〔19〕中华人民共和国专利局

〔51〕Int.Cl.⁴
G06F 15/16

CN 86 1 06444 A

〔12〕发明专利申请公开说明书

〔11〕CN 86 1 06444 A

〔43〕公开日 1987年6月10日

〔21〕申请号 86 1 06444
〔22〕申请日 86.9.30
〔30〕优先权
〔32〕85.12.2〔33〕美国〔31〕803.364
〔71〕申请人 国际电话电报工业有限公司
地址 美国纽约10022
〔72〕发明人 史蒂文·格雷葛瑞·莫顿

〔74〕专利代理机构 上海专利事务所
代理人 刘庆玫

〔54〕发明名称 单元阵列处理机的地址产生

〔57〕摘要

此文揭示了一个阵列处理机，其包括排列在垂向行和横向列中的处理机矩阵。每行具有产生一个地址的能力，其可能不同于产生所有其它行的地址。因此处理机的每行包含一专用的地址处理机，其在各个行与数据处理机以及存储器相连接。这样，在各个行中地址处理机和存储器控制的地址由该行的数据处理机的状态决定。基于这种结构，由于各行具有单独产生不同地址的能力，使处理机能够处理的应用数目有了增加。

871A05107/39-136

权　利　要　求　书

1.在一个包含一个排列在M列×N行矩阵中的多重处理单元的处理机阵列中，所述单元在所述行和列与相邻单元相联接，特征在于在所述阵列的每一行中的改进包括：

一个地址处理机在所述行上联接于至少一个处理单元，一个存储器联接于所述地址处理器，在所述地址处理机附近的所述处理单元能与所述处理单元和所述存储器通讯。

2.根据权利要求1 的处理机阵列，其特征在于所述地址处理机与所述存储器通讯以向所述存储器发送一个地址，使所述存储器向所述处理单元输送数据。

3.根据权利要求1 的处理机阵列，其特征在于进一步包括与所述处理单元相联系的向所述地址处理机和所述存储器输送状态信号表明所述处理机动作的装置。

4.根据权利要求1 的处理机阵列，其特征在于所述处理单元能够运行给定数目的位，所述地址发生器能够处理相同数目的位。

5.根据权利要求4 的处理机阵列，其特征在于所述给定数目的位是16。

6.根据权利要求1 的处理机阵列，其特征在于能执行并行树算法。

7.在一个包括排列在M到×N行矩阵中的并行处理单元的处理机阵列中，所述单元在所述行和列上与相邻单元相联接，其特征在于每个所述行进一步包括：

一个具有串行数据输入线和一个输出数据线的地址处理机，

一个具有与所述地址处理机的所述输出数据线相联接的地址输入和具有与所述处理单元的输出数据线相联接的输出数据线的存储器。

一个第一缓冲器，其具有一个与所述地址处理机的所述输出数据

联接的输入和具有一个与所述处理单元的所述输出数据线相联接的输出。

一个第二缓冲器，其具有一个与所述存储器的所述输出数据线相联的输入和具有一个与所述存储器的所述地址输出线相联的输出。

写启动逻辑装置具有一个与所述处理单元的状态输出联接的输入和具有一个与所述处理单元的输出联接的输出，用来控制所述处理单元，所述地址处理机和所述存储器之间数据的输送,以允许在数据处理模式过程中的所述单元的运行。

8.根据权利要求7 的处理机阵列，其特征在于所述处理单元为具有串行输入数据端的数据处理机。

9.根据权利要求7 的处理机阵列，其特征在于进一步包括一个与所述处理单元联接的用于连接所述阵列中其它单元的第一数据总线，和一个与所述地址处理机联接的用来连接所述阵列中其它地址处理机的第二数据总线。

10. 根据权利要求7 的处理机阵列，其特征在于所述写启动逻辑装置包括一个具有与所述处理单元的状态输出数据线联接的输入数据线的第一寄存器，可编程序逻辑阵列的输出数据线，与移位寄存器的移位输入联接的所述逻辑阵列的输出，该移位寄存器具有与一个门相联接的输出，所述门以它与所述处理单元、所述地址处理机和所述存储器相联的输出产生一个写入启动信号。

11. 根据权利要求9 的处理机阵列，其特征在于进一步包括一个向所述阵列输送指令数据和有与所述第一和第二数据总线相联接的输出的控制器。

12. 根据权利要求11的处理机阵列，其特征在于所述控制器进一步包括一个与所述控制器联接的用于储存指令数据的程序存储器。

13. 根据权利要求12的处理机阵列，其特征在于所述控制器包括

取指令逻辑装置，该装置具有与所述程序存储器和所述第一和第二数据总线联接的输入，包括一个产生地址的程序计数器，从所述程序存储器中取出指令，所述取指令逻辑装置的输出与具有与所述存储器联接的输出的微程序序列发生器的数据输入相联接，所述存储器的输出与具有与所述微程序序列发生器的指令输入联接的输出的寄存器的输入相联接。

14．根据权利要求13的处理机阵列，其特征在于所述存储器是只读存储器（ ROM）。

15．根据权利要求7 的处理机阵列，其特征在于所述存储器为随机存取存储器（ RAM）。

16．根据权利要求15的处理机阵列，其特征在于所述寄存器输出是与所述处理单元，所述地址发生器和所述存储器相联接。

17．根据权利要求7 的处理机阵列，其特征在于所述处理单元和所述地址处理机均提供相同数目的位。

18．根据权利要求7 的处理机阵列，其特征在于所述处理单元的所述状态输出包括进位、负值、零和溢出引线。

单元阵列处理机的地址产生

单元阵列处理机已为人们所熟悉，它包括比较简单的处理机或单元的阵列，其中每个单元能在垂直和水平方向访问它的相邻单元。这种处理机被排列在M列和N行中，每个单元均与一个行和列相联系并在垂直和水平方向上与它的相邻单元联接。

单元阵列处理机处理并行数据流，同时获得大的吞吐量。当一常规单处理机在某一时刻以顺序方式处理一个数据项时，单元阵列处理机却能同时对许多数据目标进行处理。这样，对于任何单一指令，数据目标必须是同种类型的，因此，提供相同的顺序指令流用于同时处理这些数据目标才有意义。这种特殊类型的处理机称为单指令多数据处理机（ S IMD）。

一台单元阵列处理机可由单位或多位计算机的一个矩形阵列组成，其可由大规模集成电路（ L S I）来实现。例如每个单位均可有一个在处理机芯片的内部或外部的存储器，其可包括很大的存储量，例如从2k到64k 位。这些单元同时执行相同的指令，各自处理自己的数据。这些单元能在所有四个方向上与它们的相邻单元或外部数据输入输出寄存器相互通讯。因此该阵列能被应用到那些需要非常复杂的数学处理来解决的问题，例如矩阵列运算、向量计算、图像处理、模式识别、以及许多其它用途。

无论如何，已有许多广为人知的单指令多数据并行处理机的已有技术的例子，例如古德依尔（ Goodyear)的巨型并行处理机（ M P P），英国国际计算机有限公司的分布式阵列处理机（ I C L　D A P），以及全国现金出纳机公司（N C R）的G A P P芯片和日本电报电话

公司的关联阵列处理机（ＮＴＴ　ＡＡＰ）芯片。宝来伊利阿克Ⅵ（Burroughs　ＩＬＬＩＡＣ　ＩＶ）计算机不同于上面提到的机器，它对8-位,32-位，或64- 位字并行操作，其64个处理机单元的每一个具有一个变址寄存器和一个地址加法器，可变换称作存储服务部件（Memory　Service　Unit)的中央控制器所发出的地址。它不提供本文描述的同时的独立地址和数据计算，也不提供如本文描述的对存储器的独立的，与数据相关的写入启动。

重要的是，一台单指令多数据并行处理机也是由排列在一个长方矩阵中的处理单元阵列组成的，该阵列被一个与程序存储器联接的控制器控制，该控制器用于指令译码并使处理机按要求运行。处理机整个并行部分的存储器地址选择通过由控制器引出的一个阵列地址发生器来实现以向存储器提供一个地址。

注意这一点很重要：单元阵列处理机一般以单位串行方式运行，即，一位接着一位处理。就这一点而论，向各个处理机提供1-位存储器的分别寻址是不经济的。本文描述的处理机是以字并行为基础的处理，例如:16-位字，从而分别寻址不但经济上可行，而且简化了机器的程序设计模式和增大了机器所适合的应用范围。

因此，那些通过横向表征码和垂向表征码或内部表征码装置工作的单元均将是现役的单元。按照这种结构，存在一个单地址发生器，它的输出被所有与处理机阵列相联系的存储器利用。举一个特别的例子，让我们假定处理机阵列是由十六个16- 位处理机构成的，每个处理机后面有N字的存储器。地址发生器产生一个单地址可被所有十六个存储器用来取或存一个操作数。

此外，所有存储器不论读（ＲＥＡＤ）操作或写（ＷＲＩＴＥ）操作是否被执行均会运行。并行处理机的结构特别适合处理机阵列的字长与所需的地址的字长没有特别关系的场合。可以再设想处理机阵

列由排列在16行×16列构形中的256 个单位处理机组成，此情况中，可设想从存储器，更确切些，从一字集中取出一个位平面。

在此情况中，当一些处理机为非现役时，避免写入一个不该被写入存储单元的唯一办法是执行一个读出- 破坏- 写入（ R E A D- M O D I F Y- W R I T E）操作。因此，所有存储单元被读并当那些非现役的处理机的数据是被写的时候，与被读的完全相同的数据被送回到那些存储单元，同时对于现役的处理机，新的数据被送回来替代旧的。

执行或完成一读出一破坏一写入操作要用二个循环，每当想要实现一单写循时总是因此使处理机减慢。这是单元阵列处理机的一个典型问题。

另一个困难是当地址发生器或从地址发生器一直到存储器的地址总线失效时导致整个处理机的失效。因此，一个单点失效会造成整个机器的失效。此外，如果处理机特别大，来自遍及整个机器的单地址发生器的地址分布会耗费大量时间，不得不靠增加电路设计的复杂性来减小对速度的影响或简单地让机器减慢速度。

第三个、进一步限制了这种并行处理体系结构的难题是在一定要写的程序中仅一个单地址足以满足所有处理机取数据。一种不适合于这种体系结构的程序类型是树检索算法，在这种树检索算法中，每个处理机通过它的存储器中的树进行检索。每个树的分支有一个指向下一个被检索分支的指针，在该情况中，在整个机器中有不同的正在使用的指针，但在单地址发生器的情况下，很清楚这是不会发生的，除非是在浪费并行处理机能力的前提下检索树。

还有一个困难是，如果想要实现地址产生，该地址是由来自处理机阵列某处的数据驱动出来的，则需要这样一个选择机械来实现，其中有一个特别的处理机提供它的数据，该数据接着被输入到地址发生

器中。这种选择装置的实现是困难的，并且传输数据的时间会进一步减慢机器的速度，因而象这样的选择装置是很少使用的。

另外，若处理机的字长与地址发生器的字长不一样，字长间的转换就有困难。而且，由于一个处理机的某一行可以有一字集在内，即，可以假想一个16- 位行，内含有二个8 - 位处理机，就存在一个转换问题，其中处理机阵列左边的位将需要与最少有效部分通讯，也就是说，与在地址发生器右半边的位通讯，这个转换又进一步复杂了结构。完全可说：在并行处理机结构中，主要的限制是用于寻址特定单元的方法。这一方法导致上述各项困难并且因此严重地损害了机器用于执行数学与其它逻辑运算的速度。

因此，本发明的一个目的是提供一分布式寻址的处理机阵列，其中在一特定行内的相应的每个处理机单元与它自身的地址发生器相联接。这样，并行树检索算法可实现，并因此可用更可靠的方式为许多在常规的单处理机中常遇到的算法提供快速的运算。

在一个包含有排列在M到N行矩阵中的多元处理机构成的处理机阵列中，所述处理机在所述行和列中与相邻处理机相联接，在所述阵列中的每一行中的改进措施包括：至少有一个地址处理机与所述处理机相联接，一个存储器与所述地址处理机相联接，并且在所述地址处理机附近的处理机能与所述地址处理机和存储器通讯。

图1 为根据本发明原理的具有分布式寻址的单指令多数据处理机的框图；

图2 给出了能由图1 处理机执行的二叉树检索的简图；

图3 说明了能由图1 处理机执行的并行N元组树检索的简图；

图4 为一个替换实例，示出了具有分布式寻址、顺序地址和数据处理的阵列处理机；

图5 为把变量变换到存储器线性阵列图；

图6 说明了按行序把变量变换到存贮器中方形数据阵列的表；

图7 说明了按列序把变量变换到存储器中方形数据阵列的表；

图8 给出了把变量变换到存储器立方矩阵中去的表；

图9 为能用于本发明的16- 位中央处理机的框图；

图10为图9 的中央处理机的简化框图；

图11为根据本发明的写入启动逻辑框图；

图12为应用于本发明中的典型的处理机行构造框图；

图13为能与本发明一起使用的系统控制器的简化框图。

参照图1,其为根据本发明的具有分布式寻址的阵列处理机。

阵列处理机包括一个多元数据处理机，各个处理机排列在一行中，数据处理机是分别独立的处理部件，并且可变化出很多构造。

参照图1,其示出了处理机的每行不仅包含一个数据处理机20，而且包含一个存储器如存储器21，和一个地址处理机22。在每一行中，地址处理机22向存储器发出一个地址并且从存储器来的数据返回数据处理机20或可通过双向缓冲器23被送回到地址处理机中。正如能从图1 可见的，处理机阵列的各行包括相同的部件一即，数据处理机、地址处理机和存储器。

借助与程序存储器25联接的控制器24控制整个阵列，控制器和程序存储器均为常规的设备并被用在已有技术的并行型处理机中。另外，来自数据处理机20的输出信号被送向地址处理机22和存储器21。数据处理机的类型可以是具有灵活地自行打开和关闭能力的装置，而不同于具有外部横向表征码和垂向表征码的或一简单2 - 位内部表征码的装置，供处理机选择。

举一个类型为根据输入数据能自行打开和关闭或启动和撤消的处理机为例，参见由本发明人 S．G．Morton1985 年11月13日提交的题为“具有内部单元控制和处理的阵列结构”（ ARRAY

REORGANIZATION WITH INTERNAL CELLULAR CONTROL & PROCESSING）这一相关的未定申请。为了解释它的操作，让我们首先假定包含在处理机一行中的字数为1,例如设想一个阵列包含16行，每行有一个单16位数据处理机如处理机20。

进一步假定具有与处理机上述行相联接的64k 字的存储器已是足够，在这种情况中，地址处理机也是16位宽。由于经常需要产生一个基于数据的地址，假定数据处理机20和地址处理机22一样宽，有利于它们之间的信息通过。

在图1 结构中，每行有它自己的地址处理机如处理机20，并因此可完成并行树检索算法。图2 给出了这种并行树检索算法的一个例子，如图2 所见，每行跟随一个二叉树，树的每个分支有指向下一个分支的指针，所有树可被同时检索或者不同的分支可在处理机的各个行被检索。

因此，参照图2,图2 给出了象征树0 、树1 、树2 和树3 的并行二叉树检索。在图中清楚地说明了检索特性，如可理解的，每行沿着所指出的二叉树其每个树分支指向下一个分支。不同的分支可在处理机的各个行中检索。

参照图3,其给出了N- 元组分支的扩充，在那里一些分支被截断。注意，因为地址处理机位于一单行中，如果那个地址处理机失效，则仅那行受影响，在此情况下，可建立一个有备行的处理机以补偿一个或多个失效行。

此外，由于地址处理机是在行上，从地址处理机到存端器的相互连接长度被减至最小，这进一步改善了性能。依据那些现役的数据处理机，不同行将以不同的处理组合被现役。在任何情况中，地址处理机如处理机22跟踪数据处理机20是必要的，鉴于在常规应用中，这是

指一个运行在单指令单数据机中的程序，如果数据条件被检测，其中产生程序转移，然后很清楚，在被转移回来的这段编码中不会实现地址计算。

模拟情况是，如果数据处理机20非现役，那就是说不执行某一运算，则同样关联的地址处理机也为非现役。此外，通过利用活动位控制存储器，存储器写（WRITES）可被关闭而不去用“读出—破坏—写入”操作来操纵处理机是非现役的情况。

更困难的情况是数据处理机20的宽度与地址处理机的宽度不同。例如，由于一行仅需要64k字的存储器,有可能需要进行32-位数据运算而不仅是16- 位的地址运算。此情况中当数据处理机20基于数据产生一个地址时，仅仅字的低位需要被传向地址处理机22。

更复杂的情况，是要求数据处理机的宽度比地址处理机的宽度小，在此极端情况中，假设想要一个单位的地址处理机，同时地址处理机为16位宽。由于一行确实包含繁多的这种1 - 位数据处理机，问题是如何处理繁多的具有单地址处理机的数据处理机。这种情况事实上模拟了常规的单地址发生器处理处理机阵列的多重的行和列的结构。

设想每个变量有其自己的地址。那么就有大批可借助机器实现的应用，如果每个地址不得不有多于1 个的变量,则应用的数目被明显地减少。为了解决这个问题并减少硬件数量，人们把大量的寄存器应用到地址处理机中，因此如果在地址处理机中有一个寄存器组则各个寄存器可被分派给数据处理机的各个变量。

例如，如果地址处理机中有16个寄存器并且如果数据处理机是作为一个单16- 位机器,16 个地址寄存器将是可利用的。按平移的比例，如果数据处理机被用作为二个8 - 位处理机，则可以想象地址寄存器的一半将被每个处理机利用。在极端情况下如果在数据处理机中有16个单位变量，可以想象地址处理机中的单寄存器被分配到数据处理机

的那些位中的每一个位。地址处理机的位的数目将不会改变，会发生的是操作将会减慢，因为将必须执行多存储循环即在数据处理机中每个变量对应一个存储循环。

但是，希望在数据处理机的每个变量的地址计算中有完全的通用性，则这个解决办法体现在使硬件极少。另外一个完全地址处理机必须被提供给数据处理机中的每个变量，这会意味着机器很大和具有较差的性能价格比。

所以，目标之一是提供一种机器，其中数据处理机中的位数目是比较大的，典型的有16- 位或32- 位或更多，并且整个地址处理机可被分配给机器。

参照图4,其给出了一个替换结构，图4 中，在减少费用的努力中，处理机的每一行以时间分割多路转换形式利用它的输出总线。因此，在执行一个存储循环时，处理机输出数据的每一行输入一个如寄存器30的与处理机31相联接的寄存器。因此，当执行一个存储循环时，处理机的每一行输出一个输入到类似寄存器31的数据，寄存器31将地址储存到存储器32中。在下续循环中数据被输送到存储器32或从中取出。

时间分割多路转换器用于地址和数据在同一总线上的输送，这种使用在许多微型处理机中是常规手段。这里主要差别是这些处理机不是独立地被控制的微型处理机，但这些处理机是执行一个单指令多数据方式，并且全部随动于一个如与程序存储器36相连的控制器35的单控制器。

图4 所示的系统没有图1 所示的系统复杂，但是由于在处理机的每一行与它的存储器之间只有一个单总线效果也没有图1 的好。这个总线必须首先运载地址然后是数据，而在图1 中存在分开的地址和数据总线，在这种情况下，图1 所示的效能是图4 所示电路的二倍。

附加指出的是图4 中可用的寄存器数目大概是图1 可用的寄存器

数目的一半。如果每一处理机的寄存器数目，不管它是地址处理机或是数据处理机，保持常数，这对编译系统有影响，如果寄存器总数有一定的最小值，则一定的编码优化可做得较好、在图4 的构型中，仅有一半的寄存器可利用。

图5 、6 、7 和8 显示了变换变量到存储器的方式。图5 展示了一个线性机阵列。为简单起见假定存在一个4 - 行机因而每行需要二个地址以储存所有八个变量。在该情况下，简单平面寻址，即整个机器的单地址，已能足够地存取这些变量。

在图7 的方阵中仍示出了一台被假定的4 - 行机和一个8 ×8 矩阵。如果该矩阵能如图6 、7 所示的以行序或列序储存的话，在任一种情况下，简单平面寻地已足够能存取在图6 情况中的行单元或图7 情况中的列单元。可是，如果想要存取在矩阵逆运算中常用的对角元时，由于每个处理机必须有一个不同地址，在单运算中就不可能实现。

例如在图6 中，需要行0 中的地址0 、行1 中的地址2 、行2 中的地址4 和行3 中的地址6 。在这种情况下，有能力在每一行中产生不同地址就简化了对角元的存取。同样，在图8 中，一个立体矩阵被变换到存储器中并在这种情况下假定了一个2 - 行机和一个4 ×4 矩阵，能再次确认通过在每一行中有不同的地址大大简化了存取对角元。特别地，行0 会提供地址0 以得到元A、O、O、O和1 会提供地址10来得到元A1 、1 、1 。

图2 给出了根据本发明可实现的并行树检索。假定在树的每一个层上，对所有树执行同样测试。依靠这些测试的结果，树的一个分支将被送作为下一个测试。作为每个测试的结果，二个指针中的一个将被跟着用来指向处理机的将要在树的下一层中被测试的数据。四个不同的通过树的途径已被给出，用来展示每个树中的分别寻址允许采用独立的途径。

图3 给出了并行N- 元组树检索。该例中在树的每一层中有多重选择。在每层中一个测试集被执行，用来确定许多分支中的哪一个是下一个执行的分支。其中一些分支会发生在那些处理机当其它处理机继续运行时它们会变成非现役的那些点上截断，在任何情况下每个处理机需要能力跟随一个特定的通向执行的树的寻址通道，在这种情况下对整个处理机的单地址阻碍了这种类型结构的计算。

参照图9,其给出了一个典型的16- 位中央处理机(CPU) 的典型框图。该框图与《现代微型设备》(ADVANCED MICRO DEVICES),1983年AM2900家庭数据手册(FAMILY DATA BOOK),第557 页中的结构相类似。基本上图9 是取自该页中图2 。这个16 -位的中央处理机被用在数据处理机和如图1 所示例子的地址处理机中，并且其被包含在如数据处理机20和地址处理机22的每行中。

该16 -位中央处理机能被用来实现如图1 中说明的地址处理机22和数据机20的功能。

参照图10，其示出了一个简化了的16- 位中央处理机的框图作为图9 所示构型的一个例子。值得指出的是图2 所示的垂直总线与2903地址总线(DB) 总线相联接，并且图1 的M总线被联接到2903的Y总线。控制信号，WE(写入启动) 、指令0 到指令8(I0 - I8) 、输入启动B(OEB),和输出启动Y(OEY) 都来自如图1 中作为时钟的控制24的控制器。

附加在两个总线上的芯片的主要输出为四个状态线一即，进位、负值、零和溢出由C、N、Z、O表示。

图11给出了写入启动逻辑，其是引起适当的运算所必需要的。本程序块的初始输入为4 状态输入，条件选择，和进栈与退栈指令，并且有单一输出，由WE代表的写入启动，从图11可见：进位，负值、

零、溢出这些状态输入被用作寄存器40的输入。寄存器40的输出被指向一个4-输入线用作表明条件选择的可编程序逻辑阵列41。来自可编程序逻辑阵列41的输出被指向一输出寄存器，此寄存器收到一个退栈（POP）信号就向左移位，收到一个进栈（PUSH）信号就往右移位。它还包括用来表明往左或往右移动的第一与第二个数据输入。

可编程序逻辑阵列41选择十六个测试条件中的一组，作为例子，不论测试表明值小于零或大于零或等于都是依赖于对C、N、Z、O、这些输入在条件选择下的选择。这样，条件选择引起可编程序逻辑阵列41在对C、N、Z、O组成的输入值的一个选择产生后就输出一个真条件。

处理机单元能是一个现役阵列，并且处理机单元的结构也是众所周知的。作为例子参见由本发明人S．G．Morton 于1985年11月13日提出的“具有内部单元控制和处理的阵列结构”这一相关的未定申请。

对于相互未定申请，图4 给出了作为在单元阵列或单元处理机中使用的一个处理机单元的详细框图。在此应用中，处理机单元的计算部分由一个多端口随机存取储存器（RAM）组成，其中二个单元可以由读地址和读/ 写地址选择同时被读，并且其中一个单元在写单元是读/ 写地址时可以被写。

多端口随机存取存储器的输出进入到一个数学逻辑单元（ALU的A与B输入。此数学逻辑单元与多端口随机存取存储器是如AMD 2903中的常规的设计。对处理机单元的控制发出指令形式的输入，读地址与读/ 写地址，所有这些通过如图4 的控制器35来实现的。

数学逻辑单元的输出与缓冲器相联接，缓冲器的输出与一多路转换器相联接。数学逻辑单元的另一输出与多路转换器的另一个输入相联接，这样，该多路转换器可以从数学逻辑单元的输出线上选择一个

输入并把内容送到状态寄存器中。

状态寄存器与一缓冲器相联接，该缓冲器的输出与一多端口随机存取储器的输入相联。从这一应用可得出：状态寄存器与一可编程序的并且可能含有一微处理机的状态逻辑阵列相联系。

在这一结构中，上述C，进位；N，负值；Z，零；和O，溢出；是利用本发明的和从数学逻辑单元获得的输出。

上述应用给出了在每一处理机单元中实现控制RAM写启动信号，使得在一个阵列中的每一单元在确定了该单元是否被选用后，执行一条指令的结构。这样，在一个阵列中的每个单元在同一时刻执行某一条指令。在一个阵列中的其余单元对这同一指令将闲置。这样，在本应用中描述的处理机特别适合于本发明的应用领域。

输出门44通过监测储存在寄存器42中的位来确定一处理机单元是否被使役以确定写入启动线的现役条件。

图12给出了一个阵列处理机的行框图。从图12中可以得出：有具有数据输出与随机存取存储器52的数据输入相联的16- 位数据处理机50。上述数据也可以通过门55被用到随机存取存储器52上。地址处理机51也是一个16- 位处理机，并且基本上从包含与前述图11相连的逻辑结构的写入启动逻辑单元53接受它的输出值。

须指出的是：本发明的主要因素是在一处理机中，一单行有一地址处理机51和一数据处理机50，并且它们各自执行分离的指令流。这些由11- 位输给出，对数据处理机50用 I_D 表示对地址处理机51表示用 I_A 表示。数据处理机的状态输出被用作为写和启动逻辑的输入。接着这个写和启动逻辑能使数据处理机50，地址处理机51和随机存取存储器或RAM52运行。可以想象图12给出的多重电路能包括一孤立行，但是这些区域中的每一个象图中给出的那样是相互连接的。

须指出的是：地址处理机51有状态输出，但是仅利用了从数据处

理机50得出的状态输出，这就简化了机器的程序设计模式，使得它与通常的程序一致，其中进行对一些数据的运算，作为上述运算的结果，一个程序流可以被修改或不被修改，虽然在原则上可以用地址处理机51的输出作为对写入启动逻辑53的进一步输入，但是这样会使程序设计模式复杂化。

正如在图12中给出与表明的：有一对缓冲器54和55，它们能使在地址处理机51中的M个输出与数据处理机50的M个输出间的 传输得以实现。这些如54、55的缓冲器有助于将随机存取储存器52的数据输出作为地址处理机51输入的使用，并且也有助于在地址处理机51与处理机50间的数据运动。

在许多应用中，执行一关于数据的运算并且利用该结果来计算一地址。因此，数据处理机50的一个输出不得不被送入到将执行一个计算来产生一个地址的地址处理机51。

参照图13，给出了一个如图1 控制器24的控制器简单框图。存在一个取指令逻辑电路60，其中由一用来产生地址的一个程序计数器组成，通过它可以从程序存储器中取出一条指令。来自程序存储器的指令将被输入到指令寄存器中。指令寄存器的输出为了对微程序只读存储器（ ROM）产生一个开始地址，一般是通过一个只读存储器变换。这个地址穿过D输入到2910微程序序列发生器。2910的 Y输出寻址微程序只读存储器62，这个存储器的内容将驱动一寄存器63并且作为一条执行下一个微指令的指令，一些位被反馈到2910。须指出的是:291(是可以从AMD和其它来源中得到的常规的微序列发生器电路。

寄存器63的输出对阵列处理机提供了不同的控制线。用 VD与 VA表示的阵列处理机的双总线是与程序存储器相连的，使得来自程序存储器的数据可以通过上总线或使得数据可从阵列处理机被读出并送回到程序存储器中。这些地址总线在关于阵列处理机行的框图12中

给出。

当然，可以理解到数据处理机和地址处理机包括控制器能利用通常的元件来装配组成。上述本发明的基本点是提供一包括有一系列行的单指令多数据处理机。这些行中的每一行有能力来产生一个地址，这一点隐含着与其它所有行中产生的地址是不同的。这些地址处理机的地址控制和各行中的存储器是由那一行中的数据处理机的状态所决定。这样的可以被处理的应用范围由于在每一行有不同的地址产生而被大大地扩展。由于不存在由于它的失效能造成整个机器失效的孤立的发生器，错误的影响也被减小了。

进一步，因为地址的传输距离由整体减为局部，使得效率也有提高。因为地址处理机与数据处理机可以同时工作，效率也被进一步提高。在给定行中的现役的地址处理机受控于在的那一行中现役的数据处理机，就象存储器与每一行相联。

具有分布寻址的单指令多数据阵列处理机

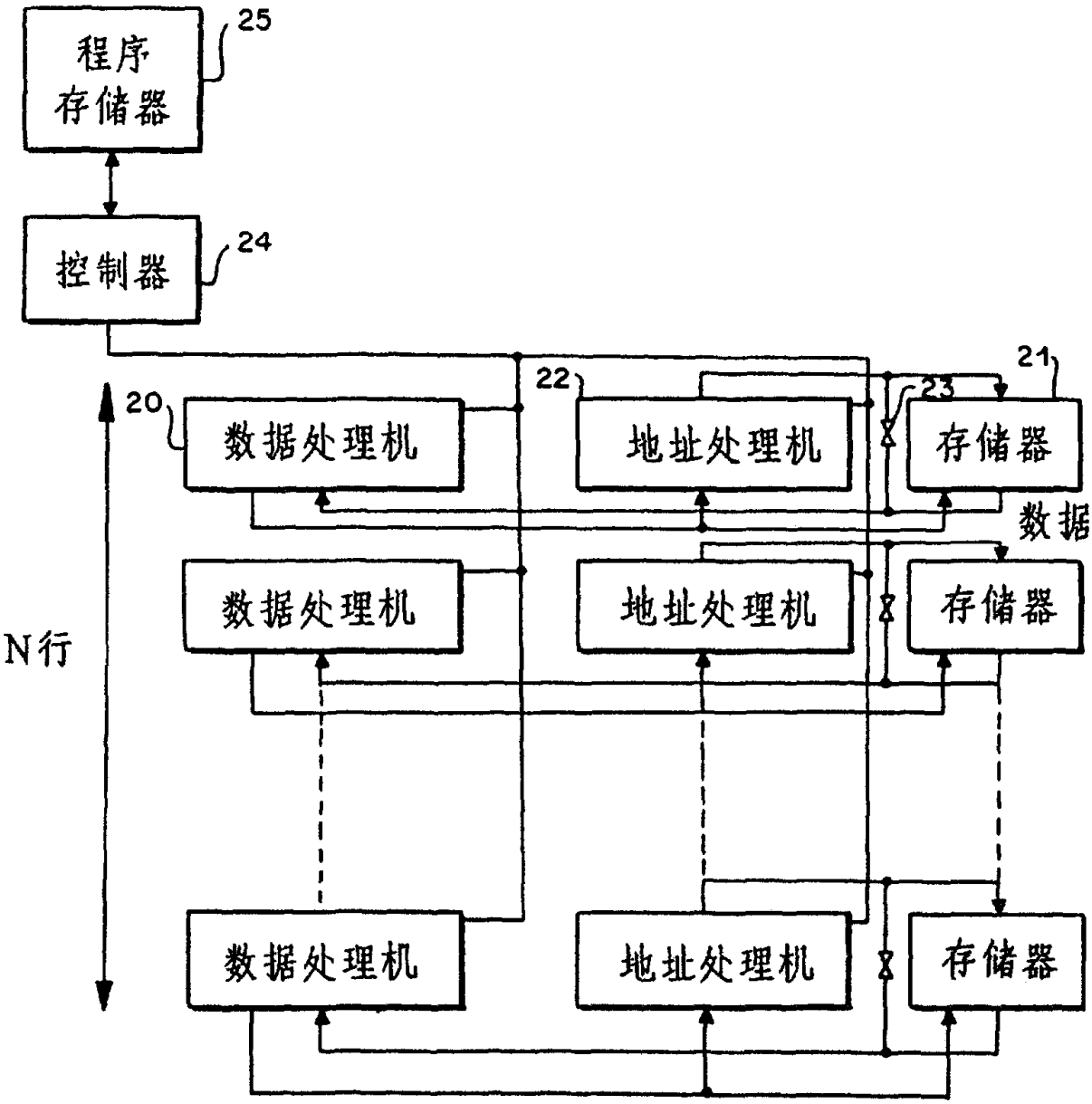


图 1

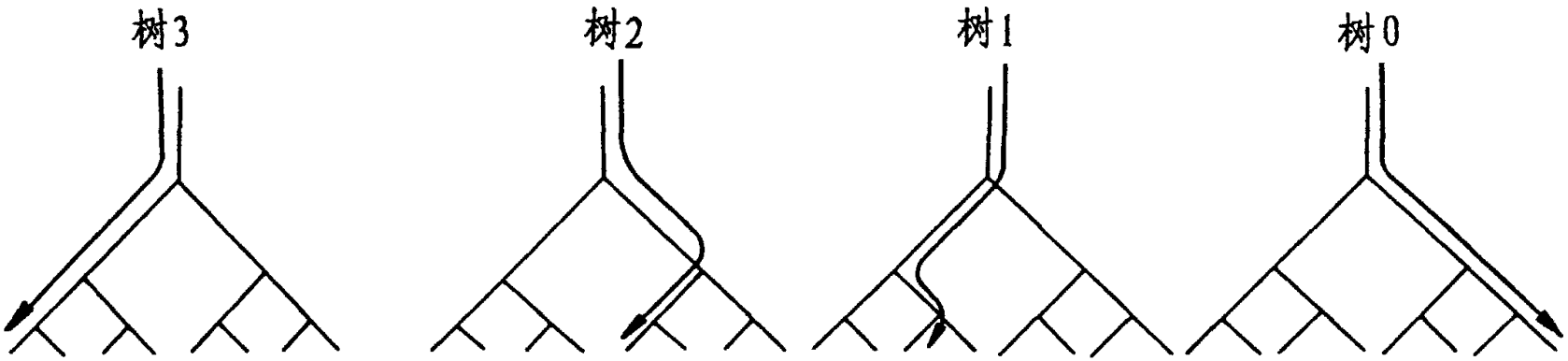


图2

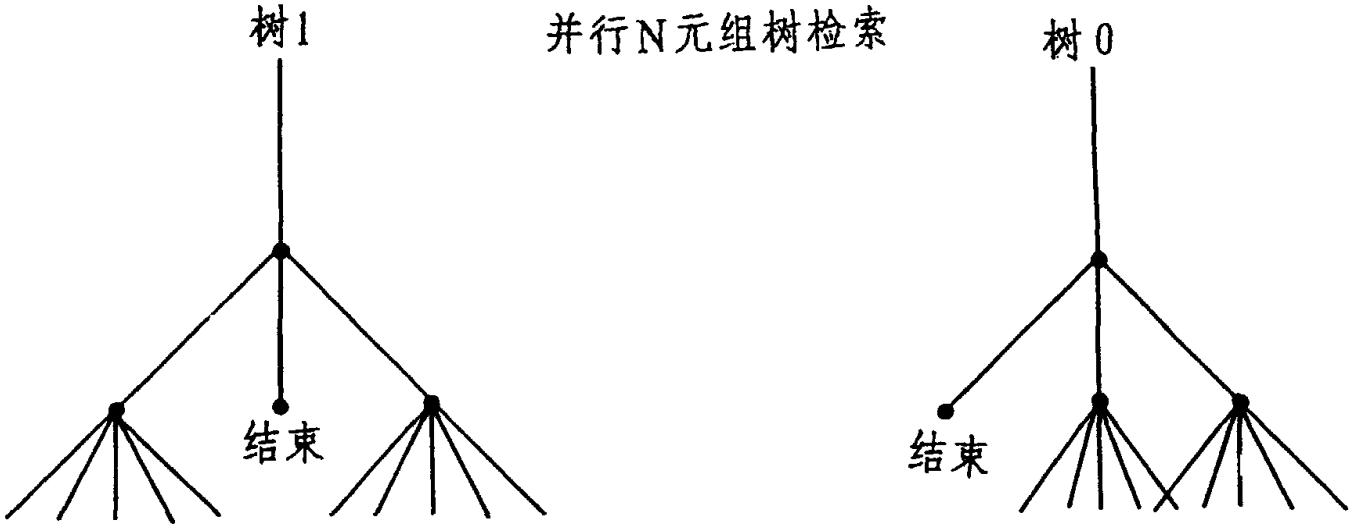


图3

具有分布寻址的单指令多数据阵列处理机

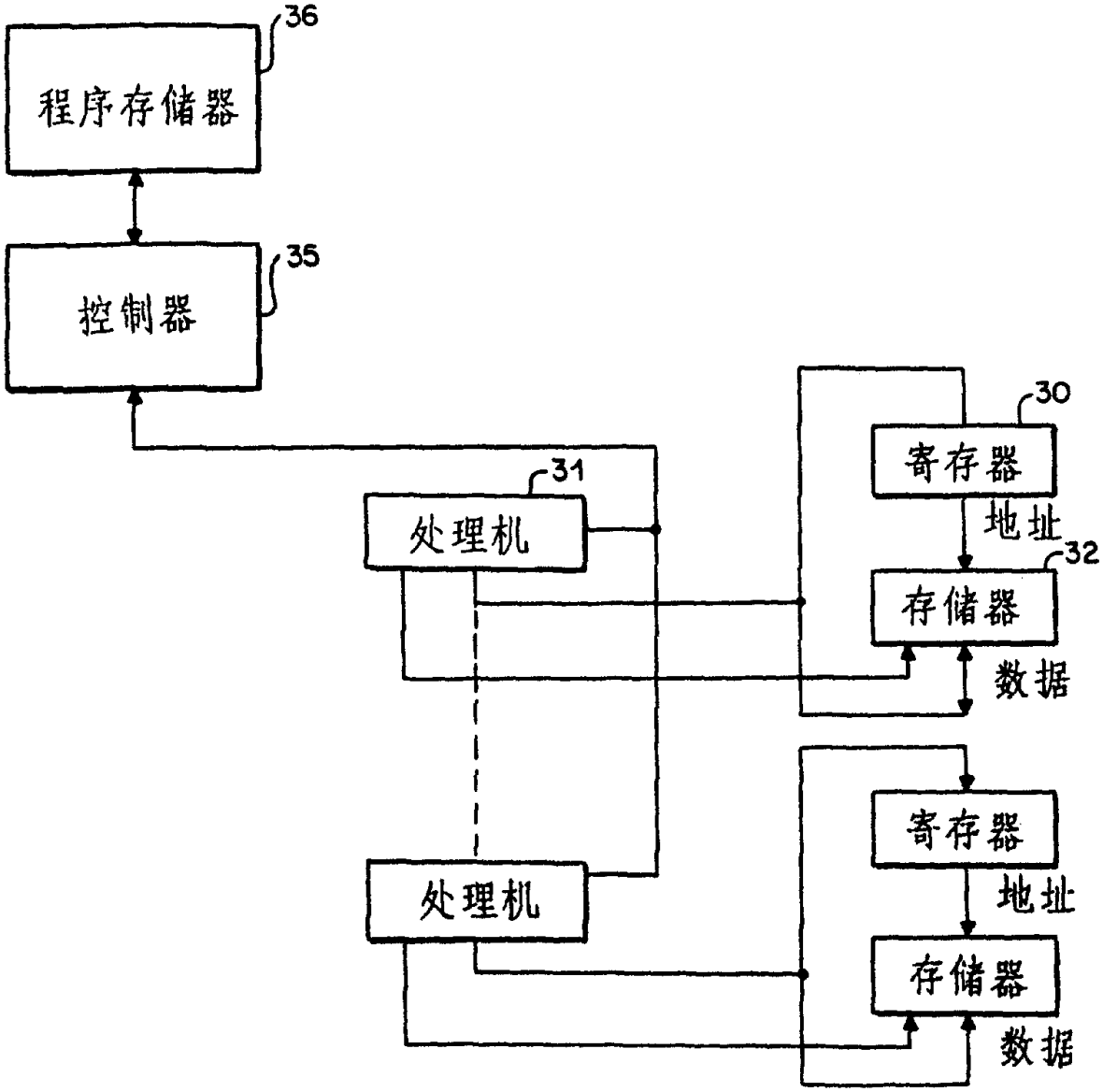


图 4

将变量变换到线性存储器阵列

为简单起见，假定一个4 行机，进一步假定地址对一个处理机行是局部的，而不是对整个控制器总体的。

地址	行3	行2	行1	行0
0	A3	A2	A1	A0
1	A7	A6	A5	A4

标志 A N中，N是行号

图 5

将变量变换到矩形存储器，行序

地址	行3	行2	行1	行0
0	A0,3	A0,2	A0,1	A0,0
1	A0,7	A0,6	A0,5	A0,4
2	A1,3	A1,2	A1,1*	A1,0
3	A1,7	A1,6	A1,5	A1,4
4	A2,3	A2,2*	A2,1	A2,0
5	A2,7	A2,6	A2,5	A2,4
6	A3,3*	A3,2	A3,1	A3,0
7	A3,7	A3,6	A3,5	A3,4
8	A4,3	A4,2	A4,1	A4,0
9	A4,7	A4,6	A4,5	A4,4*
10	A5,3	A5,2	A5,1	A5,0
11	A5,7	A5,6	A5,5*	A5,4
12	A6,3	A6,2	A6,1	A6,0
13	A6,7	A6,6*	A6,5	A6,4
14	A7,3	A7,2	A7,1	A7,0
15	A7,7*	A7,6	A7,5	A7,4

* 代表对角元

标号 A N，M中， N是行号，M是列号。

图6

将变量变换到矩形存储器，列序

地址	行3	行2	行1	行0
0	A3,0	A2,0	A1,0	A0,0*
1	A7,0	A6,0	A5,0	A4,0
2	A3,1	A2,1	A1,1*	A0,1
3	A7,1	A6,1	A5,1	A4,1
4	A3,2	A2,2*	A1,2	A0,2
5	A7,2	A6,2	A5,2	A4,2
6	A3,3*	A2,3	A1,3	A0,3
7	A7,3	A6,3	A5,3	A4,3
8	A3,4	A2,4	A1,4	A0,4
9	A7,4	A6,4	A5,4	A4,4*
10	A3,5	A2,5	A1,5	A0,5
11	A7,5	A6,5	A5,5*	A4,5
12	A3,6	A2,6	A1,6	A0,6
13	A7,6	A6,6*	A5,6	A4,6
14	A3,7	A2,7	A1,7	A0,7
15	A7,7*	A6,7	A5,7	A4,7

* 代表对角元

标号AN，M中，N是行号，M是列号。

图 7

将变量换到立方存储器矩阵

为了简化假定一个 2- 行机
立方阵- A0，0，0- A3,3,3
对于每个平面n：

A0,3,n	A0,2,n	A0,1,n	A0,0,n
A1,3,n	A1,2,n	A1,1,n	A1,0,n
A2,3,n	A2,2,n	A2,1,n	A2,0,n
A3,3,n	A3,2,n	A3,1,n	A3,0,n

列序：

地址	行1	行0	地址	行1	行0
0	A1,0,0	A0,0,0	16	A1,0,2	A0,0,2
1	A3,0,0	A2,0,0	17	A3,0,2	A2,0,2
2	A1,1,0	A0,1,0	18	A1,1,2	A0,1,2
3	A3,1,0	A2,1,0	19	A3,1,2	A2,1,2
4	A1,2,0	A0,2,0	20	A1,2,2	A0,2,2
5	A3,2,0	A2,2,0	21	A3,2,2	A2,2,2*
6	A1,3,0	A0,3,0	22	A1,3,2	A0,3,2
7	A3,3,0	A2,3,0	23	A3,3,2	A2,3,2
8	A1,0,1	A0,0,1	24	A1,0,3	A0,0,3
9	A3,0,1	A2,0,1	25	A3,0,3	A2,0,3
10	A1,1,1*	A0,1,1	26	A1,1,3	A0,1,3
11	A3,1,1	A2,1,1	27	A3,1,3	A2,1,3
12	A1,2,1	A0,2,1	28	A1,2,3	A0,2,3
13	A3,2,1	A2,2,1	29	A3,2,3	A2,2,3
14	A1,3,1	A0,3,1	30	A1,3,3	A0,3,3
15	A3,3,1	A2,3,1	31	A3,3,3*	A2,3,3

* 代表对角元

标号 A N，M，L中，N是行号，M是列号，L是平面号。

图 8

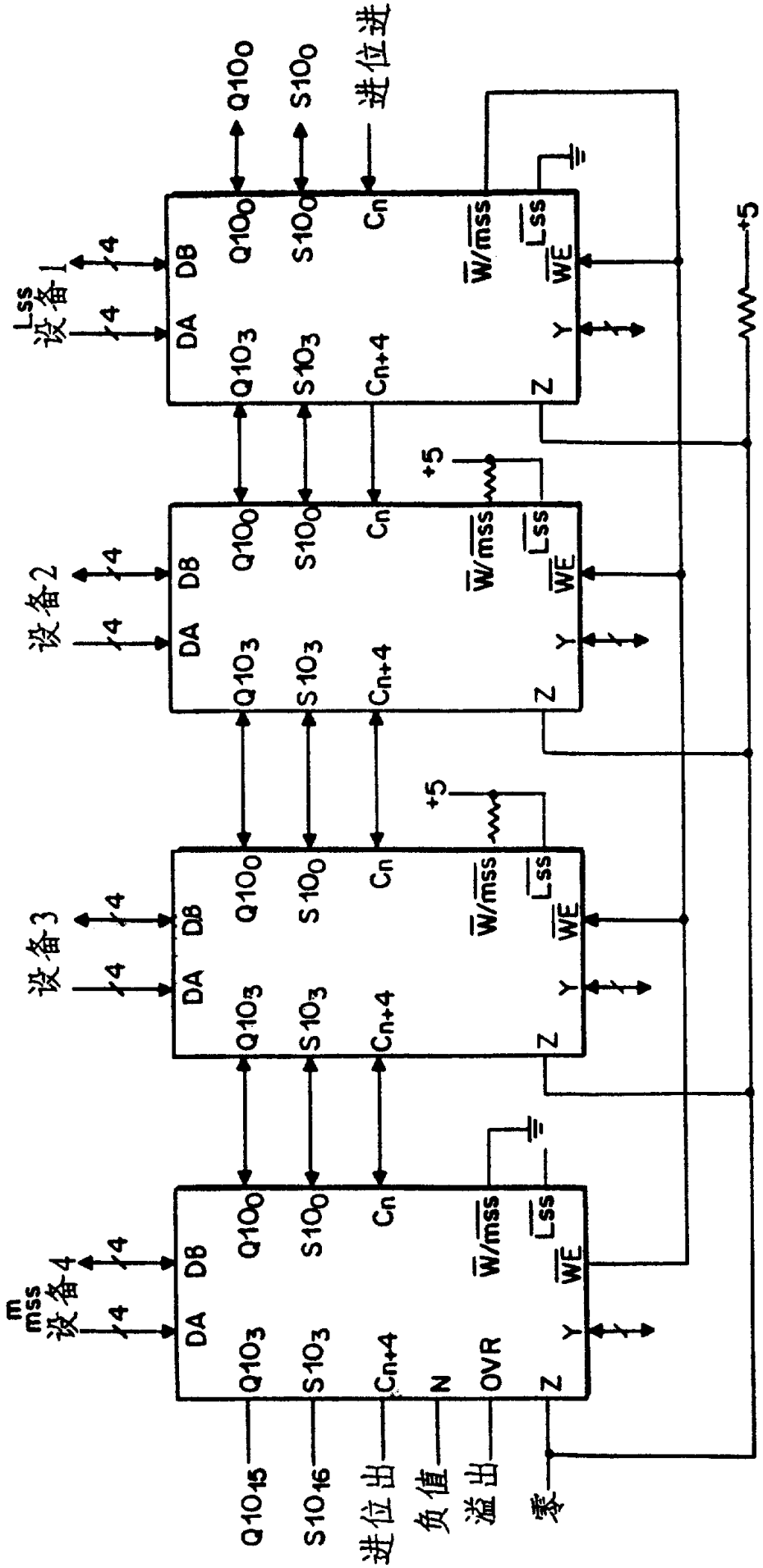

设备1
设备2
设备3
设备4
DA
DB
Q10₀
S10₀
Cn
Q10₃
S10₃
Cn+4
W/mss
Lss
WE
Y
Z
+5
进位进
进位出
负值
溢出
零
Q10₁₅
S10₁₆
N
OVR

图 9

16-位中央处理机简化框图

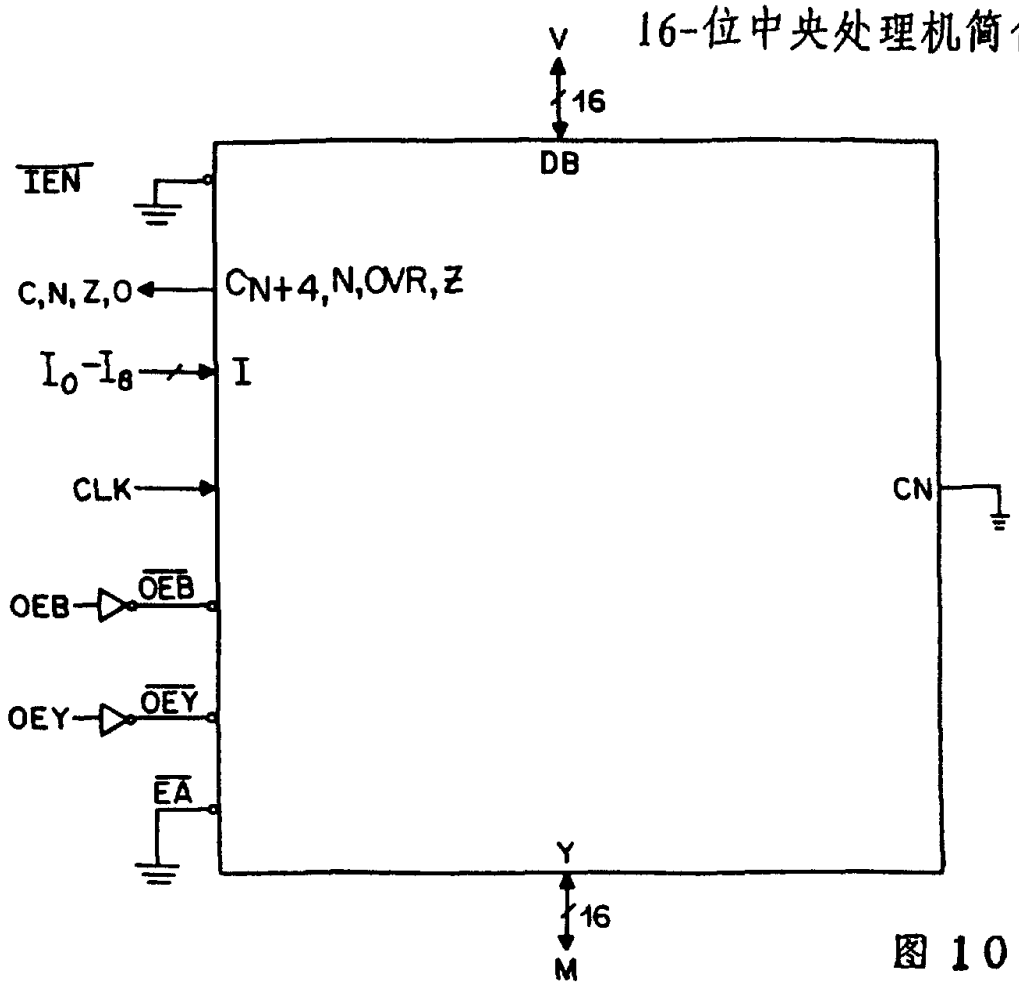


图 10

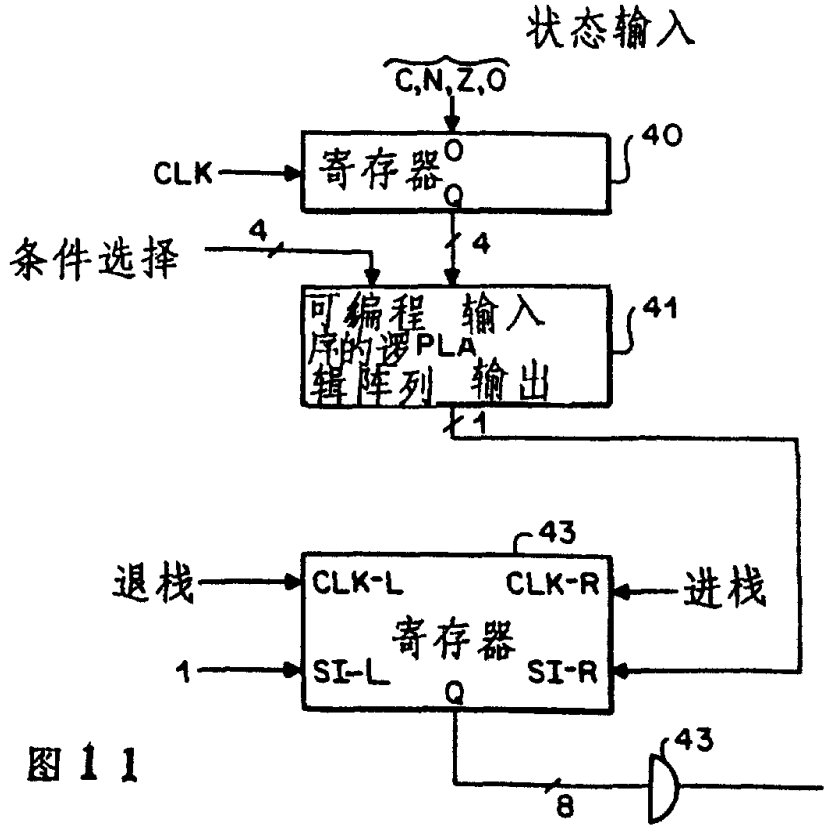


图 11

阵列处理机行框图

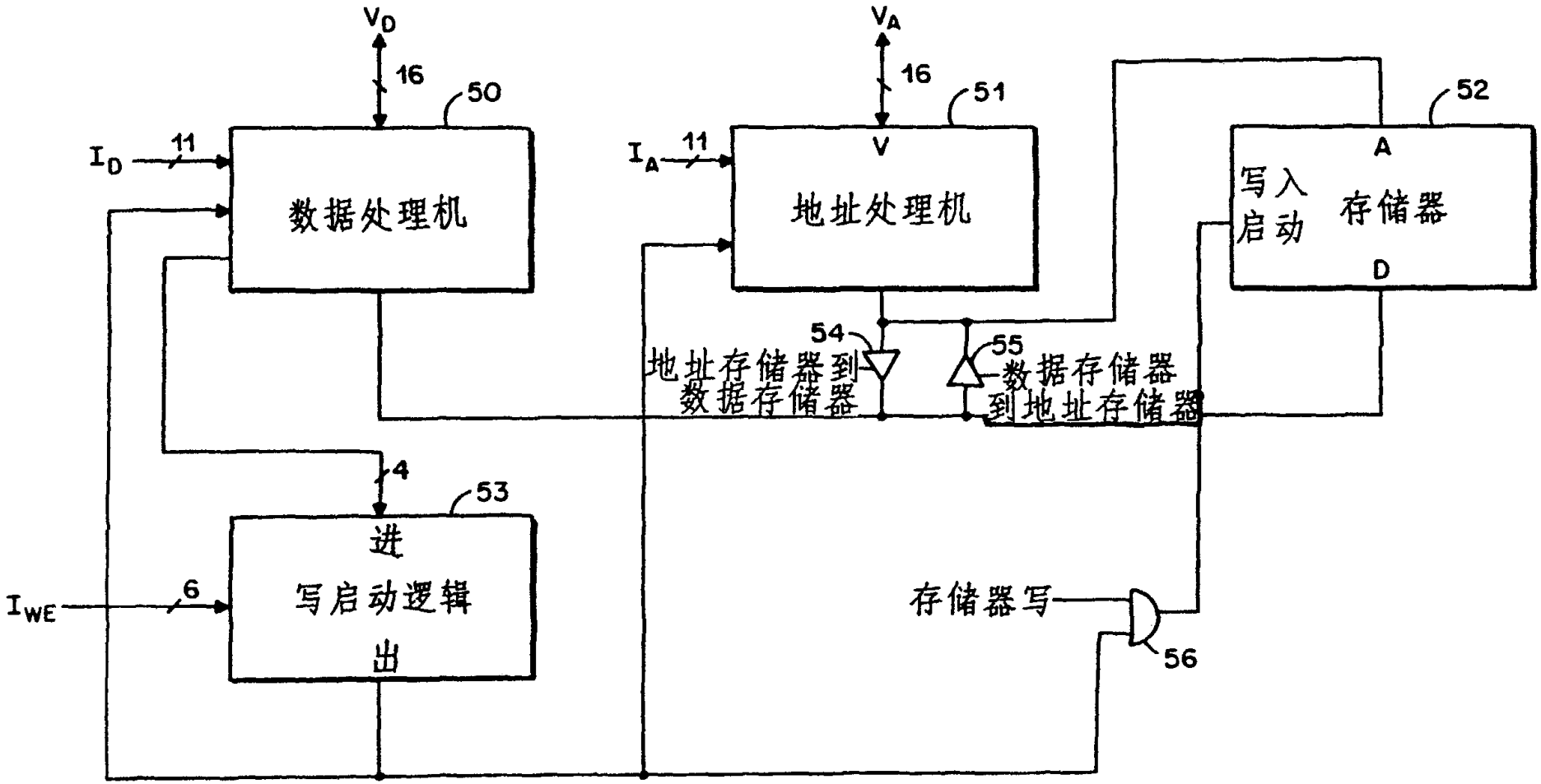
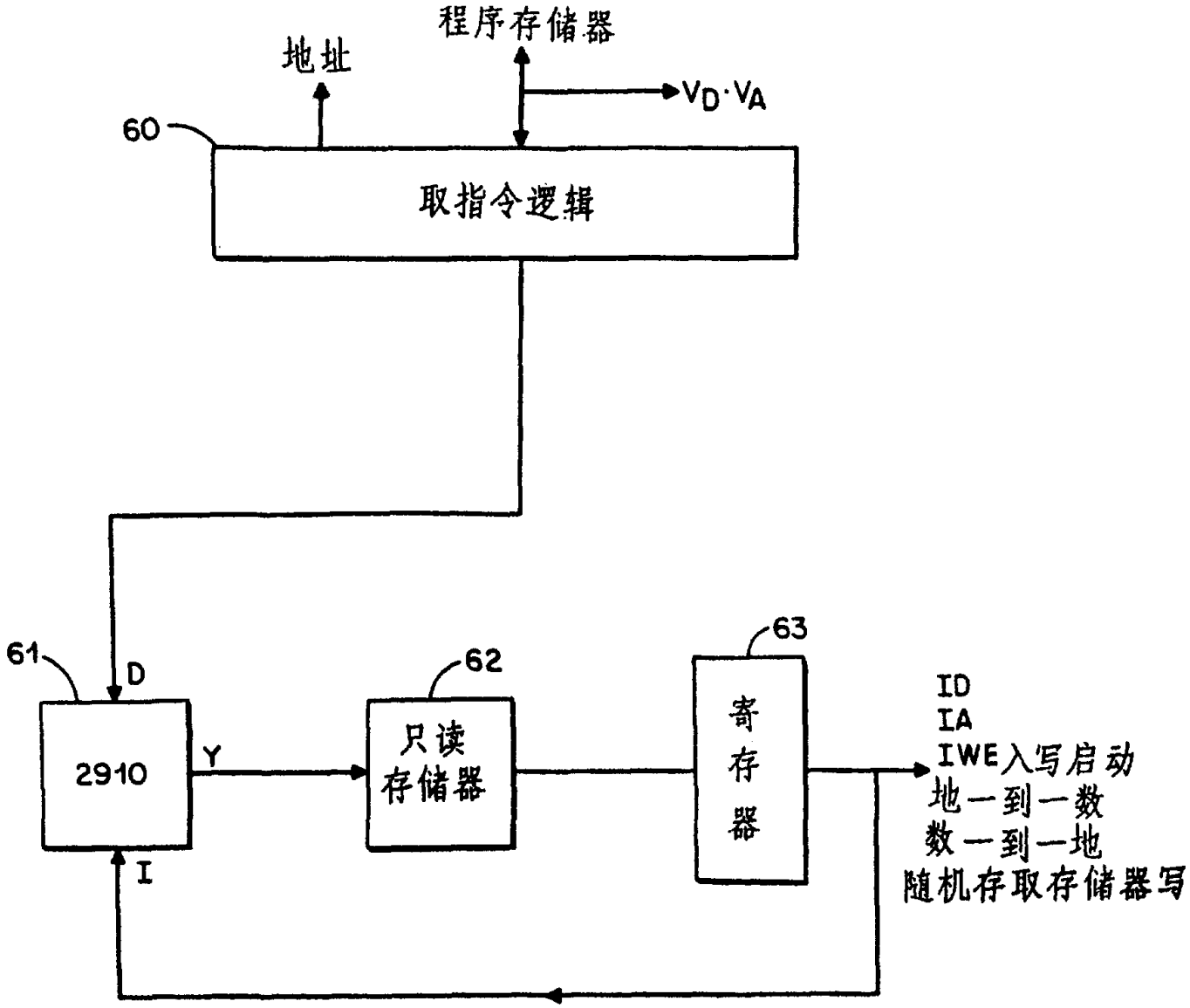


图 1 2



简化系统控制器框图

图 1 3