

[19]中华人民共和国专利局

[51]Int.Cl.<sup>4</sup>

G06F 9 / 06



[12] 发明专利申请公开说明书

[11] CN 85 1 09307 A

CN 85 1 09307 A

[43]公开日 1986年11月26日

[21]申请号 85 1 09307

[22]申请日 85.12.26

[30]优先权

[32]85.5.28 [33]加拿大 [31]482,566

[71]申请人 米特尔公司

地址 加拿大安大略省卡拿达邮箱13089号

[72]发明人 博莱斯拉夫·西科拉

[74]专利代理机构 中国专利代理有限公司

代理人 匡少波

[54]发明名称 计算机存储器管理系统

[57]摘要

计算机存储器管理系统是用来检测联接在计算机系统含有多个存储器模块的预先确定的存储体上是否存在存储器芯片,接收和重新变换地址信号,而在响应时,随即启动预先确定的存储体。一个可编程的逻辑阵列启动存储器模块上的预先确定的存储体而发启动信号,以响应接收来自一个主控制器的地址和控制信号。这重新变换处理是在加电过程中完成的,因此,在以后的存储器访问过程中就无需等待状态。该系统是自适应和廉价的。

## 权 利 要 求 书

---

1、一个用于连接第一个存贮模块1和第二个存贮模块2的计算机存贮管理系统包括一个用于产生地址和控制信号的主控制器5，以及连接在该主控制器和第一个，第二个存贮模块1和模块2上的相应的可编程逻辑电路6A和6B，每一个存贮模块有一组存贮体 $B_0, B_1, \dots, B_n$ ，每个存贮模块可预先确定这些存贮体中有哪几个连接了一个或多个存贮电路，

本计算机管理系统的特征在于，上述的主控制器5检测存贮电路的存在，此电路连接上述的第一个存贮模块1的预先确定的存贮体 $B_0, B_1, B_2, \dots, B_n$ 并产生上述的第一个和第二个控制信号以作为响应，上述的可编程逻辑电路6A和6B分别接收到对应的第一个和第二个控制信号和上述产生的地址信号中预先确定的那些信号，并作为响应产生启动信号来触发启动上述第一个模块1和第二个模块2的预先确定的存贮体 $B_0, B_1, B_2, \dots, B_n$ ，使得当寻址该模块时，连接在上述第一个和第二个存贮模块上的存贮电路数量上的变化可自动补偿。

2、根据权利要求1所确定的计算机存贮管理系统，其特征在于加入用于周期地产生刷新信号的刷新控制电路21，22，23和24。

3、根据权利要求2所确定的计算机存贮管理系统，其特征在于，上述的可编程逻辑电路6A和6B中的任何一个都是一个可编程阵列逻辑电路9。

4、根据权利要求3所确定的计算机存贮管理系统，其特征在于，一级和二级的行和列编码电路的附加性能，每一编码电路包括：

(a) 一个连接在上述主控制器5的反相器45，用来接收从主控制器来的存贮器选择控制信号；

(b) 一个连接在上述反相器 4 5 上的延迟电路 4 6 用于延迟收到的存贮器选择控制信号；

(c) 第一与非门组 1 1—1 5 的输入端连接在上述的刷新控制电路和上述可编程阵列逻辑电路 9 的一个对应的预先确定的输出端；

(d) 一组非门 1 6—2 0 的输入端连接在上述可编程阵列逻辑电路 9 的一个对应的输出端，用来接收和反相上述的启动信号；

(e) 第二与非门组 2 5—2 9 的第一输入端组连接在上述第一与非门组 1 1—1 5 的输出端，第二输入端组与上述的反相器 4 5 连接，用来产生行选择信号以响应接收上述的存贮器选择信号和生成刷新或启动信号。

(f) 第三与非门组 3 0—3 4 的第一输入端组与上述的非门组 1 6—2 0 的输出端相连接，第二输入端组与上述延迟电路 4 6 的输出相连接，用来产生列选择信号，以响应接收延迟的上述存贮器选择信号和反相的启动信号，借助于这些行、列选择信号的产生，可启动上述存贮体中对应的行和列。

5、根据权利要求 1、2、3 或 4 所确定的计算机存贮器管理系统其特征在于，上述的可编程逻辑电路 6 A 和 6 B 都是一个可编程逻辑阵列。

6、一种用于一个或多个存贮器模块的一种计算机存贮器管理方法，该存贮器模块有一组存贮体  $B_0, B_1, B_2, \dots, B_n$ ，预先确定这些存贮体中的哪几个有一个或多个存贮电路与其连接，该管理方法的特征在于下列步骤：

(a) 产生地址信号和控制信号；

(b) 接收上述的第一组信号用响应存贮体的信号来检测是否存在

连接在预先确定存贮体的存贮电路；

(c) 接收上述的第二组信号并邻接启动相应的那些预先确定存贮体，以响应接收的第二组信号以及检测上述存贮电路，因此，连接在存贮模块上的存贮电路数目的变化，在对上述模块寻址时被自动补偿。

7、根据权利要求6所确定的方法中，其特征在于，对检测存贮电路存在的步骤可进而由下述几步骤组成：

(a) 产生一个或多个数据信号存贮在上述的存贮体里；

(b) 从上述预先确定的存贮体中检索存贮的数据信号，从上述的其它存贮体中检索缺席信号，从而进一步产生用于表明上述被检测的存贮电路存在的控制信号。

8、根据权利要求6所确定的适用于主存贮模块1和扩展存贮模块2的一种计算机存贮管理方法，每个存贮模块都有一组存贮体，并预先确定这组存贮体中的哪几个连接有一个或多个存贮电路，该管理方法的特征在于进而具有下列步骤：

(a) 产生第一个地址信号加给连接在上述主存贮模块1上的第一可编程逻辑电路6A，并作为响应依次启动主存贮模块1的存贮体 $B_0$ ， $B_1$ ， $B_2$ ，…… $B_n$ ；

(b) 检测上述主存贮模块1的被启动的存贮体中，那些预先确定的存贮体上是否连接有存贮电路；

(c) 生成和存贮第一和第二个控制信号，表示存贮电路连接在上述主存贮模块1中预先确定的存贮体上；

(d) 把上述的第一个控制信号加给第一可编程逻辑电路6A；

(e) 把上述的第二控制信号加给连接在该扩展存贮模块2上的第二个可编程逻辑电路6B；

(f) 进一步产生地址信号加给上述的第一个和第二个可编程逻辑电路 6 A 和 6 B，并邻接地启动上述主存贮模块 1 和扩展存贮模块 2 中那些预先确定的存贮体以作为响应，因此，连接在上述主和扩展存贮模块上的存贮电路的数目变化在对这些模块寻址时自动补偿。

## 计算机存贮器管理系统

本发明通常与计算机系统有关，特别是与存贮器管理系统有关。这管理系统是用来检测一个计算机系统中是否存在存贮器电路，在响应时重新变换存贮器地址的。

计算机系统通常由一个或多个存贮器模块组成，每个模块由多个存贮体组成，而每个存贮体上连接着存贮电路，例如“动态随机访问存贮器”(DRAMS)。一个主控制器(例如一个CPU)一般生成地址信号，用以访问存贮器电路的预先确定的存贮器单元。例如，一个地址信号通常由存贮模块接收并在其中译码，使之启动一个预定的存贮体，并访问一个由地址信号所指定的DRAM存贮器单元。

在一个固定的(不可扩展的)存贮空间的简单的计算机系统中，通常在一个已生成的地址信号和DRAM存贮单元之间存在一个——对应的关系。

可以期望的是存贮体的数目是可以扩展的，这为实现软件的修改或存贮大量的数据等功能，提供大量的存贮器电路。在过去，为了进行存贮器的扩展，通常需要对昂贵的存贮器模块地址译码电路进行重新设计和重新布线。

为了克服昂贵的重新设计和重新布线这一缺点，存贮器管理系统先前的技巧是利用了多个硬件地址多路转换器和一个包括了在应用程序执行过程中用来进行地址重新变换的子程序操作系统。这个重新变换处理被看作是“联机的”，这个经过重新变换的地址信号用来对实际的DRAM存贮单元进行访问。前面提到的重新变换子程序和附加的硬件是复杂的，而且人们发现该子程序的执行是消耗时

间的。典型地，为了完成地址重新变换需要一个或多个主控制器处于等待状态，这是因为有多条存贮器切换访问指令插在程序编码中。这样做，实际上既降低了应用程序的运行速度，又增加了存贮器访问出错的可能性。

该重新变换子程序是专用的，或是不能自适应的。因为若进一步对存贮器进行扩展（或减少），需要对程序的各种参数进行修改，这通常也需要替换存放操作系统的软盘或只读存贮器（ROM）。

根据目前的发明，一个廉价的、自适应的存贮管理系统提供了用来检测在一个计算机系统中的多个存贮器模块上，预先确定的存贮体上是否连接着存贮器电路，在一个初始化过程中，主控制器产生重新变换地址信号，接着，产生一个启动信号，紧接着启动预定的存贮器。

根据一个择优方案，该启动信号电路由可编程的逻辑电路组成，该逻辑电路为启动一个预定的存贮体而产生前面讲到的那种启动信号。在每个程序的访问过程中没有任何处于等待状态方式中的那种时间消耗。

一个典型的计算机系统可以有二个存贮器模块，即主存模块和扩展的存贮模块，每个模块容纳高达2.5兆字节的存贮（每个模块使用5个512 K字节的存贮体），主存模块与扩展模块的存贮器地址必须是紧连的。因此，扩展存贮模块的第一个存贮单元地址应该是280000（十六进制）。然而，如果主存模块没有达到最大限度（2.5MB）而实际上仅有1 MB的DRAM芯片连接在那里（即5个存贮体中只有两个是可填充的）那么，在主存模块中，最后一个有效的存贮单元地址应是0FFFFF（十六进制）且下接相邻的地址应该是100000（十六进制）。

根据目前的发明，在主存模块中应该仅检测出 1 兆字节的存储器，并且，预置的控制信号被传递给启动信号电路。如果主控制器产生一个 100000(十六进制)地址信号，那么在扩展模块上，启动信号电路把从 280000(十六进制)开始的存储器地址重新变换到一个 100000(十六进制)的地址上，同时，主存模块上的启动信号电路闭锁从 100000(十六进制)以后的存储器地址空间，因而使得对扩展存储模块上的第一个单元的访问而不会在主存储模块上的单元地址 100000 上进行。而，主存储模块的第二个存储体是与扩展存储模块的第一个存储体相邻接的。

该系统是廉价的，存储器检测处理最好是在第一个存储器访问前执行(即在加电时)，如前述的技巧——“联机”技术一样，在以后的存储器访问过程中，不需要等待状态。

如果主存储器模块的后续体是被填充的(即第三，第四或第五个体等等)，在初始化阶段或加电时，该存储器管理系统测试被扩展的存储体是否存在，并且重新变换扩展存储模块的第一个存储单元地址，使得与已扩充的主存储模块的最后一个有效地址相邻接。因此，该系统是自适应的，并且克服了先前技巧中的缺点——需要通过对软盘或只读存储器的替换或大量的重新设计和重新布线，来修改操作系统程序。

通常，本发明是用于连接多个存储器模块的一个计算机存储器管理系统，每个存储器模块有一组存储体，并预先确定哪些存储体上有一个或多个存储器电路连接在上面。本发明由以下几部分组成：用于生成地址信号的电路；用于接收第一批地址信号，作为响应检测连接在预定的存储体上的存储器电路是否存在的电路技术；用于连接和重

新变换下一批地址信号，作为响应而邻接地启动预定的存贮体这样的电路技术。在对上述存贮模块进行寻找时，接到前述的多个存贮器模块上的存贮器电路的数量变化是自动进行补偿的。

更具特色的是：本发明是一个用于连接多个存贮器的计算机存贮器管理系统，每个存贮器有多个存贮体，并已预定它们有一个或多个存贮器电路连到那里。本发明由以下电路组成：一个生成地址信号的电路，用于接收已预定的第一批地址信号，并作为响应与启动一组存贮体中后继的一些存贮体而生成的启动信号的电路；一个用于检测连接到预定的上述已成功启动了的一个存贮体上的存贮电路的存在与否，并且作为响应而生成一个或多个控制信号的电路；用于接收该控制信号和下一个地址信号，作为响应，重新变换已接收的后继地址信号并邻接地启动预定存贮体的电路。

本发明也是一个计算机存贮器管理的方法，它用于连接一个或多个存贮模块，各个存贮模块有多个存贮体，其中预先确定的一组上连接有一个或多个存贮电路。本方法包括生成地址和控制信号等步骤，接收第一组地址和控制信号，作为响应，检测连接到预定存贮体上的存贮器电路，而接收第二组地址和控制信号，并邻接地启动相应的一组预定存贮体以作为对接收的第二组地址和控制信号的响应，然后检测存贮器电路是否存在。

如要对本发明更好的了解，可参考下列相应图解和随后的详细说明。

图 1 是依据现发明的一个存贮器管理系统的框图。

图 2 是依据现发明的一个择优方案的启动信号生成电路的示意图。

参看图 1，一个主存贮模块 1 和一个扩展存贮模块 2 都是由多个

存贮体  $B_0, B_1, B_2, \dots, B_N$  所组成。每个存贮模块通常由一个或多个 DRAM 芯片 (例如, 64K 或 256 K 的 DRAM 芯片) 所组成。存贮体的各动态随机访问存贮器的数据端接到数据总线 3 上, 存贮体上的地址输入端连接到一个连在主控制器 5 上的地址总线 4 的  $A_1 \sim A_i$  地址线上, 主控制器类似于微处理器。

第一级和第二级启动信号生成电路 6 A 和 6 B 的启动输出端分别连到主存贮模块 1 和扩展存贮模块 2 各自相应的存贮体  $B_0, B_1, B_2, \dots, B_N$  的输入端上。启动信号电路上的第一级多个地址输入端被连接到地址总线 4 的地址  $A_1 \sim A_j$  上, 并且, 第二级多个输入端被连到一个连在主控制器 5 上的控制总线 7 上。此外, 在地址总线 4 和控制总线 7 信号分别控制下, 一个从主控制器 5 输出的 RSTVEC 信号被连到启动信号电路 6 A 的另一个输入端上, 并通过锁存器 8 接到启动信号电路 6 B 的另一个输入端上。

在运行时, 为了使用启动信号电路 6 A, 主控制器 5 在地址总线 4 和控制总线 7 以及它自己的输出端 RSTVEC 上生成预定地址信号和控制信号。作为响应, 电路 6 A 在软件控制下连续地启动主存贮模块 1 的存贮体  $B_0, B_1, B_2, \dots, B_N$ , 并且主控制器 5 检测连接在那里的存贮器芯片是否存在。

例如, 存贮器  $B_0$  由启动信号电路 6 A 来启动, 并且主控制器 5 在数据总线 3 上生成一个数据信号, 存贮在存贮体  $B_0$  的一个预先确定的存贮单元中, 然后, 主控制器 5 读出这个存贮单元中的数据信号, 如果存贮体  $B_0$  上有 DRAM 芯片连接, 则主控制器 5 读的是先前存贮在其中的数据信号。然而, 如果存贮体  $B_0$  上没有 DRAM 芯片连接, 则主控制器 5 就会由于访问了不存在 DRAM 芯片而读到一个缺席信号, 如 FF (十六进制), 在检测 DRAM 芯片的过程中, 存贮体  $B_0$  保持已启动状

态，启动存贮体  $B_1$ ，反复对 DRAM 芯片作检测处理。

对空存贮体（即没有连接 DRAM 芯片的存贮体）检测时，主控制器 5 闭锁这个空存贮体，并在启动信号电路中存放预定的控制信号，表明具有 DRAM 芯片的主存贮模块上的存贮体  $B_0, B_1, B_2, \dots, B_N$  的号码。这样，扩展存贮模块 2 的第一个有效地址被重新变换，以便与主存贮模块 1 的最后一个有效地址相邻接。对图 2 将在下面更详细地说明。

因此，在主控制器 5 执行一个程序的过程中，存贮器地址由电路 6 A 和 6 B 进行重新变换，以致分别地启动主存贮模块 1 和扩展存贮模块 2 的邻接存贮器单元。

RSTVEC 信号是一个复位信号，当系统恢复时允许对 ROM（只读存贮器）进行向量存取，它闭锁所有的存贮模块。为了解决在重新变换处理中对数据总线的竞争，经过锁存器 8，有选择地启动各主、扩展存贮模块。

参看图 2，这里给出了启动信号 6 A 或 6 B 的简图，根据本发明的择优方案，存贮模块 1 和 2 都是由表示为  $B_0 \sim B_4$  的 5 个存贮体组成，一个可编程的阵列逻辑（PAL）9 的 ADDRESS 输入端与地址总线 4 的地址线  $A_{17} \sim A_{22}$  相连接，地址线  $A_{17} \sim A_{22}$  又与图 1 中的地址线  $A_1 \sim A_5$  相对应。图 1 的数据总线 3 的数据线  $D_0 \sim D_4$  被分别连接到数据寄存器 10 的相应的数据输入端  $1D \sim 5D$  上。寄存器 10 的复位输入端连到控制总线 7 的控制线 RESET 上。寄存器 10 的输出端  $1Q, 2Q, 3Q, 4Q$  和  $5Q$  连到可编程的阵列逻辑 PAL9 的控制输入端 CTRL 上，并且相应地产生控制信号 A、B、C、D 和 X。这些将在下面更详细地论述。用 RSTVEC 表示的 PAL9 的另一个控制输

入端按上面讨论如图 1 所示的那样，是与主控制器 5 的 RSTVEC 输出端相连接的，在启动信号电路 6 B 中，PAL 9 的输入端 RSTVEC 连接到与主控制器 5 的输出端 RSTVEC 相连接的锁存器 8 的输出端。

PAL 9 的输出端 RCAS0, RCAS1, RCAS2, RCAS3 和 RCAS4 被相应地连接到“与非门”11, 12, 13, 14 和 15 的第一组输入端上，并且相应地接到“非门”16, 17, 18, 19 和 20 上。“与非门”11~15 的第二组输入端和“与非门”21 的输出端相接。

“与非门”21 的第一个输入端与“非门”22 的一个输出端相连，“非门”22 的输入端与连在控制器 5 的控制总线 7 的控制线 REFRESH 相连接。“与非门”21 的第二个输入端与串联的“非门”23 和 24 的输出端相连，“非门”24 的输入端连到主控制器 5 的地址线 A23 上。并且，“非门”24 的输出端连到“非门”23 的输入端上。

“与非门”11~15 的输出端相应地连到“与非门”25~29 的第一组输入端上，并且，非门 16~20 的输出端相应地连到“与非门”30~34 的第一组输入端上。“与非门”25~34 的输出端相应地连到电阻器 35~44 的输入端上。

输出电阻器 35~39 相应地连到行地址选择输出端 RAS0~RAS4 上，同时，输出电阻器 40~44 相应地连到列地址选择点 CAS0~CAS4 上。行、列地址输出端 RAS0~RAS4 与 CAS0~CAS4 连到主存贮模块 1 的存贮体 B0~B4 的相应的行和列启动输入端上。

从主控制器 5 来的一个地址选择输出端 AS 接到“非门”45 上，而“非门”45 的输出端接到“与非门”25~29 的第二组输入端

和延时线 46 的一个输入端 (IN) 上。延迟线 46 的一个输出端 (OUT) 连到 “与非门” 30~34 的第二组输入端上。

在运行时, 首先检测主存贮模块 1 的存贮体 B<sub>0</sub>~B<sub>4</sub> 的相邻存贮空间 (见图 1), P A L 9 为响应接收到其地址和控制输出 (ADDRESS 和 CTAL) 端上的预定信号, 而在 RCAS<sub>0</sub>~RCAS<sub>4</sub> 的预定输出端上生成有源低信号。象大家所熟悉的编程一样, 这结果是根据本表 1 和表 2 中列出的方程式得来的。

根据择优方案, 正如上面所讨论的存贮模块 1 和 2 (图 1) 都是由 5 个包含 64 K 或 256 K 字节的 DRAM 芯片的存贮体 B<sub>0</sub>~B<sub>4</sub> 组成的, 如果主存贮模块 1 是由 64 K 的芯片组成, 它可容纳达 640 K 字节的数据。如果主存模块是由 256 K 的芯片组成, 它可容纳达 2.5 兆字节的数据, 同样地, 扩展存贮模块 2 也将由 64 K 或 256 K 芯片组成, 但二者不同时使用。

用在 RCAS<sub>0</sub>~RCAS<sub>4</sub> 输出端上生成有源低启动信号和启动预定存贮体的 P A L 方程式 (逻辑表达式) 在表 1 和表 2 中给出:

表 1 64K 芯片的 PAL 方程式

|                                |                     |                     |                |           |          |    |          |    |    |
|--------------------------------|---------------------|---------------------|----------------|-----------|----------|----|----------|----|----|
| <u>RCAS0*</u>                  |                     |                     |                |           |          |    |          |    |    |
| /A22*/A21*/A20*/A19*/A18*/A17* | RSTVEC              | *                   | /X             | $\infty$  |          |    |          |    |    |
| /A22*/A21*/A20*                | A19*/A18*           | A17*                | RSTVEC         | *         | X        | *  | /C       | *  | D  |
| /A22*/A21*                     | A20*/A19*/A18*/A17* | RSTVEC              | *              | X         | *        | C  | *        | /D | *  |
| /A22*/A21*                     | A20*                | A19*/A18*/A17*      | RSTVEC         | *         | X        | *  | C        | *  | /D |
| /A22*                          | A21*                | A20*/A19*/A18*/A17* | RSTVEC         | *         | X        | *  | C        | *  | /D |
| /A22*                          | A21*                | A20*                | A19*/A18*/A17* | RSTVEC    | *        | X  | *        | C  | *  |
| /A22*                          | A21*                | A20*                | A19*           | A18*/A17* | RSTVEC   | *  | X        | *  | C  |
| /A22*                          | A21*                | A20*                | A19*/A18*/A17* | RSTVEC    | *        | X  | *        | C  | *  |
| <u>RCAS1*</u>                  |                     |                     |                |           |          |    |          |    |    |
| /A22*/A21*/A20*/A19*/A18*      | A17*                | RSTVEC              | *              | /X        | *        | A  | *        |    |    |
| /A22*/A21*/A20*                | A19*/A18*/A17*      | RSTVEC              | *              | X         | *        | /C | *        | /D | *  |
| /A22*/A21*                     | A20*/A19*/A18*      | A17*                | RSTVEC         | *         | X        | *  | C        | *  | /D |
| /A22*/A21*                     | A20*                | A19*/A18*           | A17*           | RSTVEC    | *        | X  | *        | C  | *  |
| /A22*                          | A21*/A20*/A19*/A18* | A17*                | RSTVEC         | *         | X        | *  | C        | *  | /D |
| /A22*                          | A21*                | A20*                | A19*/A18*      | A17*      | RSTVEC   | *  | X        | *  | C  |
| /A22*                          | A21*                | A20*                | A19*           | A18*/A17* | RSTVEC   | *  | X        | *  | C  |
| /A22*                          | A21*                | A20*                | A19*/A18*/A17* | RSTVEC    | *        | X  | *        | C  | *  |
| <u>RCAS2</u>                   |                     |                     |                |           |          |    |          |    |    |
| /A22*/A21*/A20*/A19*           | A18*/A17*           | RSTVEC              | *              | /X        | $\infty$ | A  | $\infty$ | B  | +  |
| /A22*/A21*/A20*                | A19*                | A18*                | A17*           | RSTVEC    | *        | X  | *        | /C | *  |

A  $\infty$  D +

/A22\*/A21\* A20\*/A19\* A18\*/A17\* RSTVEC \* X \* C \*/B \*/A ∞ D +  
 /A22\*/A21\* A20\* A19\* A18\*/A17\* RSTVEC \* X \* C \*/B \*/A ∞ D +  
 /A22\* A21\*/A20\*/A19\* A18\*/A17\* RSTVEC \* X \* C \*/B \*/A ∞ D +  
 /A22\* A21\*/A20\* A19\* A18\*/A17\* RSTVEC \* X \* C \*/B \*/A ∞ D +  
 /A22\*/A21\*/A20\* A19\* A18\*/A17\* RSTVEC \* X \* C \*/B \*/A ∞ D

RCAS3 \*

/A22\*/A21\*/A20\*/A19\* A18\* A17\* RSTVEC \*/X ∞ A ∞ B ∞ C +  
 /A22\*/A21\*/A20\*/A19\*/A18\*/A17\* RSTVEC \* X \* C \*/B \*/A ∞ D +  
 /A22\*/A21\* A20\*/A19\* A18\* A17\* RSTVEC \* X \* C \*/B \*/A ∞ D +  
 /A22\*/A21\* A20\* A19\* A18\* A17\* RSTVEC \* X \* C \*/B \*/A ∞ D +  
 /A22\* A21\*/A20\*/A19\* A18\* A17\* RSTVEC \* X \* C \*/B \*/A ∞ D +  
 /A22\* A21\*/A20\* A19\* A18\* A17\* RSTVEC \* X \* C \*/B \*/A ∞ D +  
 /A22\*/A21\*/A20\* A19\*\*A18\* A17\* RSTVEC \* X \* C \*/B \*/A ∞ D

RCAS4 \*

/A22\*/A21\*/A20\* A19\*/A18\*/A17\* RSTVEC \*/X \* A ∞ B ∞ C ∞  
 /A22\*/A21\* A20\*/A19\*/A18\* A17\* RSTVEC \* X \* C \*/D \*/A ∞ D +  
 /A22\*/A21\* A20\* A19 /A18\*/A17\* RSTVEC \* X \* C \*/D \*/A ∞ D +  
 /A22\* A21\*/A20\*/A19\*/A18\*/A17\* RSTVEC \* X \* C \*/D \*/A ∞ D +  
 /A22\* A21\*/A20\* A19\*/A18\*/A17\* RSTVEC \* X \* C \*/D \*/A ∞ D +  
 /A22\* A21\* A20\*/A19\*/A18\*/A17\* RSTVEC \* X \* C \*/D \*/A ∞ D +  
 /A22\*/A21\* A20\*/A19\*/A18\*/A17\* RSTVEC \* X \* C \*/D \*/A ∞ D

表 2 256 K 芯片的 PAL 方程式

|                     |   |          |         |          |             |         |  |  |  |
|---------------------|---|----------|---------|----------|-------------|---------|--|--|--|
| <u>RCAS0 *</u>      |   |          |         |          |             |         |  |  |  |
| /A22*/A21*/A20*/A19 | * | RSTVEC*  | /X      | *        |             |         |  |  |  |
| /A22*/A21*/A20* A19 | * | /A18 *   | A17 *   | RSTVEC * | X */C */D * | A +     |  |  |  |
| /A22*/A21* A20*/A19 | * | RSTVEC * | X *     | C */D *  | /A +        |         |  |  |  |
| /A22*/A21* A20* A19 | * | RSTVEC * | X *     | C */B *  | A +         |         |  |  |  |
| /A22* A21*/A20*/A19 | * | RSTVEC * | X *     | C *      | B */A +     |         |  |  |  |
| /A22* A21*/A20* A19 | * | RSTVEC * | X *     | C *      | B *         | A +     |  |  |  |
| /A22*/A21*/A20* A19 | * | RSTVEC * | X */C * | B *      | A           |         |  |  |  |
| <u>RCAS1 *</u>      |   |          |         |          |             |         |  |  |  |
| /A22*/A21*/A20* A19 | * | RSTVEC * | /X *    | A +      |             |         |  |  |  |
| /A22*/A21* A20*/A19 | * | /A18 *   | A17 *   | RSTVEC * | X */C */B * | A ∞ D + |  |  |  |
| /A22*/A21* A20* A19 | * | RSTVEC * | X *     | C */B *  | /A ∞ D +    |         |  |  |  |
| /A22* A21*/A20*/A19 | * | RSTVEC * | X *     | C */B *  | A ∞ D +     |         |  |  |  |
| /A22* A21*/A20* A19 | * | RSTVEC * | X *     | C *      | B */A ∞ D + |         |  |  |  |
| /A22* A21* A20*/A19 | * | RSTVEC * | X *     | C *      | B *         | A ∞ D + |  |  |  |
| /A22*/A21* A20*/A19 | * | RSTVEC * | X */C * | B *      | A ∞ D       |         |  |  |  |
| <u>RCAS2 *</u>      |   |          |         |          |             |         |  |  |  |
| /A22*/A21* A20*/A19 | * | RSTVEC * | /X *    | A *      | B +         |         |  |  |  |
| /A22*/A21* A20* A19 | * | /A18 *   | A17 *   | RSTVEC * | X */C */B * | A ∞ D + |  |  |  |
| /A22* A21*/A20*/A19 | * | RSTVEC * | X *     | C */B *  | /A ∞ D +    |         |  |  |  |

|                    |               |            |        |        |        |     |        |        |     |         |         |    |   |   |
|--------------------|---------------|------------|--------|--------|--------|-----|--------|--------|-----|---------|---------|----|---|---|
| /A22*              | A21*/A20*     | A19        | *      | RSTVEC | *      | X   | *      | C      | */B | *       | A       | ∞  | D | + |
| /A22*              | A21*          | A20*/A19   | *      | RSTVEC | *      | X   | *      | C      | */B | *       | /A      | ∞  | D | + |
| /A22*              | A21*          | A20*       | A19    | *      | RSTVEC | *   | X      | *      | C   | */B     | *       | A  | ∞ | D |
| /A22*              | A21*          | A20*       | A19    | *      | RSTVEC | *   | X      | *      | /C  | */B     | *       | A  | ∞ | D |
| <u>RCAS3</u> *     |               |            |        |        |        |     |        |        |     |         |         |    |   |   |
| /A22*              | A21*          | A20*       | A19    | *      | RSTVEC | *   | /X     | *      | A   | ∞       | B       | ∞  | C | + |
| /A22*              | A21*          | A20*/A19** | /A18   | *      | A17    | *   | RSTVEC | *      | X   | */C*/D* | A       | ∞  | D | + |
| /A22*              | A21*/A20*     | A19        | *      | RSTVEC | *      | X   | *      | C      | */B | *       | /A      | *  | D | + |
| /A22*              | A21*          | A20*/A19   | *      | RSTVEC | *      | X   | *      | C      | */B | *       | A       | ∞  | D | + |
| /A22*              | A21*          | A20*       | A19    | *      | RSTVEC | *   | X      | *      | C   | */B     | *       | /A | ∞ | D |
| A22*/A21*/A20*/A19 | *             | RSTVEC     | *      | X      | *      | C   | */B    | *      | A   | ∞       | D       | +  |   |   |
| /A22*              | A21*/A20*/A19 | *          | RSTVEC | *      | X      | *   | /C     | */B    | *   | A       | ∞       | D  |   |   |
| <u>RCAS4</u> *     |               |            |        |        |        |     |        |        |     |         |         |    |   |   |
| /A22*              | A21*/A20*/A19 | *          | RSTVEC | *      | /X     | *   | A      | *      | B   | *       | C       | *  | D | + |
| /A22*              | A21*/A20*     | A19        | *      | /A18   | *      | A17 | *      | RSTVEC | *   | X       | */C*/D* | A  | ∞ | D |
| /A22*              | A21*          | A20*/A19   | *      | RSTVEC | *      | X   | *      | C      | */B | *       | /A      | ∞  | D | + |
| /A22*              | A21*          | A20*       | A19    | *      | RSTVEC | *   | X      | *      | C   | */B     | *       | A  | ∞ | D |
| A22*/A21*/A20*/A19 | *             | RSTVEC     | *      | X      | *      | C   | */B    | *      | /A  | ∞       | D       | +  |   |   |
| A22*/A21*/A20*     | A19           | *          | RSTVEC | *      | X      | *   | C      | */B    | *   | A       | ∞       | D  | + |   |
| /A22*              | A21*/A20*     | A19        | *      | RSTVEC | *      | X   | *      | /C     | */B | *       | A       | *  | D |   |

择优方案的操作将在图 1、2 以及表 2 中说明，并且已考虑到把 256 K 芯片都分配在主存储器模块 1 的  $B_0$  和  $B_1$  存储体上，以及扩展存储器模块 2 的  $B_0$  存储体上。

根据择优方案，3 号数据总线是一个 16 位的数据总线，而 DRAM 芯片都是一位组件。为了产生 16 位的数据信号，每一存储体上配置 16 个 DRAM 芯片。因此，按照上述情况布置，每一存储体安装一个 512 K 字节的存储器，从而主存储模块 1 的存储容量为 1 M 字节扩展存储器的模块 2 的存储容量为 512 K 字节，整个系统存储容量达 1.5 M 字节。

在响应系统复位式加电时，执行一个初始化程序。此时，主控制器 5 在地址总线 4 的  $A_3 - A_{23}$  地址线上（图 1），以及在控制总线 7 的  $\overline{RSTVEC}$  输出端和  $\overline{RESET}$  线上产生逻辑低信号，使得寄存器 10 的 1 Q—5 Q 输出端和 PAL9 的 ADDRESS, CTRL 和  $\overline{RSTVEC}$  输入端具有逻辑低信号，并且逻辑低信号  $\overline{RSTVEC}$  按照已知的方式被锁存在由地址总线 4 和控制总线 7 控制的锁存器 8 中。此时，一个逻辑高信号在  $\overline{RSTVEC}$  输出端产生并被施加给启动信号电路 6 A。因此，PAL9 的  $\overline{RCASO}$  输出变高，使主存储器模块 1 的存储体  $B_0$  上 512 K 字节存储区被启动。

由于  $\overline{RSTVEC}$  被锁存器 8 锁存在低逻辑电平，启动信号电路 6B 仍然处于闭锁（图 1），故扩展存储模块 2 的  $B_0 - B_5$  存储体没有一个被启动，从而避免了在 3 号数据总线上的竞争（图 1）。

为了确定 DRAM 芯片是否和存储体  $B_0$  相连接，在上面所说的已启动的 512 K 字节中，选择一个预先确定的存储单元（细节下述）并且通过 3 号数据总线把 5 号控制器中的一个十六进制数据写入此单元

中，当一个DRAM芯片被连接到已启动的存储体 $B_0$ 时，这个数据信号被5号主控制器读出。但是，倘若无一芯片与这个已启动的存储体 $B_0$ 连接，5号控制器从那里接收到一个缺席信号FF（十六进制）。

为了确定存储器的容量（即存储体 $B_0$ — $B_4$ 已连接芯片的个数），每个存储体按上述方法逐次的被启动和检查。

例如，为了启动存储体 $B_1$ ，5号控制器产生一个逻辑高信号，施于寄存器10的1D输入端，并把信号存在那里。一个逻辑高信号（A）从寄存器10被输送到PAL9中控制输入端（CTRL）中的一个对应端。因此，当X、B、C和D处于逻辑低电平而A处在逻辑高电平，PAL9的（ $\overline{RCAS1}$ ）输出变为逻辑低电平，以响应由5号主控制器在 $A_{11}$ — $A_1$ 地址线上产生的地址信号0001（见表2）。

从 $\overline{RCAS1}$ 输出的逻辑低信号被加到与非门12的第一输入端。与非门21的输出通常处在逻辑高电平，但是，当响应一个有源低REFRESH信号时就变为逻辑低电平。此REFRESH低信号是由地址线 $A_{13}$ 接收到的逻辑高信号并通过与它相连接的反相器22反相而产生的。所以，与非门12的输出从通常逻辑低电平变为逻辑高电平，以响应PAL9的 $\overline{RCAS1}$ 输出端变为逻辑低电平，或者是在与非门21的输出端变为逻辑低电平（即在刷新DRAM期间。其进一步细节在下面讨论）。

从与非门12输出的逻辑高信号被加到与非门26的第一个输入端。与非门26的第二个输入端接收一个逻辑高信号。此信号是响应于主控制器5产生的一个有源低地址选择信号 $\overline{A_5}$ ，由反相器45接收反相为高信号后加到与非门26。与非门26输出端在其输入端接收到上面所说的逻辑高电平时，变为逻辑低电平，从而产生了一个行

存贮选择信号  $\overline{RAS_1}$ , 用来启动在  $B_1$  存贮体上存贮器的所有行。

有源低  $\overline{RCAS_1}$  信号输出被与非门 17 反相并加到与非门 31 的第一输入端。反相后的  $\overline{As}$  信号通过延迟线 46 大约被延迟了 80 毫微秒, 然后加到与非门 31 的第二个输入端。因此, 与非门 31 的输出变为逻辑低电平, 从而在  $\overline{RAS_1}$  信号产生大约 80 毫微秒后, 产生一个列存贮选择信号  $\overline{CAS_1}$ , 用来启动存贮体  $B_1$  上存贮器的每一列, 进而全部启动  $B_1$  存贮体。

在用上面已说过的方式将 8 位列地址信号锁存前, 在行选择输出  $\overline{RAS_1}$  和列选择输出  $\overline{CAS_1}$  之间安排了大约 80 毫微秒的时间延迟, 这是为了锁存 8 位行地址信号提供足够的时间, 而地址信号的锁存是为了多路传输各 DRAM 电路的地址输入信号 (即是用一个预先确定好数量的电容负载与 DRAM 芯片相联系)。

为了检测 DRAM 存贮芯片是否连接在  $B_1$  存贮体上, 当存贮体  $B_1$  接到  $\overline{RAS_1}$  和  $\overline{CAS_1}$  信号被启动时, 主控制器 5 按上述讨论的, 往存贮体  $B_1$  中那个预先确定的存贮单元写入, 然后再读出上面所说的数据信号。

按照上述讨论, 根据现在的情况, 主存贮模块的  $B_0$  和  $B_1$  存贮体上已连接有 DRAM 芯片。从当前的情况, 由主控制器 5 来检测 DRAM 芯片的存在, 接着检查存贮体  $B_2$ 。

为了检查存贮体  $B_2$ , 逻辑低 C、D 和 X 信号被用作 PAL9 的 CTRL 输入, 并且逻辑高 A 和 B 信号也加到那上面, 使得  $\overline{RCAS_2}$  输出产生一个有源低逻辑信号, 作为接收到由  $A_{2,2} - A_{1,0}$  地址线产生的地址信号 0010 (十六进制) 的响应 (表 2 和图 2)。为了启动存贮体  $B_2$ , 与非门 27 和 32 的  $\overline{RAS_2}$  和  $\overline{CAS_2}$  输出分别变为逻辑低。主控制器

5 向存贮体  $B_0$  中的一个预先确定的单元写入并试图读出一个数据信号。但是，由于没有 DRAM 芯片连接在那里，读出的只是预先确定的缺席信号 FF（十六进制）。故，主控制器 5 仅仅对存贮块  $B_0$  和  $B_1$  检测 DRAM 芯片的存在，并且产生一个有着逻辑高 D。信号和逻辑低  $D_1-D_3$  信号的一级控制信号存入寄存器 10。从而，寄存器 10 产生一个控制信号加到 PAL9 的 CTRL 输入端，其中 A 控制信号处在逻辑高电平，B、C 和 D 信号处在逻辑低电平。因此，只有主存贮模块 1 的  $B_0$  和  $B_1$  存贮体被启动来响应接收再从主控制器 5 来的地址信号。

然后，主控制器 5 产生一个二级控制信号存放在启动信号电路 6B 中，为使在地址线  $A_{17}-A_{22}$  上接收的地址信号被重新变换以便使主存贮模块 1 上的最后一个可用的存贮单元和扩展存贮模块 2 上的第一个存贮单元相邻接。

从上面的讨论看出，启动信号产生电路 6A 和 6B 具有相同的设计（如图 2 所示）。但是，电路 6A 中，PAL9 的 RSTVEC 输入端是直接连接在主控制器 5 的 RSTVEC 输出端，而电路 6B 中 PAL9 的 RSTVEC 输入端是连接在锁存器 8 的输出端并且锁存器 8 的一个输入端和主控制器 5 的 RSTVEC 输出端相连接。为了能容纳 64K 或 256K DRAM 芯片，电路 6A 或 6B 的任一 PAL9 可以按表 1 或表 2 的公式来编程。

主控制器 5 把上述的二级控制信号装入启动信号电路 6B 的寄存器 10 中。根据二级控制信号，X 和 C 信号处在逻辑高电平，并被施加到 PAL9 的控制入端（CTRL）。A、B 和 D 信号处在逻辑低电平，并且也被加到那里。此时主控制器切换锁存器 8，使得对部件 6B 的 RSTVEC 变为逻辑高电平。启动电路 6B 的 PAL9 的 RCASO 输出端

以响应由  $A_{22}$ — $A_{10}$  地址线产生的地址信号 0010 (十六进制) 而变成一个有源逻辑低电平, 导致了 PAL 公式中的第三个“或”运算项为真 (表 2)。

关于表 2 可看到: 随着主存贮模块 1 的  $B_0$  存贮体被启动, 地址线  $A_{22}$ ,  $A_{21}$ ,  $A_{20}$  和  $A_{10}$  的值全为 0; 随着存贮模块 1 的  $B_1$  存贮体的启动,  $A_{22}$ ,  $A_{21}$ ,  $A_{20}$  和  $A_{10}$  的值分别为 0, 0, 0 和 1; 随着扩展存贮模块 2 的存贮体  $B_0$  的启动,  $A_{22}$ ,  $A_{21}$ ,  $A_{20}$  和  $A_{10}$  的值分别为 0, 0, 1 和 0。因此, 可看到扩展模块 2 的存贮器起始地址和主存贮模块 1 的最后一个地址单元是邻接的。

为了确定 DRAM 芯片是否连接在扩展存贮模块 2 的  $B_0$  存贮体上, 主控制器 5 在  $B_0$  中的一个预先确定的存贮单元中写、读数据。

下一步, 扩展存贮模块 2 的  $B_1$  存贮体被启动以响应启动电路 6 B 中 PAL 9 的  $\overline{RCAS1}$  输出端变为有源逻辑低信号。 $\overline{RCAS1}$  输出变为有源低电平以响应被加到 PAL 9 的  $A_{22}$ ,  $A_{21}$ ,  $A_{20}$  和  $A_{10}$  输入端上一个 0011 的值, 导致了表 2 中  $\overline{RCAS1}$  公式的第三个“或”运算项为真。 $B_1$  存贮体被写入信号用来确定 DRAM 芯片是否被连接在其上。而从  $B_1$  体中读出值为 FF, 表明没有 DRAM 芯片连接在  $B_1$  存贮体上。

如上面讨论, DRAM 芯片在响应主控制器 5 产生的  $\overline{REFRESH}$  信号时, 周期地被刷新。与非门 11—15 的每一个输出端变为逻辑高电平以响应  $\overline{REFRESH}$  信号的产生和地址总线 4 中  $A_{25}$  地址线上的逻辑高信号 (图 1)。与非门 25—29 的输出端变为逻辑低电平以响应与非门 11—15 的输出端变为逻辑高电平和响应对逻辑低  $\overline{AS}$  信号经过非门 45 而变为逻辑高信号的接收。因此, 与非门 25—29 的  $\overline{RAS0}$ — $\overline{RAS4}$  行地址输出端同时变为逻辑低电平, 以造成 2.5 M 字节存贮器的

刷新(或在 64 K 芯片中使用 640 K 字节)。

在一个成功的样机中,主存储器模块 1 可容纳 6 个存储器,这个数决定于是用 64 K 还是 256 K 字节 DRAM 芯片。为了能够再定位初始地址以便邻接主存储模块 1 的最后有效存储单元,表 3 列出了在主存储模块 1 上与各种存储体相一致的 A, B 和 C 信号的逻辑电平,它们可用于启动信号电路 6B 的 PAL 9 的控制输入(CTRL)。

表 3

| C | B | A | 存储体积(字节数) |
|---|---|---|-----------|
| 0 | 1 | 1 | 512 K     |
| 0 | 0 | 1 | 640 K     |
| 1 | 0 | 0 | 1 M       |
| 1 | 0 | 1 | 1.5 M     |
| 1 | 1 | 0 | 2 M       |
| 1 | 1 | 1 | 2.5 M     |

为了恰当地利用本发明,DRAM 芯片必须连接在主存储模块的邻近的存储体。例如,如果主存储模块 1 的  $B_0$ ,  $B_1$  和  $B_2$  存储体都连接 DRAM 芯片,而  $B_3$  存储体却没连接芯片,则本发明就不能恰当地起作用。特别是,主控制器 5 就只检测连接到  $B_0$  和  $B_1$  存储体上的芯片存在。

变动或更换设置可以由非常熟悉本发明的本技术领域的人来设计。例如,主存储模块 1 和扩展存储模块 2 可由多于或少于五个的存储体或更多数量的存储模块来组成,相应的启动信号电路也能够被使用,并且,被连接在这上面的存储器芯片可以分别大于或小于 64 K 或 256 K 字节,同时表 1 和表 2 给出了 PAL 方程作合适的改变。

另外，虽然这儿描述的只是参考采用 DRAM 芯片，其它有贮电路，比如磁泡存贮器或静态随机存贮存贮器等都可使用，但要对启动信号产生电路 6 A 和 6 B 作适当的修改。

此外，虽然 P A L 电路在这里被优先选来实现本发明的启动信号电路，其它可编程逻辑电路也可使用。例如，可编程逻辑阵列，等等。

对设计上的其它设置或变更可由理解本发明的人来设计。所有在发明范围内和所涉及的都在后面所附的权利要求中确定。

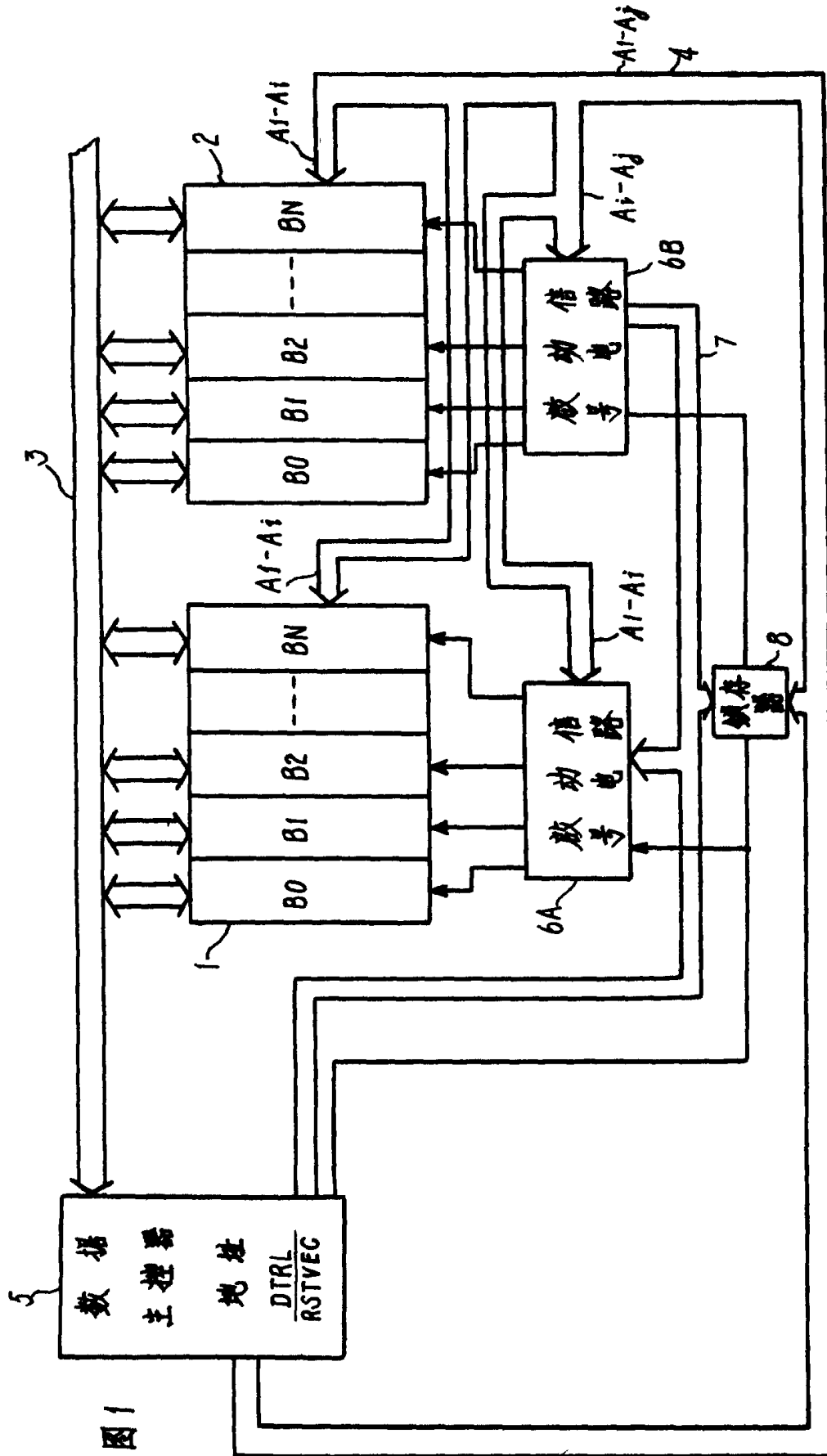


图1

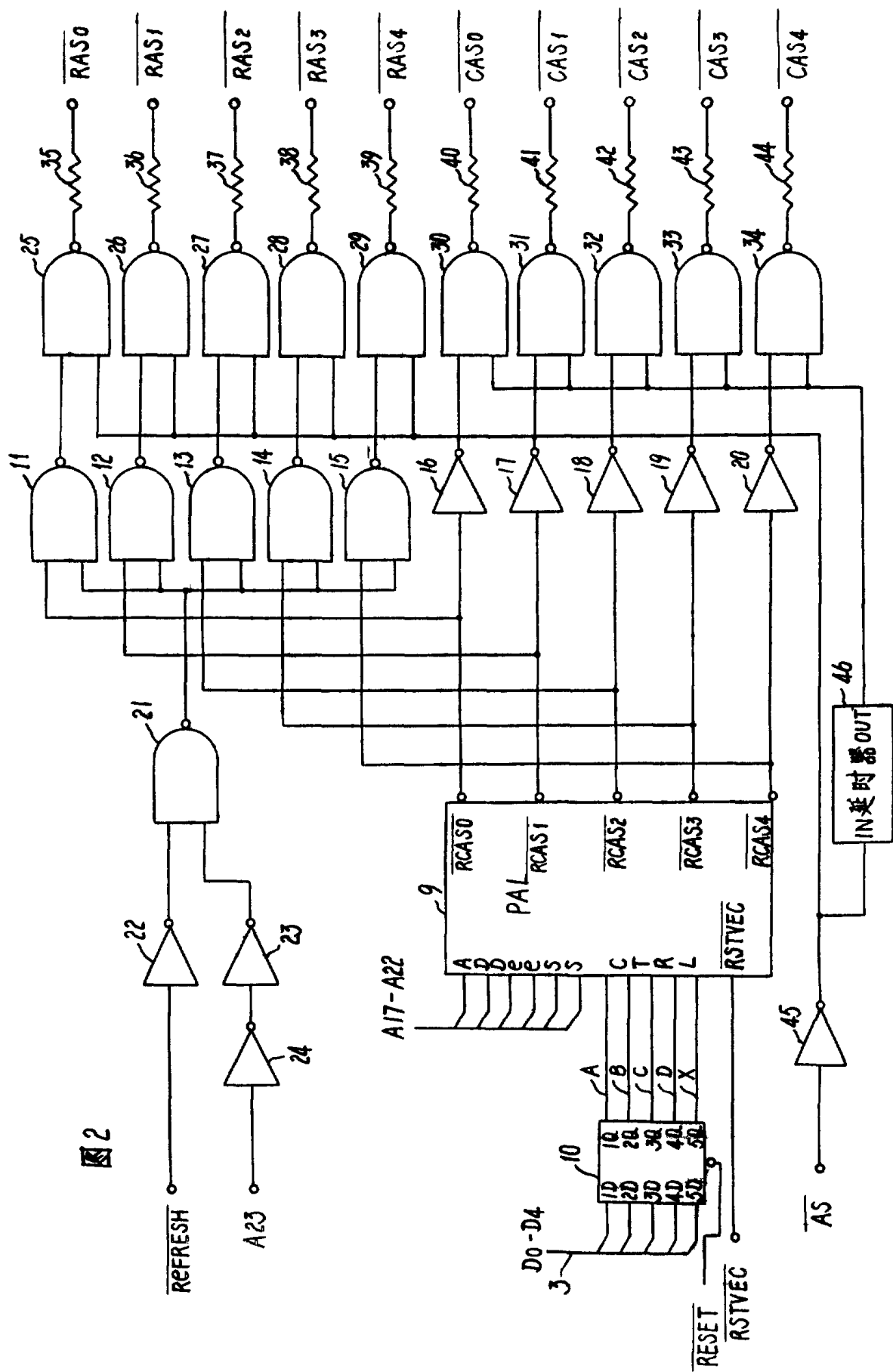


图 2