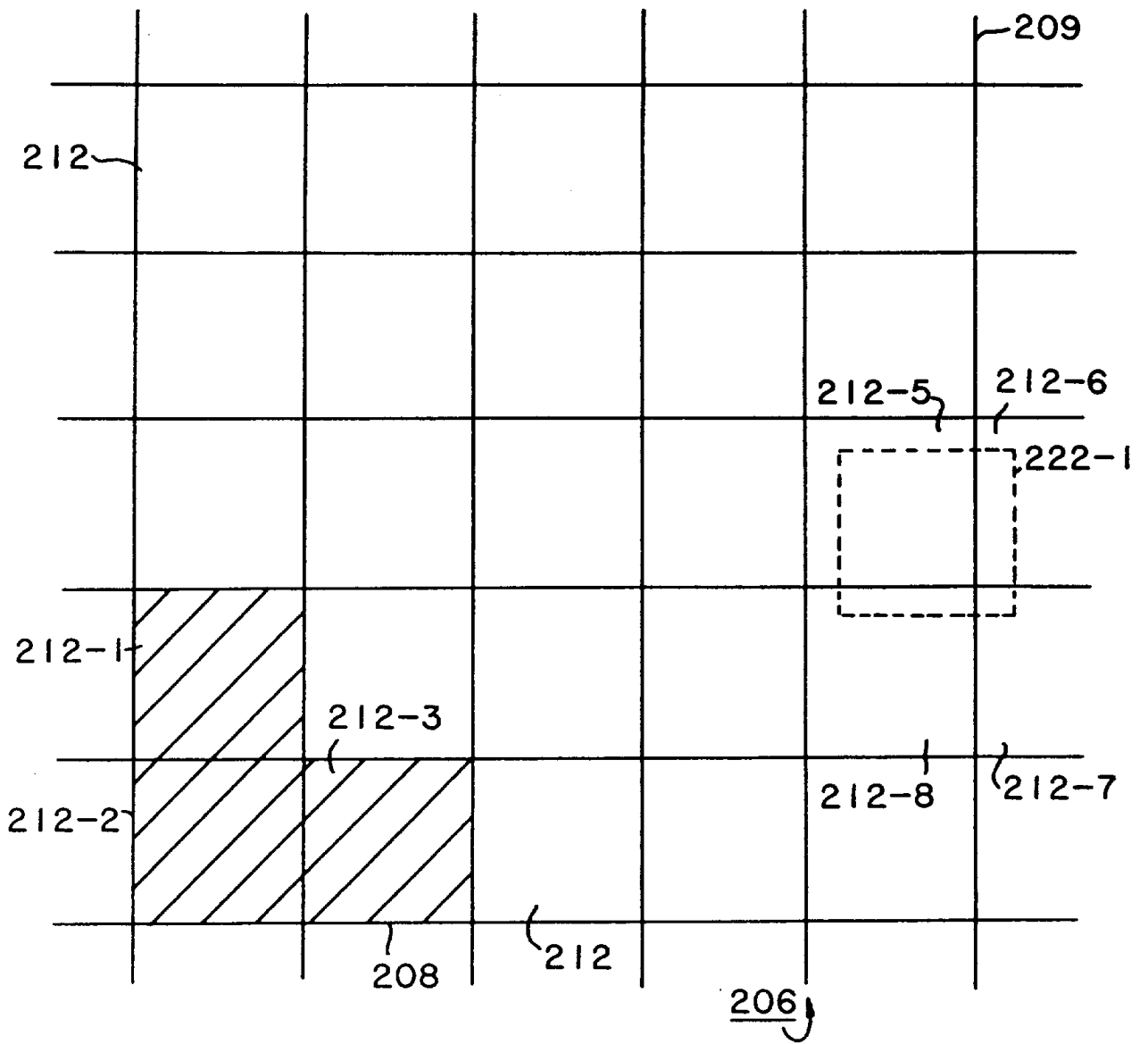


图5

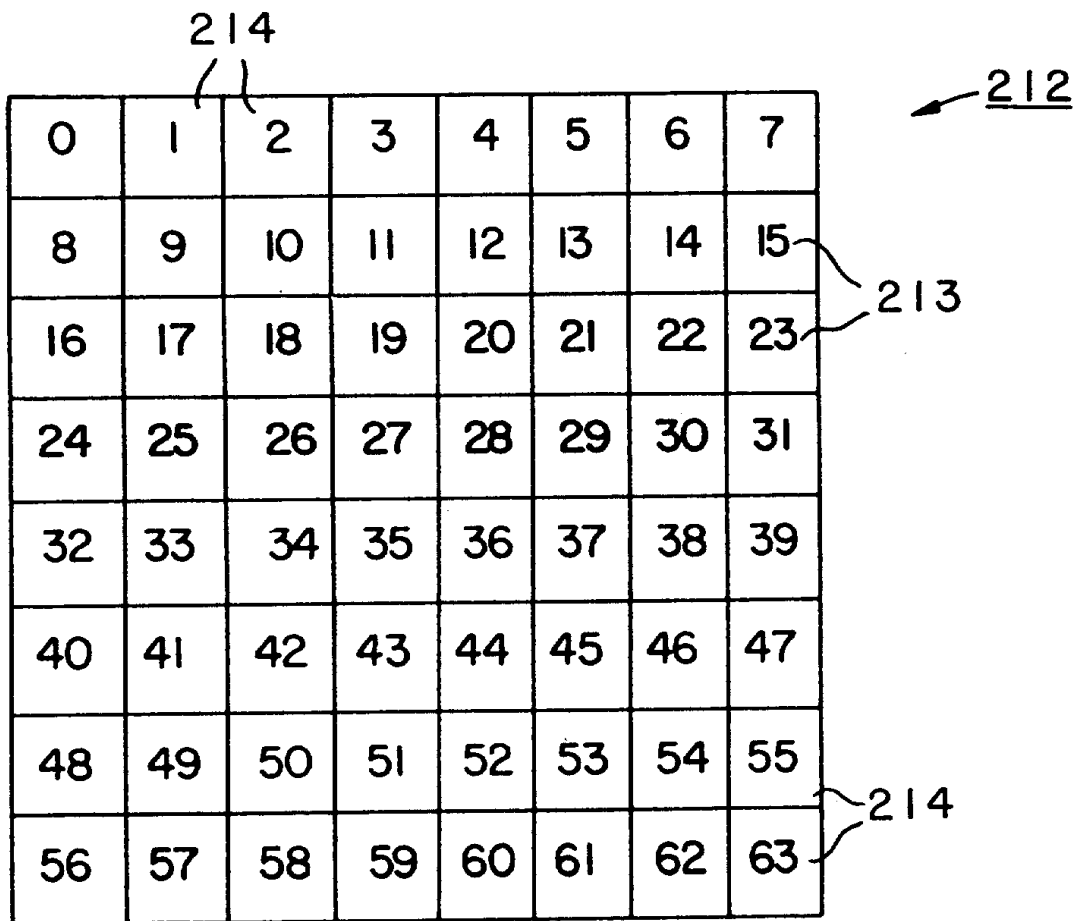


图6

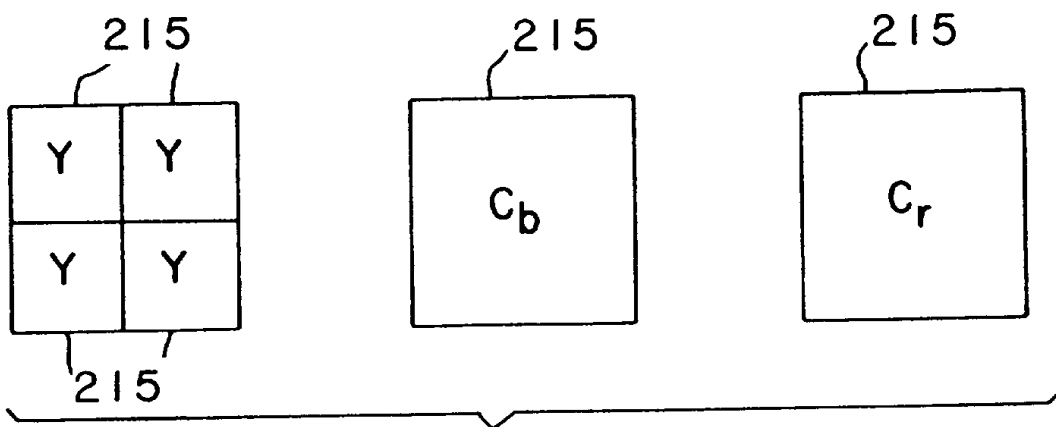


图7

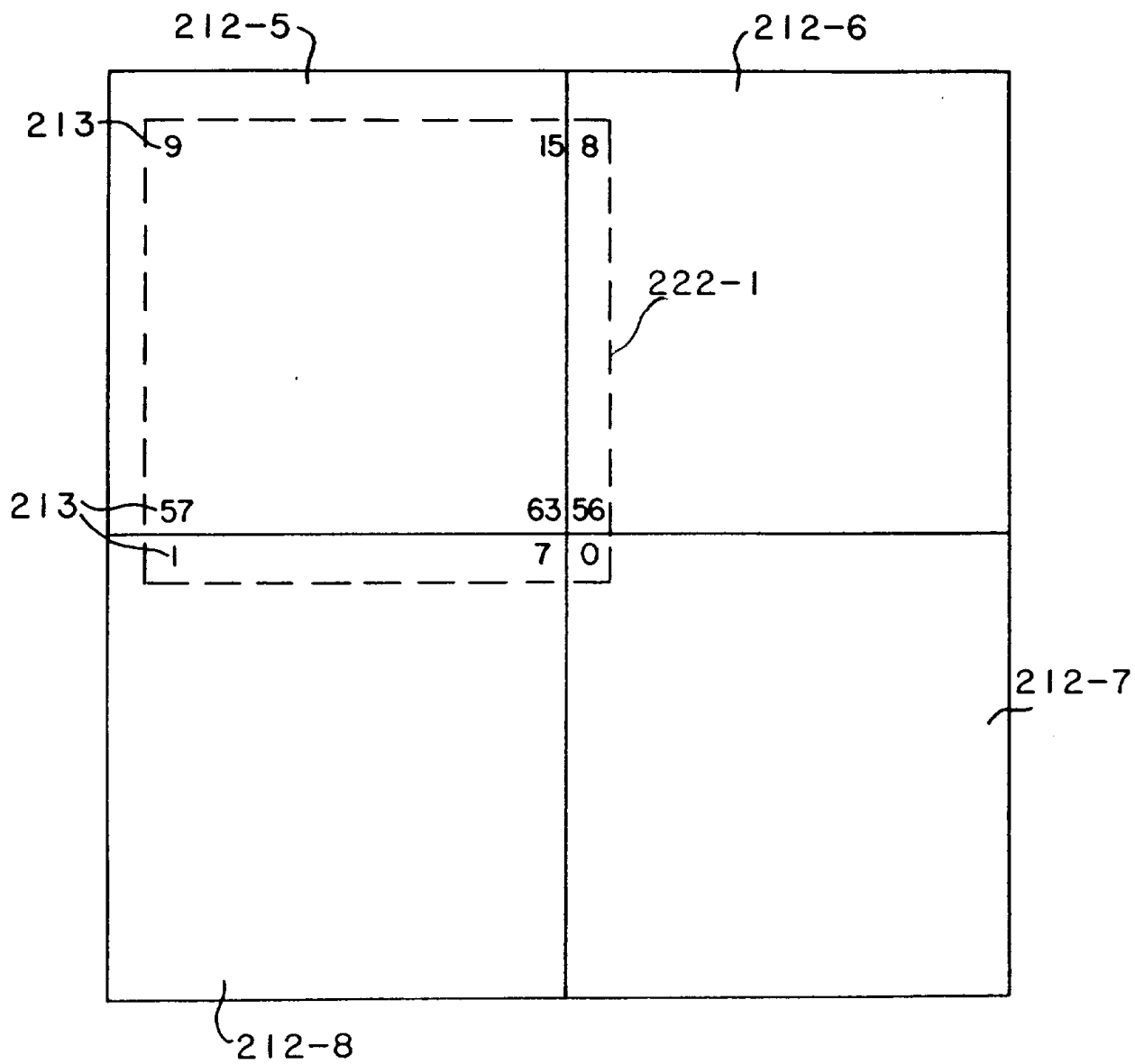


图8

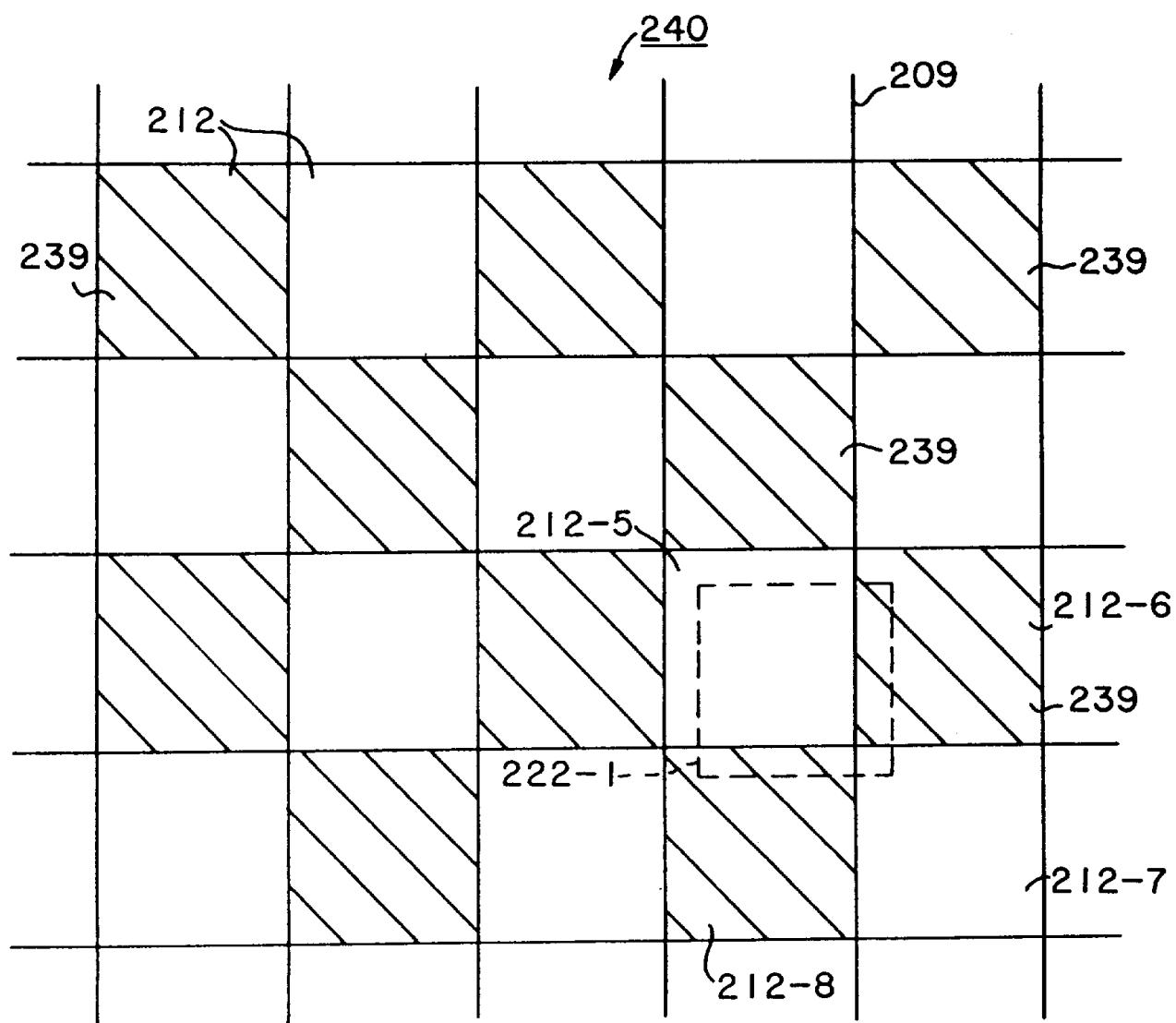


图9

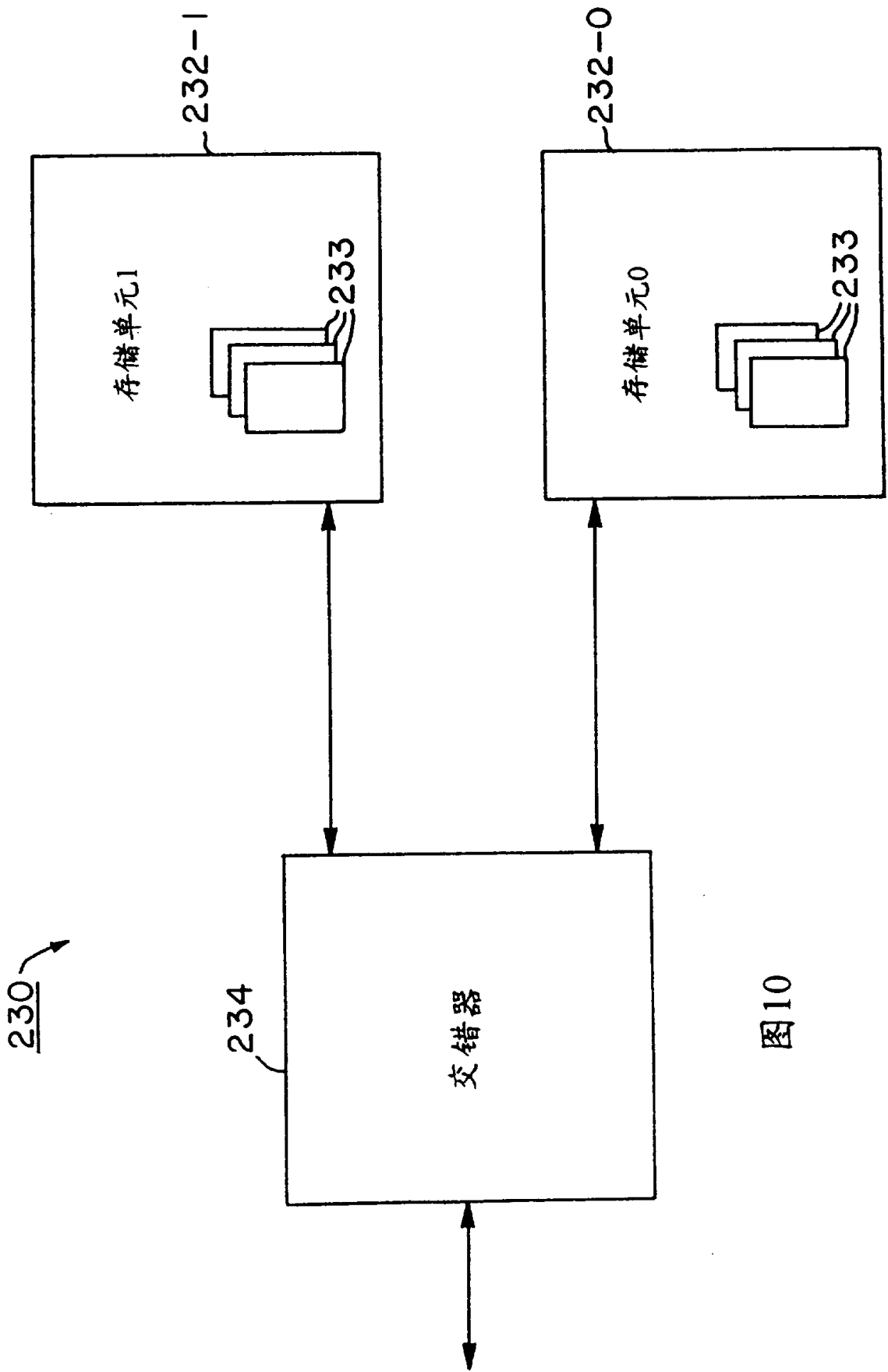


图10

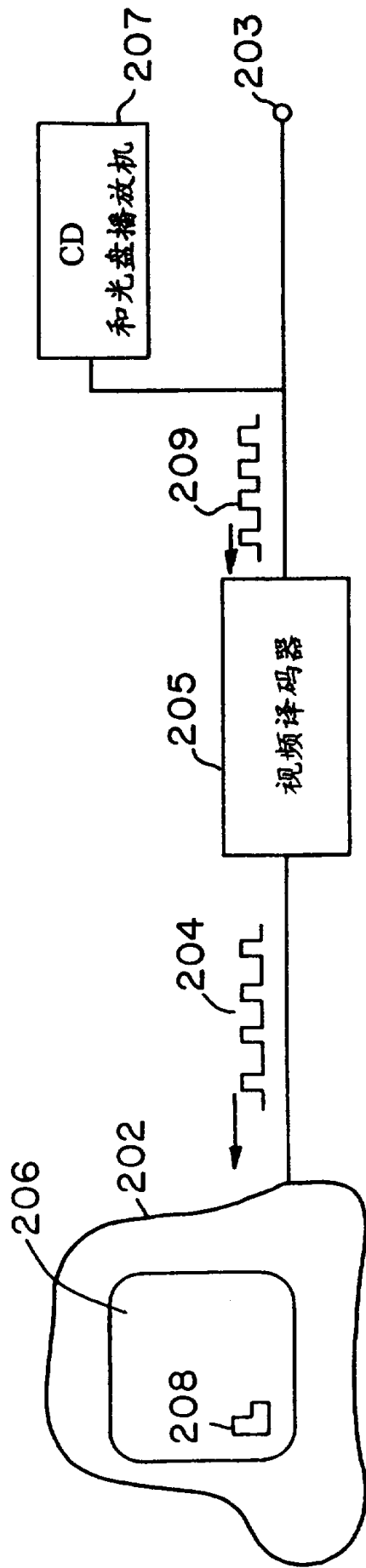


图11

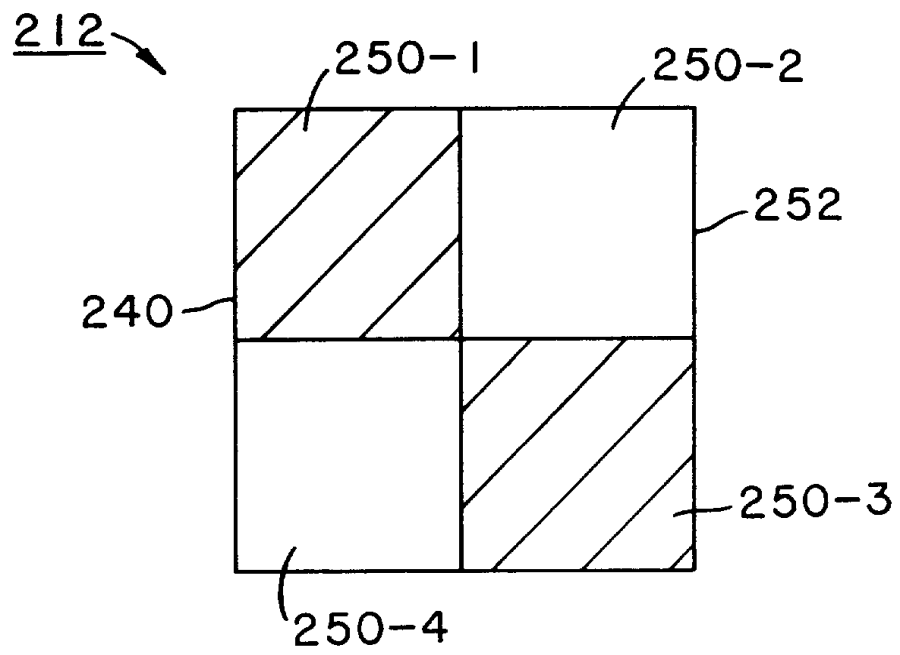


图12

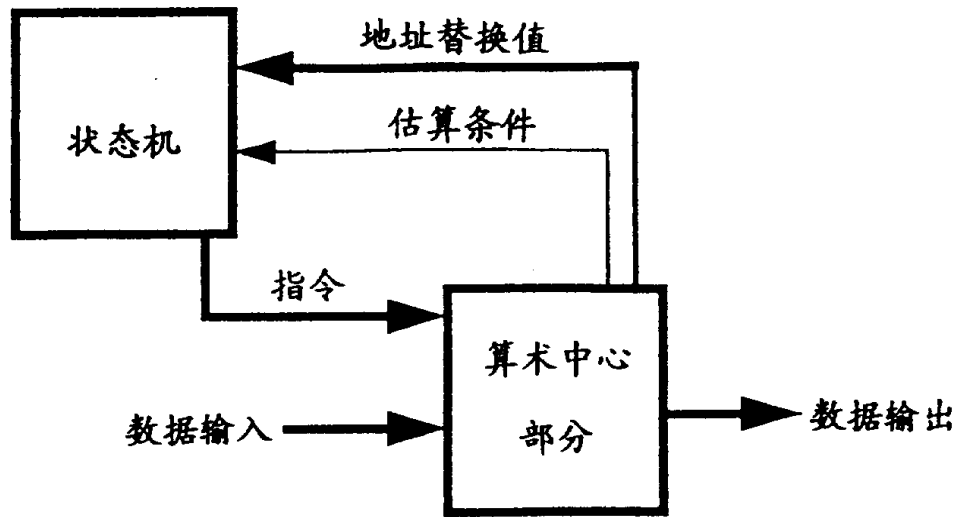
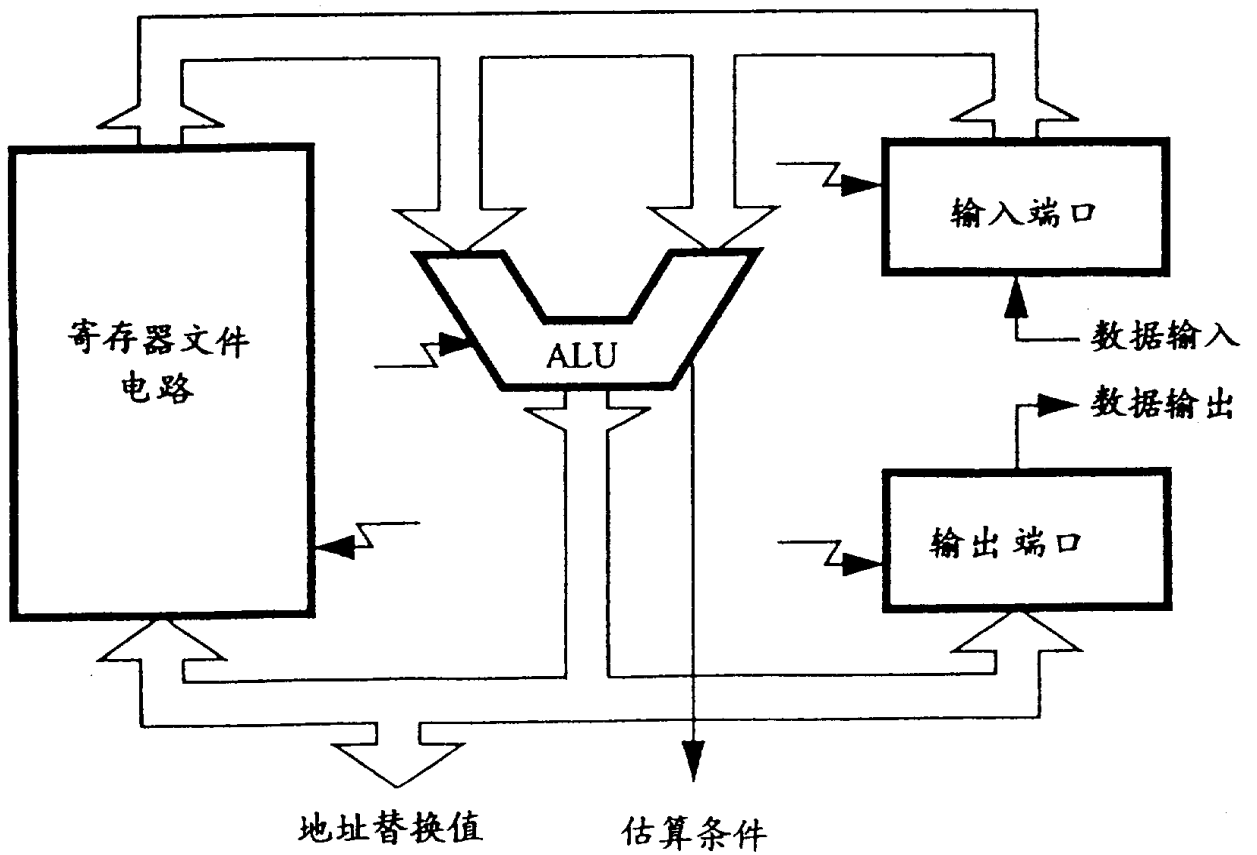


图13



表示微码控制

图14

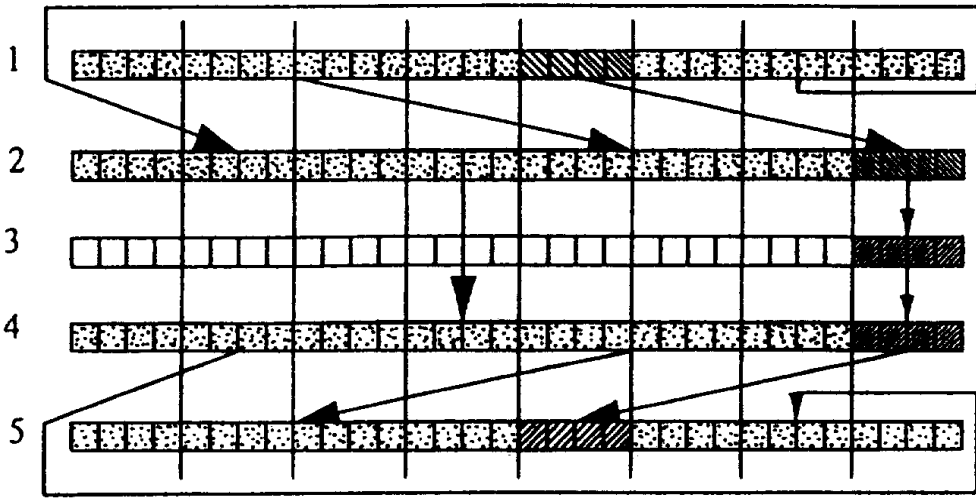


图15

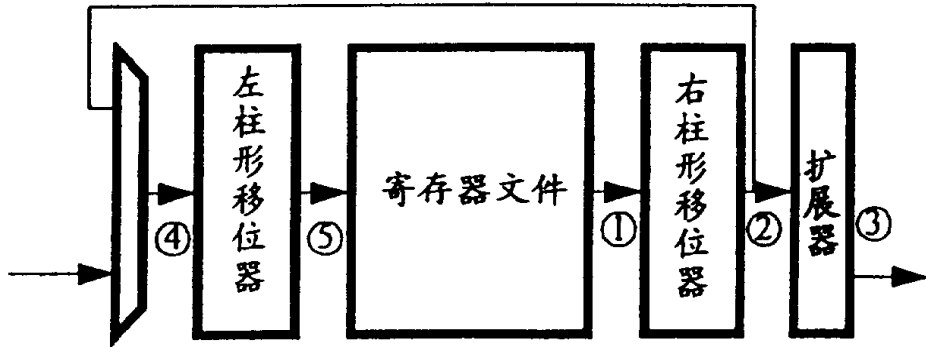


图16

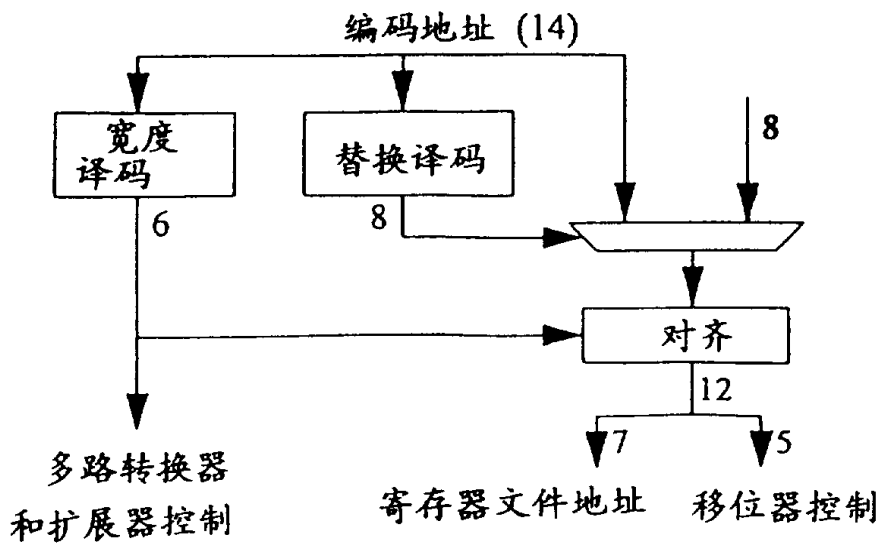


图17

用于寻址的定宽字		
宽度定义字段		地址字段
连续标记	终止标记	
uu.....uu	vv.....vv	aa.....aa

图18

用于寻址的固定宽字			
地址字段			替换指示符
替换字段			
	终止标记		连续标记
aa.....aa	yy.....yy	xx.....xx	ww.....ww

图19

用于寻址的固定宽字					
宽度定义字段		地址字段			替换指示符
连续标记	终止标记	替换字段			
					终止标记
uu.....uu	vv.....vv	aa.....aa	yy.....yy	xx.....xx	ww.....ww
000	1	1101	1	000	0
111	0	1101	0	111	1

图20

可变字段固定宽字		
第一字段	第二字段	
		终止标记
aa.....aa	yy.....yy	xx.....xx

图21

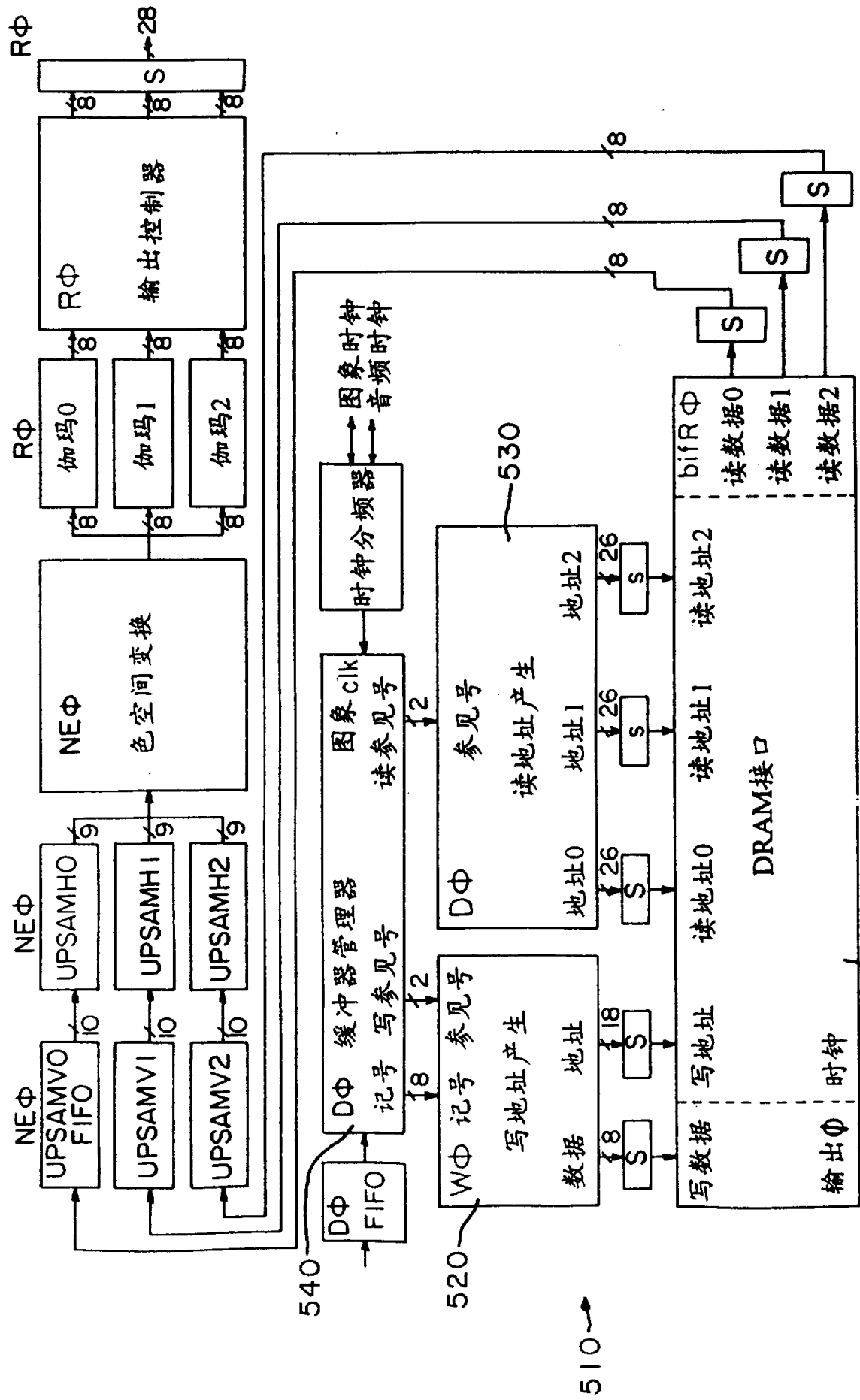


图22

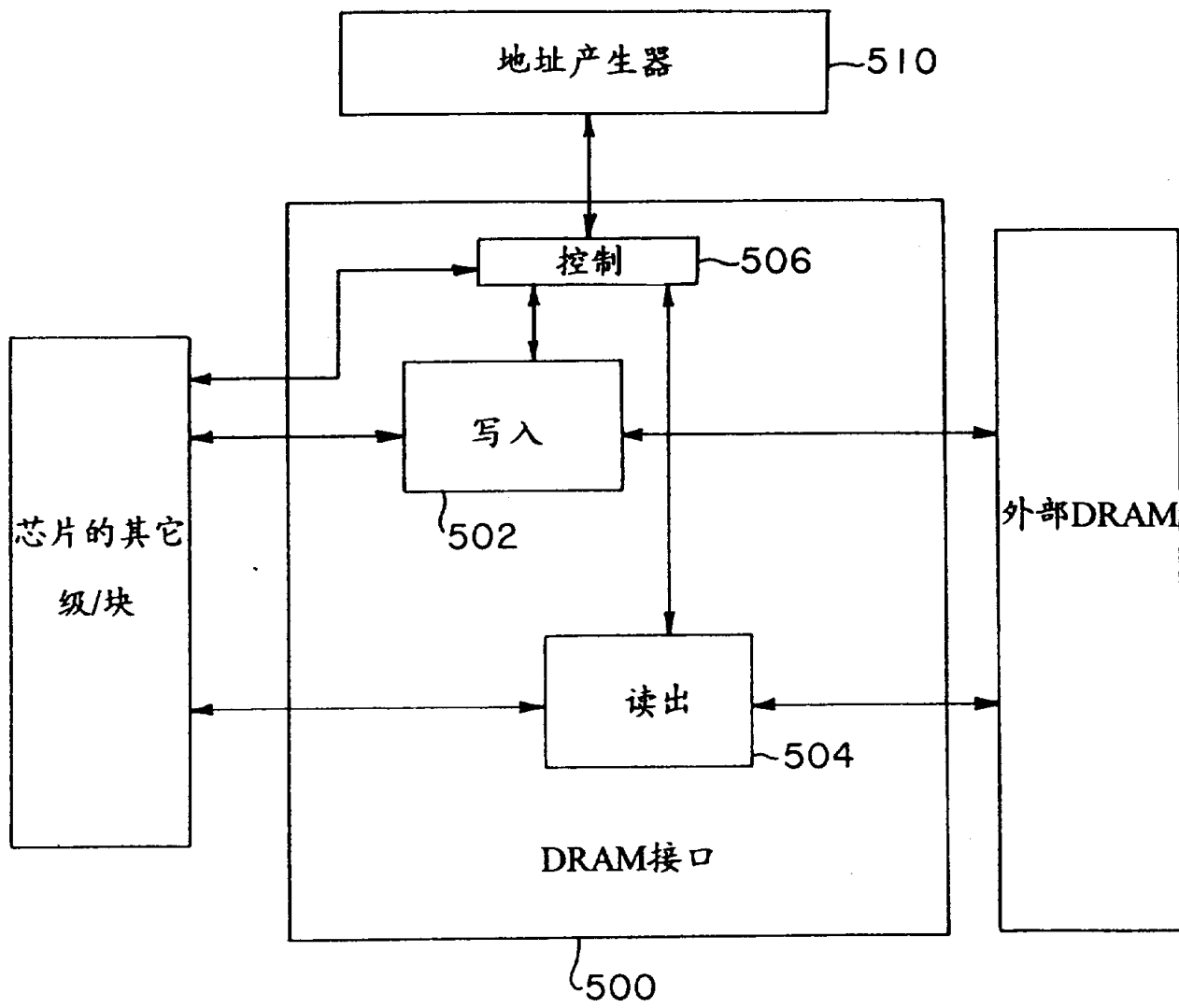


图23

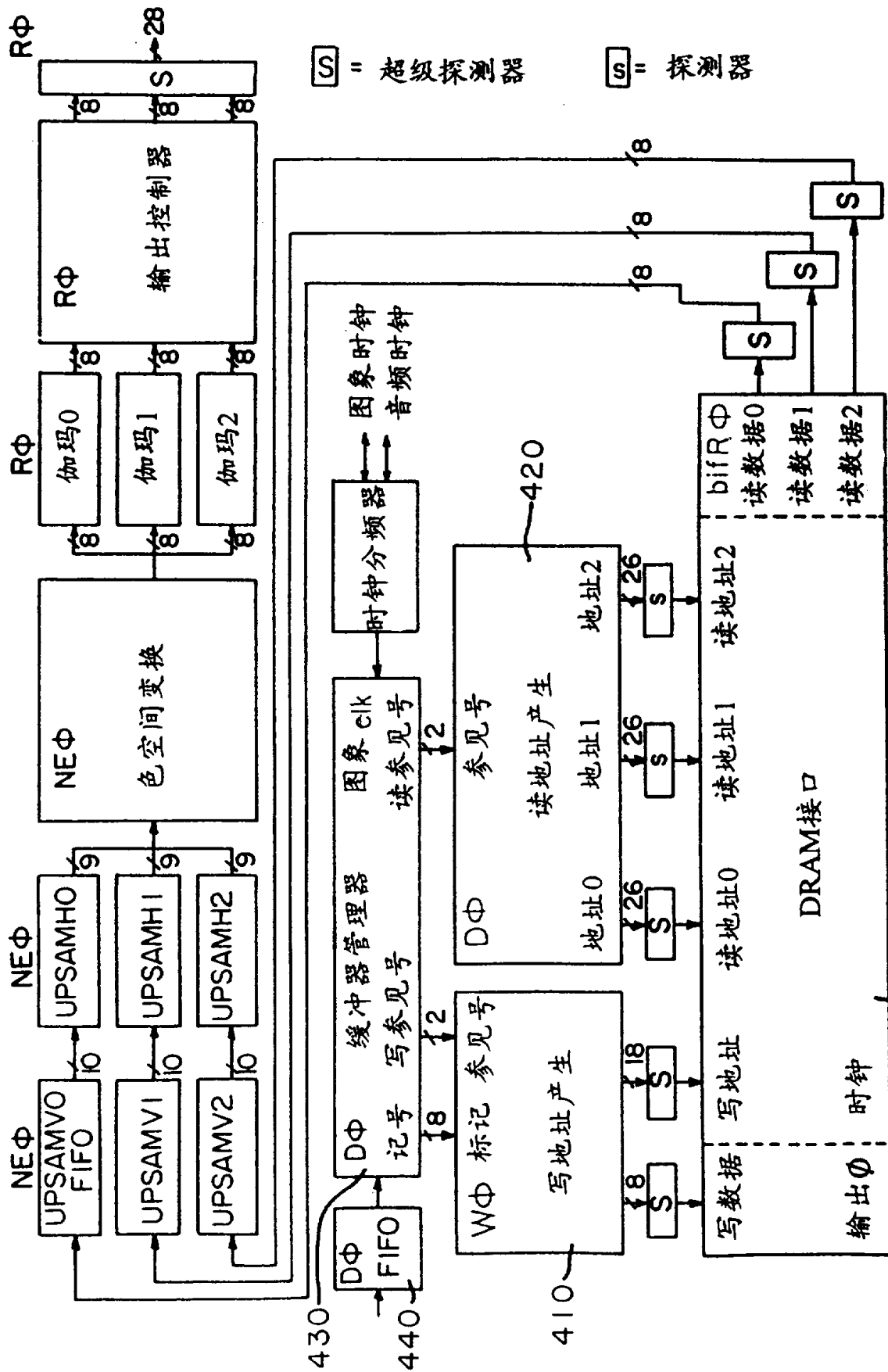


图24

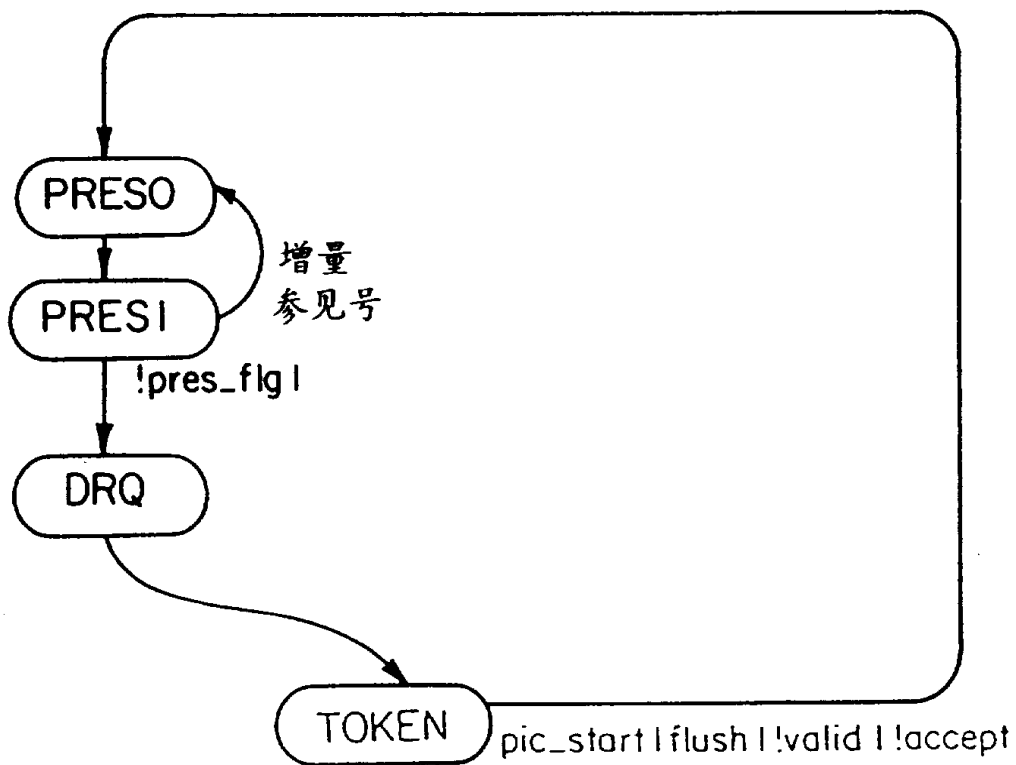


图26