

[19]中华人民共和国国家知识产权局

[51]Int. Cl⁷

G06F 15/16

[12] 发 明 专 利 说 明 书

[21] ZL 专利号 91109637.X

[45]授权公告日 2000 年 3 月 29 日

[11]授权公告号 CN 1050919C

[22]申请日 1991.10.11 [24]颁证日 1999.11.13

[21]申请号 91109637.X

[30]优先权

[32]1990.11.13 [33]US [31]07/611,594

[73]专利权人 国际商业机器公司

地址 美国纽约

[72]发明人 詹姆斯·沃伦·狄凡德佛

彼得·米歇尔·考吉

保罗·安巴·威尔金森

尼古拉斯·杰罗姆

[56]参考文献

GB2223867 1990. 4.18 G06F13/12

审查员 夏国红

[74]专利代理机构 中国国际贸易促进委员会专利商标事
务所

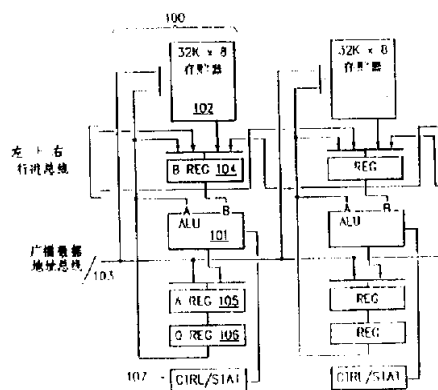
代理人 范本国

权利要求书 9 页 说明书 28 页 附图页数 5 页

[54]发明名称 并行相联处理器系统

[57]摘要

本发明提供了多处理器并行计算系统和字长串行 SIMD 处理器并行体系,从而能以一种适于在空气冷却环境下用芯片实现的简化体系实施并行阵列处理。本发明中提供的阵列是由字长宽度的处理单元构成的 N 维阵列,其中的每个处理单元都装有适当容量的字长存储器和控制电路。包含几个处理单元的阵列的可分离的一部分可以被包含在一块芯片上,处理阵列中的单元最好由处理单元与一个局部存储器组合而成。



权 利 要 求 书

1. 一种并行处理系统,其特征在于:多个阵元单元,每个阵元单元都有一位并行处理单元与一个局部存贮器进行组合,上述的局部存贮器与用于在所有的阵元单元中与相联方式对信息进行并行处理的处理单元相连,其中每个阵元单元适合于执行一个单元的相联处理。

2. 根据权利要求1的并行处理系统,其特征在于:上述的多个阵元单元的存贮器被设置成一个阵列,带有存贮器的阵元阵列中包括一个集合相联存贮器。

3. 根据权利要求1的并行处理系统,其特征在于:在一块硅基片上形成许多阵元单元,这些阵元单元构成一个阵列,毗邻的阵元单元并排设置,这样,在一块基片上就形成一个具有许多存贮器的阵元处理阵列,每个这种处理单元都可以是一个字节宽的列内容可寻址存贮器,上述这些存贮器被安排成一个逻辑行或者逻辑阵列,另外还有用于在阵元单元之间双向地传送数据以及把数据向外送出的邻元通讯通道。

4. 根据权利要求1的并行处理系统,其特征在于:一块具有许多阵元单元和存贮器的阵元处理器阵列存贮器芯片,在每个阵元

的处理单元与存贮器之间以及阵元单元之间都有数据流通道。

5. 根据权利要求1的并行处理系统,其特征在於:许多的阵元单元被安排成一个阵列,供数据在阵元单元之间流动的通道具有一一对应的存贮器,阵列中的每个处理单元的左边或右边都有相邻的阵元单元,滑动装置在相隔较远的单元之间提供一种滑动通讯。

6. 根据权利要求5的并行处理系统,其特征在於:滑动装置能在一个周期内通过阵元地址位置把信息转送到一个非毗邻单元,上述的阵元地址位置在信号到来之前对要传送的信息而言是不透明的,上述的信息将要由收到它的最近的工作着的阵元接收。

7. 根据权利要求6的并行处理系统,其特征在於:如果在传送周期之前通过把中介的阵元单元关掉的方法使中介单元成“透明”,那么滑动装置通过沿着“关断”的阵元向一个非邻位置送信息可以使能一种滑动功能,该功能使一个第一阵元单元向远处的一个阵元传送信息。

8. 根据权利要求6的并行处理系统,其特征在於:沿着格栅、二维网孔或者在阵列中的三维方向线性地传送数据的装置。

9. 根据权利要求1的并行处理系统,其特征在於:每个阵元单元都有一个能访问与之相连的局部存贮器的处理器;具有字符宽度、多倍字符宽度的数据和指令能在一个系统时间周期内在阵元单元之间流动。

10. 根据权利要求 1 的并行处理系统,其特征在于:每个阵元芯片都有自身的局部存贮器,这些存贮器在阵元单元阵列中提供了字符宽度的集合相联存贮器,为每个局部存贮器至少提供有 32 千字节的存贮器,有 16 个阵元单元被提供为线性子阵列的节点。

11. 根据权利要求 2 的并行处理系统,其特征在于:一根多字节宽的用于阵元单元之间进行通讯的广播总线,使所有的阵元在同一时刻看到同一数据;以及在这根广播总线传送阵元控制和地址传播传送。

12. 根据权利要求 2 的并行处理系统,其特征在于:阵元单元的局部存贮器可以是存贮器阵列中的 DRAM CMOS 存贮器,它通过删除存贮器后面的列多路分离就可支持行一列访问,并能提供一个能读出一行存贮器阵列而使数据并行流动行地址。

13. 根据权利要求 2 所述的并行处理系统,其特征在于:存贮器除了数据之外还能包含“第三个比特”或叫“第三态”,这样,逻辑电路可以识别出逻辑“1”,逻辑“0”和“忽略”;集合相联存贮器的存贮阵列中连续的存贮位置上都包含着“第三态”。

14. 根据权利要求 1 所述的并行处理系统,其特征在于:用于为阵元单元的单独运算提供控制功能的阵元单元控制单元。

15. 根据权利要求 1 所述的并行处理系统,其特征在于:局部存贮器具有多比特的二进制基准存贮地址。

16. 根据权利要求 1 所述的并行处理系统,其特征在于:一个

外部控制存贮和控制装置,该装置用于实现这样的控制功能,即中止状态输出满足一个特定条件的各个阵元单元中的操作的功能、忽略功能和禁止功能以及根据阵元单元中的条件对存贮器的使用能写以及在从上述的外部控制存贮器查询之后提供给阵元单元的控制功能。

17. 根据权利要求 2 的并行处理系统,其特征在于:阵元单元中的每个本地存贮及处理单元都具有字长转送装置以及输入用于查找存贮器中的信息的数据和掩蔽码的装置;其中还提供了用来对存贮器寻址从而执行检索的装置、以及用于输入用来在存贮器中的 N 个字中找出一个字的检索中使用的数据和掩蔽码的装置;其中在匹配位置上的一根匹配线的电位将升高,一个分离的操作将读取或选取第一个匹配,另外还配备有用于进行广播运算的广播装置,在上述的广播操作中,升高了的选中线将指示出参与状态,广播数据将被复制到所有选定的字位置上。

18. 根据权利要求 1 的并行处理系统,其特征在于:所述的多个阵元单元的局部存贮器包含一个或多个数据集,每个局部存贮器中含有每个数据集的远端部分,该远端部分包括一些存贮在可寻址位置上的数据,上述并行处理系统中还设有一条与上述的多个阵元单元相连的互连通路,该互连通路给每个阵元单元提供共同数据及相关的运算。

19. 根据权利要求 18 的并行处理系统,其特征在于:上述的互

连道路还提供一个地址，各个阵元单元可以使用该地址从局部存储器中得到一个数据集的远端部分。

20. 根据权利要求 1 的并行处理系统，其特征在于：每个处理单元包括：ALU、掩蔽寄存器、门锁电路以及与每个处理单元中的本地存储器成一一对应地相连的状态寄存器(SR)和数据流寄存器(DF)。

21. 根据权利要求 1 的并行处理系统，其特征在于：本地存储器为多位宽的 DRAM，每个阵元处理单元的逻辑电路与上述的 DRAM 本地存储器做在同一块硅基片上；在上述的局部存储器与其处理元件之间有直接的一一对应的连接。上述的本地存储器具有许多拥有多位地址的单元。

22. 根据权利要求 1 的并行处理系统，其特征在于：每个处理单元都配置有多个掩蔽码寄存器和一个门锁电路，设置上述门锁电路是使之为每个处理单元取得沿着一条上述的多个处理单元公共的通讯线的连接端口的作用。

23. 根据权利要求 1 的并行处理系统，其特征在于用来控制该系统中各个阵元处理单元的状态的外部控制定序器和本地控制寄存器装置。

24. 根据权利要求 23 的并行处理系统，其特征在于：上述的多个阵元具有自制权，也可以被设置成一个由上述的阵元单元形成的相连单元阵列。

25. 根据权利要求 24 的并行处理系统,其特征在於:上述的系统可以被设置成 *SIMD* 和 *MIMD* 系统,许多组上述的多个处理单元被赋予一个可编程结构,其中,系统的各个单元都有一定程度的局部自制权。

26. 根据权利要求 23 的并行处理系统,其特征在於:一个主处理器系统,该主处理器系统通过一条总线与上述的外部控制定序顺进行通讯,该主系统还具有使全局指令被上述的并行处理系统执行的装置。

27. 根据权利要求 23 的并行处理系统,其特征在於:上述的外部控制定序器与一个微码存贮器相连,通过封装的功能程序可以对上述的微码存贮器进行编程。

28. 根据权利要求 26 的并行处理系统,其特征在於:上述的定序顺被提供有用于控制通过总线与上述定序器相连的处理单元的功能的高函数宏对系统中的本地存贮器进行的存贮器寻址能使字长宽度的数据流和模 8 算术逻辑被用于逻辑、加乘和除函数,对于上述并行单元中的浮点运算以及通过对各个阵元处理单元进行分别寻址实现的单独的“睡眠”及“打盹”状态也作了准备。

29. 根据权利要求 26 的并行处理系统,其特征在於:函数的运算可以通过程序定位,可以在上述主处理器系统,具有封装程序的上述外部控制定序器、上述局部存贮器装置或者在上述的多个阵元处理单元中进行;需要对多个数进行扩展的处理的单独指令被赋

予配置成 SIMD 处理的上述的多个处理单元。

30. 根据权利要求 24 的并行处理系统,其特征在於:系统中的处理单元被连接成处理单元阵列,该阵列中的处理单元具有可编程的局部自制权;每个处理单元可以在它自身的指令流控制下工作,也可以根据数据条件进入或者退出与其他处理单元相连的操作;该系统中的各个处理单元可以独立执行与浮点运算有关的定位和正规化运算。

31. 根据权利要求 23 的并行处理系统,其特征在於:上述的外部控制定序顺可以被设置成可以通过一个封装程序运行库提供的封装程序执行 *CALL()*, *Load*, *Block*, *Sin*, *Cos*, *Find*, *Min*, *Range*, 及 *Matrix Multiply* 等运算。

32. 根据权利要求 31 的并行处理系统,其特征在於:上述的封装程序用被提供用来执行控制、*Load*、*Read*、*Add*、*Multiply* 和 *Match* 等操作。

33. 根据权利要求 2 的并行处理系统,其特征在於:处理单元可以在一个相邻处理单元的阵列中以及在系统中的各页面之间通过一个互连网络交换数据。

34. 根据权利要求 2 的并行处理系统,其特征在於:为阵元阵列提供有一个外部控制处理器;微控制码被并行传送至上述阵列中一个阵元处理器组中的所有阵元中;上述的控制处理器与阵元处理单元和相同的时钟系统保持同步,这样,由外部控制处理器控

制的函数能在相同的时钟时间内执行。

35. 根据权利要求 34 的并行处理系统,其特征在於:在上述的阵元阵列的外部控制器上方有一个通过微通道总线与该外部控制处理器相连的总控系统处理器,该总处理器管理着阵元单元的活动;该系统能连接成这样,即在某一给定时刻,阵列中的所有阵元处理单元执行着相同的指令,虽然处理器的子集可能对控制流有不同的反应。

36. 根据权利要求 1 的并行处理系统,其特征在於:许多个阵元并行处理单元沿着一根地址总线排列,与一个外部阵元控制器进行通信;阵元系统中的各个本地存贮器都有相同的矢量地址;一个用来控制什么数据通过阵元系统的每个本地存贮器和控制与阵元控制器的通讯的数据矢量地址寄存装置;各存贮器共有一根矢量地址总线,数据矢量地址寄存器控制着什么数据通过每组关联存贮器。

37. 根据权利要求 2 的并行处理系统,其特征在於:许多个阵元单元被安排在一个阵列中,为数据在具有通讯路径的阵元单元之间流动提供通道,上述的通讯路径为阵列中的阵元与其他阵元进行多位通讯提供了方法和装置;同时还提供了用来从一个阵元向其他任何阵元传递信息的连接路径。

38. 根据权利要求 37 的并行处理系统,其特征在於:通过用线或点式连接的阵元构成的 N 维阵列可以把信息从一个地方传送

到另一个地方。

39. 根据权利要求 2 的并行处理单元,其特征在於:一小部分的本地存贮器可以这样设置,即该部分中每一位置上都可以对一个给定码型作匹配比较。

并行相联处理器系统

总体上讲，本发明涉及多处理器并行运算系统，更具体地说，涉及一种字节串行SIMD处理器并行结构以及通过一种适合于在空气冷却的环境下进行芯片实施的简化结构进行并行阵列处理的系统和方法。

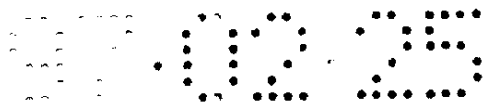
1970年10月27日批准的授予R. A. 斯多克(Stokes)等人的第3,537,074号美国专利中描述了一种阵列计算机，它包括：多个并行处理器，一个单一的可编程控制单元，用于存贮配对物矢量的寄存器，掩蔽寄存器，以及对一个或多个控制单元指令构成的序列作出响应而对当前在矢量寄存器中的数据进行运算的装置。在整个七十年代，斯多克发明的并列处理器被称之为“SIMD（单指令多数据）”处理器。这些装置也可以被认为是一个可编程控制单元驱动由几个并行处理器构成的一个阵列，其中每个处理器都具备存贮部分、算术单元、程序译码部分以及输入/输出部分。这些系统是一些大“盒子”，有时候与主计算机相联。SIMD与更普遍的处理器的一个重要区别在于：在这种系统中的所有SIMD处理器在其对应的处理器中都可以有一套不同的数据，同时所有的处理器又由一个共同的控制器加以管理。SIMD计算机与一般的凡·纽曼(Von Neumann)处理器也有不同之处，即每条指令对一个数据矢量而不是对一个单独的操作数进行运算。

最常见的多处理器系统是“多指令多数据系统”，其中每个处理

器都可以对一组单独的数据执行一个单独的 程序。一个 M I M D (多指令多数据) 系统中的处理器既可以执行不同的任务, 也可以执行一个共同的主任务中的不同子任务。

当人们在研制 S I M D 并行处理器时, 人们发现: 正象授予罗尔 (Lorie) 等人的名为 “ S I M D 矢量处理器中条件转移的执行方法 ” 的第 4, 435, 758 号美国专利中所述的那样, 当系统要执行的各任务之间高度独立且不争用信息时, S I M D 并行处理器是非常合适的; 但如果各任务之间争用信息时, 则可以指示出一种在 S I M D 状态下工作的同步处理器网络。实际上, 第 4, 435, 758 号美国专利中描述了一个问题, 并且还讲述了他们为了提供条件转移的执行而不得不对 1978 年 7 月 18 日授于 R. A. 斯多克斯 (Stokes) 等人的第 4, 101, 960 号美国专利所作的改进。

人们已习惯上把最高级的 S I M D 机描述为设置成一个 $N \times N$ 矩阵的同步位串行处理器, 正象 H. 李 (Li) 和 M. 马莱斯卡 (Maresca) 在 “ 国际并行处理大会会刊 (Proceedings of the International Conference on Parallel Processing) ” 中第 411 至 414 页的一篇题为 “ 多环网络 (Polymorphic - Torus Network) ” 的论文中描述的示例性的多处理系统那样 1989 年 8 月 3 A 号的 I B M 技术手册第 32 卷中详细描述了用于提高稀疏矩阵与矢量的相乘执行速度的、适用于 H. 李等人描述的通过一个网孔布局和一个被另一个用于重布目的的交换网络所覆盖的网状网络而物理地连接的大规模并行结构的矩阵矢量的相乘方法。



1990 年的 IBM 技术手册显示，人们曾想制造能通过多个处理器行进行运算的 SIMD 和 SIMD/MIMD 机器，其中某一行中的所有处理器执行完全相同的指令，但各行中的程序是完全不同的。从 1990 年 1 月号 IBM 技术手册第 32 卷 8 B 中 H. 李、B. 多姆 (Dom) 及 R. 杰夫 (Jaffe) 等人的示例中即可看到一个具有这种结构的一个体系，叫做并行本地运算器 (PLOE)，用于执行具体的竞争存贮器检查功能。

本申请中设想建立一种可用现有技术实现的空气冷却系统，它对上面建立的体系中的芯片组进行工作。

如果对现有技术作一番后见之明的研究，可以发现也有其他人描述了具有一个用于静态指令的 ROM 和用于在一块单独的硅片上存贮和访问数据的处理器的使用，1990 年 7 月 17 日授予 G. 哈爱特 (Hyatt) 的名为“单片集成电路计算机结构”的第 4,942,516 号美国专利就是一例，但是这类处理器与 SIMD 类型的复杂应用无关。

另外，还有人描述了不少用于执行不同任务的装置，我们的系统可能涉及它们。比方说，矩阵相乘可以并行，这是公知的。也有一些出版物涉及到可能使用我们的系统的、属于人工智能方面的应用。内容可寻址存贮器 (见图 3) 或者关联存贮器可以使用各种处理芯片在很高的水平上被寻址。另外，申请人还发现，在某些情况下，用行逻辑电路根据前面的检索操作进行选择是非常有用的。参看“用于人工智能的 VLSI”一书，编辑者：Jose G. Delgado - Frias 和 Will R. Moore, Kluwer 科学出版社，1989，第 95-108 页；以及 Peter Kogge, Mark Brule 及 Charles

Stormon 著的讨论应用的“VLSL 与基于规则的系统”一书。但是，也有其他人的一些巧妙的建议似乎要把发展方向引向不同于我们所遵循的方向。其中一例是由 Bob Cushman 主编的一本名为“用大量并行实行矩阵压碾 (Matrix Crunching With Massive Parallelism)”的著作中的“VLSI 系统设计”，1988 年 12 月，第 18—28 页中描述的用于大量并行的牛津“智能存贮器”。也有人把并行处理试用于 VLSI 相关存贮器，试图描述一种适用于建立一种可重配置的完全并行相关的处理方案的 VLSI 相关存贮器芯片。这个方案认为：使用传统的相关存贮器结构的话，为了进行数据转移，还需要一定数量的输出点 (Pinout)。请参见报告“VLSI 关联存贮器的并行处理”，作者 S·亨根 (Hengen) 和 I·斯切森 (Scherson)，该报告由 NSF 奖 # ECS—h 404627 加以支持，由加利福尼亚大学 (Santa Barbara, CA 93106) 电气与计算机工程系工作的作者加以报道。

我们面对的问题是制造结构紧凑且又有多种应用的处理器。在实施这个总目标时，我们发现：以前的设计局限于每个处理单元最高几位比特的存贮器以及每个电子芯片上装几个处理单元的比特串行方法，这是不适当的。结果，我们想到要设计一种结构，它可以以极高的密度制造，并适合于空气冷却环境，同时能被封装在轻而小的单元中。最后，我们找到了解决方案，产生了一个 N 维字长宽度处理单元阵列，阵列中的每个单元与字长宽度存贮器中适当的段和控制逻辑相连。含有多个处理单元的阵列的一个可分离的部分装在一个硅片中。这在下面将加以详述。我们自己的解决方案也产生为提高系统性能而必须解决的附加问题，另外我们还发明了一些进一步开发我们提

出的系统的一些方案，其中的一些方案允许处理单元进行独立操作。在回顾先有技术时，我们注意到欧洲专利申请EP-A-208,457，该申请讲述了一种处理器阵列，其中，阵列中的每个处理单元能够选择从中取得输入的单元。

杰姆斯L. 泰勒(Jame L. Taylor)于1990年5月4日递交的07/519,332号美国未决申请以及1988年9月27日递交的07/250,595号美国专利申请题为“SIMD阵列处理器”，其中描述了一个多维处理器阵列，该阵列在使并行处理的潜力被更好地开发而又不增加MIMD处理器的开销及复杂性方面具有更高的灵活性。该专利申请最初于1989年5月3日以欧洲申请EPO#88/307885/88-A的形式被公开，其中描述的系统沿着各个并行处理单元中与控制逻辑相连的总线送出一个用于进行局部比特串行执行的全局指令，并且通过程序修改全局指令中的一定的位而用于局部比特线上，在局部比特线上，这些修改后的位将被加以译码。

对先有技术进行回顾时还发现一篇对具有字长宽度SIMD的处理单元所作的讨论，那就是C. R. Jesshope和R. O. Gorman等人出版在IEEE会刊1989年5月第136卷中的“SIMD微处理器阵列的设计”一文。该论文中讲述了这二位作者对SIMD结构所作的学术研究，这二位作者执教于英国南安普顿(Southampton)大学电子与计算机科学系。论文中的有些特征与我们讨论的近似，作者在论文中描述了一种具有字节宽度SIMD结构的处理器。该论文建议，处理单元除了具有一定程度的自制能力之外，还应有一个字长宽度的处理累加器结构单元，仅受电路板限制

的RAM (1K 字节), 以及每个芯片 (4) 上有多个PE。但是, 即使使用了上面建议的这种结构, 仍不能提供相联处理。原因在于在上面结构中没有提供字节宽度的邻元通讯, 而本发明中做到了这一点。我们开发的其他一些特征该论文中也没有提到。因此, 迫切需要一种象我们的阵元系统那样能推进并行处理技术进步并能在一块集成电路芯片上实施的系统。请参阅R. Duncan的“并行计算机布局概述 (Summary of Parallel Architectures)”一文, 从1990年2月份的IEEE “Computer (计算机)”杂志上即可找到。

定义:

为了便于下面叙述, 下面对本领域内的几个新术语作一个广义的、非限制性解释:

阵元 (Picket) - 最好由处理单元与局部存贮器组合而构成的处理阵列中的一个单元, 上述的阵列能在一个时钟周期内处理信息中一个字长的串行数据位。其最佳实施例由一个字长宽度的数据流处理器, 32K字节或更多的存贮器, 原始控制电路组成, 与其他阵元有通讯线相连。

阵元芯片 - 在一块单独的硅片上包含许多阵元。

阵元处理器系统 (或叫子系统) - 由阵元阵列、通讯网络、I/O系统、由微处理器构成的SIMD控制器、封装程序处理器以及一个管理着整个阵列的一个微控制器构成的总系统。

阵元布局 - 能解决下列几类不同问题的SIMD布局的最佳实施例:

—集合相联处理 —并行数值集中处理 —与图象类似的物理阵列处理。

阵元阵列—以几何级数设置的阵元集元。

阵元(Picket, 原意“木桩”)一词出自汤姆·索耶和他的白栅栏,也可以从功能上理解为一队士兵警戒线更合适一些。

我们将要描述的体系可以被称之为“阵元”体系。该体系中结合了某些在结构上与早期的SIMD处理单元(PE)相似的元件,这些元件根据需要组成一条警戒线一样连在一起,这些元件已被进行了改进,能执行独立的凡·因曼运算。在建立该体系时,我们采用了一个N维处理单元阵列进行并行运算,在我们称为“阵元”的每个处理单元内都有控制逻辑电路。

本系统能象单指令多数据(SIMD)机那样进行运算,但其性能由于装入了更多的并行处理元件而得到很大提高。有关数据依赖性方面的问题已被解决。在SIMD机的运算中,任何处理器或功能电路都不能具有将可能使一个处理单元要求不同数量的周期的数据依赖性。

我们进行的体系开发产生这样一种系统,该系统具有许多带有一个并行处理单元的阵元单元,并且具有与上述处理单元相连用来以相联方式进行信息的并行处理的局部存贮器,每个阵元都适于执行一个单元的相联处理。我们提供了每个阵元的水平相联方法。阵元单元的存贮器被设置成一个阵列。因此,形成的阵元阵列还包括一个相联存贮器集合。

把一个集合相联并行处理系统做在一块芯片上的发明能使一个较大集合中的较小集合的“数据”被从存贮器读出,从而可对此执行相联运



算。这种相联运算（通常是相同比较）通常使用阵元的存贮器和执行单元在整个数据集中并行进行。

在阵元阵列中，每个阵元都有一部分出于一个大集合的数据。此外，每个阵元还从它的数据部分中选择一块数据。这样，每个阵元集合中的数据块都包含着所有的阵元都将并行地对之进行相联运算的数据集。

目前能实现的设计是将最多 50 个数据流门电路和控制逻辑电路及多兆位的 D R A M 存贮器做在一个芯片上。每一片这种芯片都能设置成用来容纳许多阵元或者处理单元。在进行文本处理（也能进行图形应用）的最佳实施例中，一个单独的阵元芯片上有 16 个阵元，每个阵元还带有 32 K 字节的 D R A M 存贮器，整个系统包括一个由 64 块这样的芯片构成的阵列，从而产生 1024 个处理单元。这种阵元体系可以用 C M O S 技术来制造（这种技术能使 400 万位的 D R A M 以重复的图形形成在一块芯片上），芯片上的其余芯片表面可以用逻辑标准单元来填充，这些逻辑标准单元将采用高达 50 K 逻辑元件，能在芯片上形成阵元所需的数据流和控制逻辑电路。在我们安排的系统中，阵元处理数据时具有一定局部自制权，并在阵元之间提供了“滑动”。

阵元技术是可扩展的，每个单独的阵元有 126 K 字长 D R A M（16 兆位的 D R A M 芯片）的话，阵元体系可以以和文本相同的方式处理全部 24 位的彩色图形。用目前的实施例可以处理 8 位彩色或者灰度的图形。实验性的制造技术显示出，只要产品能在一个空气冷却的环境下工作，那么，这个密度在很近的将来是可以期望得到的。对于彩色图形，我们的最佳阵元体系将把芯片上的 D R A M 数量

增加到每阵元 128 千字节，每一芯片上保持 16 个阵元。另外，对于全颜色图形处理器而言，也可以采用每块芯片上具有 24 个带有 96 千字节存贮器的阵元单元的方案。

下面根据附图来描述我们的最佳实施例。

附图中

图 1. 是一个近期的 SIMD 处理器的示意图，它描述在一份未决专利中，可以被认为是原有技术的代表。

图 2. 示出做在一块硅基片上的一对基本阵元单元，它们具有：处理器、存贮器、控制逻辑，与阵列中的其他成员具有字长通信的相关存贮器。

图 3. 中示出了相关存贮器处理。

图 4. 示出了一个 SIMD 子系统的基本 16 (n) 成员块结构，该子系统采用一个微处理器控制器，一个用于封装 (Canned) 程序的硬连线序列控制器，和一个成员块阵列；这构成了基本并行阵元处理器系统。

图 5. 示出了一个结合了多个图 4 所示的阵元处理器的一个多阵元处理器系统。

图 6. 是一个子系统的功能框图。

图 7. 进一步表示了一个子系统控制器的结构。

现在来看附图。图 1 中示出了一个典型的现有 SIMD 系统，其类型大致为泰勒 (Taylor) 在 IBM 欧洲专利申请 (第 88307855/88-A 号) 以及 UK-A-1,445,714 中描述的那些。在这种先有技术装置中，SIMD 计算机是单指令多数据计算机，具有一个并行阵列处理器，该并行阵列处理器包括许多

个并行连接的比特串行处理器，每个这样的处理器又与许多 SIMD 存贮装置中的一个相连。输入/输出 (I/O) 系统给 SIMD 单元起到中间集结的作用，该系统包括一个用于给主计算机 (可以是一个主框架或者是微处理器) 和 SIMD 计算机之间进行的双向二维数据交换提供暂存的暂存器。I/O 系统还包括输入输出处理装置，用于控制主计算机和暂存装置之间的数据流，以及控制暂存装置与许多 SIMD 存贮装置 (它们是通常结构的缓冲器部分或者一个大存贮器的一部分)。I/O 系统的输入操作涉及把数据从主计算机存贮器转送到暂存器以及第二步从暂存器送到 SIMD 存贮器装置。输出时也是一个二步骤过程，沿着二维总线在主计算机和 SIMD 计算机之间转送数据。用于 I/O 传送的输入/输出单元既可以是一个分离的单元，也可以是主计算机中的一个子单元，或者 (经常也是) SIMD 计算机中的一个单元，这里，SIMD 控制器起到控制暂时 I/O 缓冲存贮的作用。

SIMD 计算机本身包括一个具有许多处理单元的处理器阵列以及一个把各个处理单元与许多常规的分立 SIMD 存贮装置相连的网络。SIMD 计算机是一个具有大量的并行连接和操作的单独的处理单元的并行阵列处理器。SIMD 计算机中包括一个控制单元，由它产生送给处理单元的指令流，以及给计算机提供必须的时序信号。互连着各个处理单元的网络包括好多可以使各个处理单元相连的方案，如可以采用如网孔、多型环以及超正方体等多种拓扑结构。上面提到的许多存贮装置用于给各个处理单元立即存贮比特数据，在处理器数量和存贮装置 (可以是上面提到过的大存贮器的缓冲器部分) 的数量之间存在一一对应的关系。

举个例子来说，如图 1 所示，有一台主处理机 28。该处理器用于通过主机—控制器数据总线 30、和地址和控制总线 31 把微码程序装入阵列控制器 14（它包括一个暂有缓冲器），交换数据，以及监测它的状态。本例中的主处理器可以是任何适当的通用计算机（如主框架或个人计算机）。在这个先有技术实例中，处理器阵列画成二维的。实际上，阵列可以有不同的构造，如三维或四维集束结构。

SIMD 阵列处理器包括一个处理单元 $P(i, j)$ 阵列 12，以及一个用于给处理单元 $P(i, j)$ 发出全局指令串的阵列控制器 14。虽然图 1 中未示出，但该先有技术实例中具有一次只对一位进行运算的处理单元，以及一个存储块（它可以是与该处理单元相连的存储器中的一部分。各个处理单元通过所谓的 NEWS（北、东、西、南）网络通过各向比特线与它们的各个邻居相连。这样，一个处理器单元 $D(i, j)$ 将分别与它们北、东、西、南方向上的处理单元 $P(i-1, j)$ ， $P(i, j+1)$ ， $P(i, j-1)$ 和 $P(i+1, j)$ 相连。在这个典型例子中，NEWS 网络的边缘成回环状连接，这样，北边缘和南边缘或双向互连，西边缘和东边缘也类似互连。为了能使数据输入处理器阵列以及从处理器阵列输出，一根控制器—阵列数据总线 26 与 NEWS 网络相连，也可以通过与环形的东西 NEWS 连接线相连的双向三态驱动器与或者附加地与北—南边界相连。正象在最佳实施例中将要描述的那样，如果本例中的处理单元数为 32×32 ，而不是 16×16 ，那么，用先有技术可以实现 1024 个处理元件。在图中，单线表示单一的比特线，而连接功能单元的双线则被用来表示多根连接线或总线。

在本实例中，阵列控制器通过指令总线 18 向各处理单元发出并

行的指令，分别通过行选择线 20 和列选择线 22 发出行选择和列选择信号。这是指令使得处理单元从存贮器装入数据、对数据进行处理，然后再一次把数据存贮到存贮器中。由于这个缘故，每个处理器单元都要访问主存贮器的比特“片”（主存贮器的一部分或者缓冲器）。因此，从逻辑上说，阵列处理器的主存贮器为了一个 1024 处理单元阵列而划分成 1024 个部分。这意味着在传送步骤中，最多 32 个 32 位字可以一次送入或送出存贮器。为了执行读或写操作，要根据变址地址对存贮器进行寻址。上述的变址地址通过地址总线 24 送至存贮器地址线，读或写指令被并行送到每个处理单元中。在执行读操作期间，行和列选择线上的行和列选择信号指示出哪个处理单元要执行该操作。这样，在该实例中，要把一个单独的 32 位字从存贮器读， 32×32 的阵列中选定行上的 32 个处理单元中是完全可以的。处理单元与一位宽的存贮器条或块 (i, j) 相连，虽然这条或者说这块存贮器一一对应地与相应的处理单元逻辑相连，它有可能而且典型情况下就是物理上分装在另一个芯片上。我们不知道怎样用现有的布局技术制造上述的那种阵列处理器，我们的阵元可以通过处理器阵列及适当的存贮器做在具有我们下面将要描述的那类芯片上。

应该理解的是，上面那个先有技术实例中的众多处理单元包括：AL11 和带进位的输入和输出寄存器，每个进行都能存贮一位信息。还有一个多路转换器与 AL11 输入、输出相连，以及与和该处理单元 $P(i, j)$ 相连的那条存贮器 (i, j) 的双向数据端相连。

指令总线与数据总线是分离的，阵列控制器有一个微码存贮器，

表示将要被阵列执行的处理的微码由主计算机 28 通过数据总线 30 以及地址和控制总线 31 装入。一旦主机 28 启动阵列控制器的运算，微码序列即由阵列控制器 14 内部的一个与上述的微码存储器相连的微码控制单元相连。AL11 和阵列控制器中的寄存器列将被用来产生阵列存储器地址、循环计数、跳转地址计算以及各种目的的寄存器操作，它们都将被输出到阵列控制器的一根地址总线上。该阵列控制器还有多个用于对行和列掩蔽码解码的参考寄存器，特殊操作码通过一根指令总传送到处理单元。在本例中，阵列控制器还可以有一个位于控制器内部但功能上却位于主控制数据总线和控制器阵列数据总线之间的数据缓冲器。在控制存储器中的微码的控制下，数据被从该缓冲器中装入处理器阵列中，或者相反。由于这一缘故，该缓冲器被设置成处于阵列控制器中的一个微码控制电路控制下的双向 FIFO 缓冲器。有关这种先有系统的其它细节可以从上述的那个实例特别是从泰勒 (Taylor) 的名为“SIMD 阵列处理器”的 US N 07/519,332 中发现。

上面作的对现有技术的尝试的回顾可以和下面讲述的本发明的最佳实施例加以比较。图 2 中示出了基本阵元单元 100，它包括一个组合处理单元 AL11 101 和一个与该处理单元相连从而在一个时钟周期内处理一个字长的信息的局部存储器 102。如图中所示，阵元单元以线性阵元阵列的方式成形在一块硅基片或者叫阵元基片上，毗邻的单元与它的边缘（即附图的左边或右边）相邻，这样，在一块硅片上可以形成一个阵元处理阵列和多个局部存储器，对于每个设置在逻辑行或者线性阵列中的字长宽度的处理数据流都设置一个局部存储器，它们与毗邻单元的通信总线一起用来向左或向右双向地传

送数据。一个阵元芯片上的阵元集最好以几何图形设置，最好在芯片上的阵元阵列中二个阵元的典型设置情况，其中还有许多存贮器及数据流，还包括每个阵元的处理单元及存贮器之间的通信通道。在本实施例中，一对一存贮器和阵列中的处理单元之间的数据通信通道为一个字长宽度，从左边或右边进入毗邻单元。通过一次“滑动（slide）”与远处的阵元处理器进行通讯。

“滑动”可以被定义为一种在一个周期内把信息传送到一个非毗邻位置的手段，它在正常情况下能接受信息，它在信息到达之前是不能通过的，上述信息被最近的工作着的毗邻单元所接收。这样，通过把信息穿过被“关掉”的阵元送至一个非毗邻位置，滑动就起了作用。现在，假设阵元“A”想把信息传送至一个较远的阵元“G”。在该周期之前，先通过把“B”至“F”阵元关断，把隔断的阵元变得“透明”。在下一个周期中，“A”把它的信息向右送出，信息穿过“B”至“F”，这些单元是透明的，因为它们已经被关断了。最后“G”接收到该信息，因为它还被“打开”着。在正常使用中，“滑动”信息是沿格栅或直线传送的，但在二维网孔结构或多维阵列中，这种滑动方法还能工作。

在我们的最佳实施例中，处理单元的访问不是以串行比特而是以字长串行进行的。每个处理器访问的是与它自身相连的存贮器，而不是去访问一块局部存贮器或其相应的部分。提供的总线不是一位宽度，而是一个字符宽或者是字符的几倍宽。在一个时钟周期内处理的不是一位而是一个字长的信息（在将来的系统设想把处理能力提高到一个字符字长或者几倍字长）。这样在每个阵元处理单元之间能有8，16或者32位信息流动，从而与相应的存贮器的宽度相对应。

在本最佳实施例，每个阵元芯片具有32千字节的存贮器，宽度为8（9）位；最好有16个阵元有32千字节的存贮器。这样使得线性阵列上的每个阵元节点都有一个存贮器。在本实施例中，可以发现每个相关的存贮器是CMOS构成的DRAM，一个字符字长为9位（它起的作用为带有自检功能的8位字符）。

并行通道上具有字节宽度的总线数据在阵元间和处理单元之间流动，这里的存贮器是对先有技术系统中的串行位结构的一种实质性改进。同时也应认识到，并行程度提高了，也产生了需要解决的其它问题。这些问题只要在理解了新构成的结构的含义就能联想到。下面讲述一下对这些问题的一些重要解决方法：

应该理解的一个特征是，我们给上面讲述的滑动机制除了提供左、右毗邻转换外，还提供一根具有双字长度的广播总线，使得所有的阵元在相同的时刻能得到相同的数据。阵元控制信号及地址传播信号也在这根总线上传播。在进行集合结合运算和其它比较及同步数字运算时，正是这根总线提供比较数字。

具有高度并行的数据结构从而适合于在阵元数据处理单元中在单指令流的控制下进行运算的任务有：从智能图形匹配方面的应用、多传感器最佳对准中的传感器及痕迹汇合、内容查寻以及图象处理等等。但是，现在能进行的许多这些应用将不会用SIMD进行处理，这是由于在一个单独的时钟时间下进行串行比特处理的缘故。举个例子来说，SIMD机中传统的串行处理单元在每个处理器周期期间执行一比特的加法运算，而一台32位并行机在一个周期内能执行32位的加法运算。

每个处理器单元32KB的结构能比传统的SIMD机提供更多

的与每个处理单元逻辑相连的存储器。

芯片的针脚数可以保持很低，这是因为要送入和输出芯片的数据被保持在最少程度。D R A M存储器是一种常规的存储器C M O S阵列，通过在存储器阵列之后删除列的多路分解以及提供以一个存储器阵列行中读出的行地址，它还能支持“行—列”访问。

除了数据之外，存储器也包括“三态位”或“三态”，这样，有三个状态而不是传统上的2进制数字能被逻辑电路认识，即逻辑“1”，逻辑“0”及“忽略”。在匹配时，忽略既能与逻辑“1”又能与逻辑“0”匹配。存储阵列中许多连续的存储位置上都包含三态。“掩蔽”是存储在存储器中的另一种形式的数据，它们将被送入阵元处理单元中的参考寄存器中。

由于存储阵列可以包含命令，该将使一个阵元可以做与另一个阵元不同的操作。操作期间大多数阵元（不一定全部）中的各个阵元进行的“在片”控制能允许进行对S I M D操作来说很独特的执行方式。能提供一个简单的控制功能，即是取消任何其状态输出满足一定条件的阵元的运算。这样，一个非零条件可以意味着一个忽略（doze）亦即是取消操作，使阵元关闭但进入等待状态的一种条件。提供的另一个命令是根据阵元的状态，或者根据在滑动操作之前提供给总线的命令禁止或者使能对存储器的写入。

如果把16个每个带有32KB存储器的强大的阵元做一个阵元芯片上，那么，只要64块芯片即可提供1024个处理器和32768KB的存储器。阵元阵列包括一套相关存储器。本发明除了矢量处理之外，还适用于数值计算很强的图象分析。目前，这个强



有力的阵元处理阵列可以封装在2块小电路板上。可以设想：成千个阵元可以被适当地封装在更轻便的低能耗包装中，这样可以以最短的延迟或者在视频帧时间内进行图象处理。举个例子来说，可以在飞机飞行过程中进行，同时又不必考虑负荷等问题。

阵元阵列的功能还导致使用装在闭合仪器舱中的大型相关存贮器系统的可能性，这样，在系统设计者对新系统的使用习惯之后，在各种场合下使用处理功能。

图3表示出了能被称作全相联存贮器是什么样的存贮器。当要求进行相联时，一个比较值被送到所有的存贮器位置上，所有的存贮器位置同时用它们的匹配线进行答复。相联存贮器因此而得名。在本系统中，为了执行查找要使用存贮器的并行阵元和具有字节转移电路的处理单元；为了在存贮器的N个字中找出一个字K，就要有数据和查找掩模输入。所有匹配的阵元都被状态线电平升高，然后执行一个分离操作读出或者选择第一个匹配的K。这种运算通常称作集合相联，可以对连续的字沿着阵元存贮器重复进行。同样，写是通过一个广播操作来实现的，其中被升高了的选择指示了参与情况，广播数据被拷贝至所有选中的阵元中。

另一个实施例虽然不是最佳实施例，但它减少了每个阵元的DRAM存贮器，从而允许加入具有图3所示类型的完全相联存贮器。如果加入了（比方说）512字长的全相联存贮器，这样，每个阵元都可以包括一个查找索引集合。在一次操作中， 512×1024 个阵元将每次完成512K次比较，或者说每次操作用1微秒的速度完成的话每秒可完成512G次比较。带有伸展性的这个概念可以延伸至多兆—比较的范围。本实施例允许进行涉及大量信息查找的相

联任务，其能力远远超过目前的运算能力。

使用图2中所示的存储器 and 字长宽度的相连处理单元进行的相联运算，除了应用于离散数字运算、人工智能、SIMD形态下总线进行并行编码之外，还能应用于目前只有芯片结构如我们在SIMD环境下描述的那类机器能完成的许多应用：

简单的可并行算术任务，包括能在专门的存储器机器上执行的短阵相乘和其它任务；

图象匹配，图象处理；它们可以在范纽曼的机器上执行，但对于那些适合于极度并行的场合（加三维图形的图案匹配）速度将大大提高。

根据数据的查询功能；

人工智能领域中的图形匹配；

在桥路上的网络控制；可以迅速识别从网络的桥路的另一侧送向用户的信息，

门级模拟；

以及检查VLSI中接地规则的违反情况。

当应用程序员借助于新系统体系的功能而开动脑筋时，他行就会想到利用存储器列及相联处理元件的各种处理任务。

通过给每个阵元100的一个门或者逻辑元件使用一个阵列，就可以加强保持描述一个数字系统的过程。在这样的系统中通过赋予每个门描述一个门电路作为输入的信号，并且给它产生的信号命名，处理就开始了。要求每次信号发生变化时，它的名字通过总线103被广播至所有的阵元，并期望的输入信号的名字进行比较。如果发现匹配，则把信号的新值记录在阵元中的数据流寄存器位中。当所有的信



号变化都被记录之后，使所有的阵元并行读出一个控制字，该控制字告诉它们的数据流怎样当前的输入集来计算输出。使这些计算并行执行，并使其结果与局部门电路中的原值相比较。把输出改变了的阵元的所有的门记录在数据流状态位中。使一个外部控制器对所有的阵元进行询问，并要求下一个改变了的门。然后，象原来说的，把适当的信号名和信号值从该阵元广播给所有的其它阵元；然后重复这个过程，直到不再有信号变化或者过程停止为止。

另一个过程为目录名查找。名字被存贮在阵元存贮器 102 中，使得所有名字的第一个字母都可以和广播数据地址总线 103 上的希望的被广播名字的第一个字母相连。通过我们提供的控制特性，把所有未匹配阵元关掉。接下来再比较第二个字母，为以后的几个字母（字符）都重复进行这样的比较和发断步骤，直至不再有工作着的阵元保留，或者已到达字尾。这时，剩下的阵元单元即为被查寻的单元，所希望的数据索引值由一个定序读出。

图 4 中示出了一种基本阵元结构，其中包括多个并行处理器，存贮器和阵元单元，它们被安置在一个单独的硅片中的一行上作为一个可以形成一个 SIMD 子系统的并行阵列的一部分，用来表示该系统的控制结构。图中还示出了控制处理器和管理微处理器。在图 4 中，与图中标为阵元阵列的部分位于同一芯片上的还有存贮器和并行处理单元逻辑。每个存贮器为 n 位宽，最好为我们说过的一个字符宽即 8（9）位，但是概念上也有一个字宽的多倍字长宽度存贮器。这样，并行阵元处理器的存贮器部分最好为 8（9）位宽，或者（6 位、32 位宽。鉴于目前的 CMOS 加工技术，我们倾向于每个阵元处理

单元使用 8 位或称一个字长宽的相联存贮器（9 位宽的字节带有自检功能）。这些功能器与相应的一个处理单元直接地——对应地相联，上述的处理单元包括一个 A L U，参考寄存器（A 和 Q 用于掩蔽），和一个门锁电路 1 0 4（图 4 中为 S P）以及状态寄存器 1 0 7 和数据流寄存器 A 1 0 5 和 Q 1 0 6（图 4 中的 D F），它们在图 2 中的阵元示意图中已详细地作了表示。D R A M 与每个阵元处理器的逻辑不会有争夺任何争夺互连网络的问题，其原因是在多位宽的 D R A M 存贮器与它自己在芯片上的处理单元之间有着直接的、一一对应的联系。

从图 4 中可以看出，滑动 B 寄存器门锁电路（S R）1 0 4 在逻辑上位于存贮器与相应的处理单元的 A L U 的逻辑电路之间，这个门锁电路实际已成为每个处理单元沿着阵元阵列的一个耦合端口。每个阵元芯片包括许多个排列成一行（图中画线直线点线）用于与阵元控制装置进行通讯的并行阵元处理单元。各个存贮器都有一根矢量地址点线，一个数据矢量地址寄存器控制着送至每个存贮器的数据。

图 4 中还示出了主机或者微处理器电路板 M P 与子系统控制器的互连情况。在我们的实施例中，M P 为装有 P S / 2 系统的 3 8 6 微处理器。全局指令通过上述子系统控制器到达一个封装程序处理器（C R P），该处理器（C R P）再为指令定序器 4 0 2 和执行控制电路 4 0 3 提供指令，指令定序器所调用的具体微码由执行控制电路（4 0 3）加予执行。这个指令定序器在功能上与控制相类似。在封装程序处理器 C R P 内部我们还提供了局部寄存器 4 0 5，该寄存器与局部寄存器 A L U（图中未示出）一起提供广播给阵元阵列 4 0 6 中的所有阵元的所有寻址信息的基地址。这样，所有的阵元

都在一个A L U中执行地址计算而不使用阵元资源，亦即不使用阵元的执行周期。这个重要的补充给阵元单元增加了控制上的灵活性，从而能执行一些特殊任务所需的忽略（doze），禁止和其它控制功能，并允许阵元与任何广播指令或者数据函数分开。

装入了微码4 0 7的指令定序器4 0 2对阵元阵列进行广播，从而在主程序微处理器M P和封装程序处理器C R P运行时间库4 0 8中的封装程序确定的S I M D指令序列下进行处理，这样，可使S I M D能处理包含在阵元序列中的数据。

通过子系统接口提供给微处理器M P的指令为一些高级处理命令，如通过微处理器M P的子系统控制器传送给微处理器的Start Process、Write Obser、以及Read Result。这种微处理器可以认为是图4，5，6和7中所示的主系统或者子系统结构中的控制处理器。应该理解的是这个单元可能是一个具有外围输入装置（如键盘）和显示单元的独立单元。在这种独立体系中，系统M P可以是沿图7中所示的线插有定序器卡（用于为封装程序处理器设置配置）和处理器阵列卡的市售P S / 2。程序库4 1 1包括用来时运算进行整体控制的一些程序序列，如C A L L（），Kalman，Conrolve，及Nav.Update.Update。由用户程序选择这些程序，这样，整体控制可以在一台外部主机的控制下或者在装入M P中的用户程序4 1 2的控制下进行。在M P存贮器中设有数据缓冲器4 3，用于把数据送入和送出并行阵元处理器系统。指令定序器4 0 2被配置成既能执行来自M P的控制串，又能执行驻留在封装程序运行时间库存贮器4 0 8中的封装程序。有些这种程序中包括来自封装程序运行时间库4 0 8中提供的封装程序的C A L L（，），Load

Block, Sin, Cos, Find, Min, Range Comp. 及 Matrix Multiply 等等。

C R P 中还有用于对低级函数如 Load, read, Add, Multiply, 及 Match 等进行执行控制的微码 4 0 7。

我们建议为每个处理单元提供一种外部 FOR / NEXT 控制。我们还提供一种决定性的 点字节的正常实施。

使用我们提供的在系统中建立宏的决定性方法可以允许阵元分组及 GROUP 控制。另外, 还提供了一种局部忽略功能来容纳各个阵元处理的变化。

用户如果要求处理器阵列执行时, 原始命令、地址及散播数据必须要提供给阵元处理器阵列。

系统的每个部分使用的具体函数由需解决的任务确定, 并在对用户程序进行编译时设定。

通过一个相当常见的问题可以表现出系统的灵活性。以短阵相乘的问题为例

这可以被描述成下面的问题:

$$\begin{array}{ccc}
 & M & C \\
 x_1 & x_{R+1} \dots & y_1 \ y_{M+1} \dots \\
 \cdot & & \cdot \cdot R \cdot \cdot \cdot X \ M \\
 \cdot & & \cdot \cdot \cdot \\
 x_R & x_{2R} \dots x_{RxM} & \cdot \cdot \\
 & & y_M \ y_{2M} \dots y_{MxC}
 \end{array}$$

$$\begin{array}{ccc}
 & C & \\
 z_1 & z_{R+1} \dots & \\
 \cdot & & \\
 R & \cdot \cdot & \\
 \cdot & & \\
 z_R & z_{2R} \dots z_{R+C} &
 \end{array}$$

这个问题可由下面的语句来解决。与这些语句紧邻的是传递次数及句数传递的时钟周期数，这里给出的仅为参考性实例：

```

                                cycles/
                                passes   pass 01  Call Matrix Mult Fx
1      c
      (R,M,C,Xaddr, Yaddr, Zaddr) 02  xSUB = ySUB = zSUB = 1
1      3 03  DO I = 1 to C          1      3 04  DO J = 1 to R
C      3 05  z = 0                  CxR    5/6* 06  DO K = 1 to
M      CxR    3 07  ***Assign to associative parallel processor***
08      Zz = Xx x Yy + Zz          CxRxM  204/345* 09
***Return result*** 10      xSUB = xSUB + R          CxRxM    2 11
ySUB = ySUB + 1          CxRxM    2 12  NEXT K
CxRxM    3 13  xSUB = xSUB - MxR + 1          CxR    2 14
ySUB = ySUB - M          CxR    2 15  zSUB = zSUB + 1
CxR    2 16  NEXT J          CxR    3 17  xSUB = 1
C      2 18  NEXT z          C      3 19  END Call
1      1

```

注：#定点（4字节）/浮点（1+4字节）

参阅上面的实例可以看出，上面的 0 8 语句指出的任务需要 9 8 % 的周期时间。因此，它根据赋予并行阵元处理器的 S I M D 结构。其它的处理占 2 % 的周期时间，因而被保留在微处理器内的体系中。

因此，对这个矩阵乘法的实例作一下回顾可以看出：具体执行过程将被赋给 M P、C R P、L R 或者阵元阵列（在编译时，每个语句都将产生在具体的系统位置上执行的运算）。

在上述这个矩阵相乘的实例中，0 1 语句将赋给主处理器 M P，

02、05、10、11、13、14、15和17语句将赋给局数寄存器LR，03、04、06、12、16、18和19语句将在封装程序处理器CRP内部执行，其它耗时的矩阵处理将在单指令下赋给阵元阵列执行，08语句也赋给阵元阵列执行。

图5。中示出了一种装有大量并行阵元处理器的多并行阵元处理器系统510。在最佳实施例中，我们建立了一配置有1024个并行处理器。并且1024个处理器具有2至4块SEM电路板的SIMD系统，该系统可用于多目标跟踪，传感器及数据汇合，信号处理、人工智能、卫星图象处理、图形/目标识别，里德·沙勒蒙(Reed Solomon)编码/解码运算等。每个电路板512可以插入带有楔形锁定滑槽514插装式系统舱513中。电路板上装有插入拨出杆516，这样，当盖板517盖紧时，它将有效地在滑架上固定住一个具有32至64M位的存贮器，其工作能力大约为每秒二百万次运算。整个系统相当紧凑，由许多阵元构成的阵列被插入到侧板518中，该电路板上还有允许多块电路板互连的逻辑电路。具有32兆存贮器的处理器做在4块SEM E电路板上，重量大约只有30磅(约12.62公斤)。电源由图中所示的电源519提供。经空气冷却的处理器电力消耗大约为280瓦。每个SIMD系统都有二个I/O端口250，用于与相当的主计算机或其它部分进行信道适配通道。图中所示出的多并行阵元处理中每一个均由4个逻辑页面构成，使用标准模数的设备封装和总线结构(如PI，TM和IEEE 488总线与外部存贮器相连，并能通过I/O端口与一台任务计算机的存贮器总线相连，也能被看作是任务计算机存贮空间的伸展。

图中示出的多并行阵元处理器其包括1024个并行处理单元，每个处理器都有32千字节的局部存贮器，与阵元并行处理器相连的通信为并行8位或者字符宽度19位）。

每个阵元中的处理器与其它的相邻处理器以及在页面之间通过一个互连网络交换数据，上述的互连网络最好为一个纵横制网络但也可以是滑动纵横制、混合式、底3N正方体或者座8N正方体网络。

系统的各个阵元处理器都被装在二组共4块电路卡上，PS/2微处理器在一块电路卡上，封装程序处理定序器则在包括着附图6和7中示出的系统的四块电路卡中的其余卡上。多个阵元100或者阵元卡512也可以被设置成与封装程序处理器CRP脱离，从而根据数据状态在与定序器卡CRP703的执行控制部分相连的闩锁电路104和局部寄存器405的控制之下进入或退出运算。这样，阵元处理器可以独立地执行与落点运算有关的定位与规格运算。

正象下面要讲述的那样，众多的处理器由一个共同的定序器加以并行控制。定序器卡703中包含阵元处理器的控制器CRP，可能使阵元执行一串以和传统的位串行处理相似的字节串行方式编码的旨在SIMD处理器阵列上执行的指令。控制器共有三层。与现代处理器一样，阵元的宏—控制被进行微编码后并行传送到所有的阵元中。宏控制和阵元与相同的时间系统CLK同步，这样，由定序器控制的函数可以在相同的时钟时间上被执行。给宏控制定序器输送命令是封装程序处理器的功能。定序器卡703是一个经过硬连成的控制器，它在大部分功能中执行循环控制命令，并且循环地开始新的宏控制序列。该控制器通过它的封装程序库408及其循环功能使阵元得到很好的输入，且不受命令的限制。封装命令程序处理器CRP控制器包

含一个很大的宏集合，它们将被主系统调用，在子系统中起到主管理阵元控制器的作用。这是阵元阵列的最高控制系统，它是一个管理着阵元阵列的活动386微处理器。在某一给定时间，阵列中的所存单元都执行着相同的指令，虽然处理器子集可能对控制流的反应不一。

具体的反应有多种变化，因此，根据每个阵元的字节控制功能（忽略、禁止等）规定了一定的局部自治。在编程时可以利用这种局部自治，在程序编译过程中可以在系统的控制之下进行这种局部自治。

此外，正象上面所讲的那样，还有局部存贮器寻址自主性。SIMD控制器定序器提供一个共同的地址，让所有的阵元来使用。每个阵元可以局部地增大地址，从而可以提高它进行与数据有关的存贮器访问的能力。

此外，一个阵元既可以参加也可以不参加阵列活动，视其局部位置而定。

有了这一特性，现在可以在SIMD处理中引入分组这一概念了。通过使每个阵元把它自身赋予多个组中的一个或多个，处理可以根据这些分组来进行，其中配置上的变化基本上没有。在一个实施例中，在某一时间只有一组或者一个组的组合是有效的，其中每一个都执行着相同的SIMD指令串。有些操作只需要阵元的一个子组或者一组阵元工作。编程时可以利用这一能力。局部参与自主性就这样起作用。很显然，阵元越多，运算就越好。

增加参与的阵元的数量的一个方法是使每个阵元执行它自己的指令流。实际上，这是SIMD中的MIMD。现在，就可以相同的SIMD机器基本配置成一台MIMD机器或者一台另一个不同配置

的机器，原因是使阵元根据它自身的指令序列来工作了。

由于每个阵元都可以有其自己的序列，因此在阵元级对一个非常简单的指令集进行解码是可能实现的，这将允许更多的局部运算发生。能为本功能找到原始利用场合的领域是作出复杂决定，而简单的定点运算则可能是程序员感兴趣的另一个领域。

一个简单的这样的程序能把一个阵元程序中的几块（比方说不超过2K）装入阵元存储器102，这可以在SIMD控制器卡103从一个具体的XYZ地址通过执行控制装置开始局部执行时执行。在控制器对时钟信号计数时，或者通过监视图4中所示的状态汇集（SF）寄存器测试任务完成信号时，上述过程还将结束。

状态汇集（图4中的SF）利用了每个阵元的开锁电路104。每个阵元都有一个加载后可反映该阵元状态条件的开锁104。SIMD控制器通过监测阵列状态线测试这些开锁中的集合值（每个开锁器电路有一个值）。这个阵列状态线是来自各个阵元状态开锁电路的值的逻辑组合。

在下面的例子中，我们假定想把一个大于250的值调整在 $500 > \times 7 = 250$ 的范围内。下面的程序将使用状态汇集来检测任务是否结束。

```
if VALUE<500 then TURN YOUR PICKET OFF  
STAT<-PICKET OFF CONDITIDN  
IF STAT FUNNEL = OFF then fiaished...  
VALUE < -VALUE -250  
Repeat
```

因此，多并行阵元处理的组态可以象一个SIMD处理器一样可

通过许多途径加以调整。在最佳实施例中，上述的这种 SIMD 机器被编程成以传统的方式执行一串指令并在 SIMD 控制器或定序器的总控制下，以和传统的处理器相似的串行格式编码成在 SIMD 处理器阵列中执行。在应用低别上，这是由矢量或者与矢量类似的指令来实现的，矢量既可以在处理器内也可以在处理器之间被进行处理。矢量指令可以和多条宏指令（典型情况下为 6 至 10 条这样的指令）相加。

最佳实施例中的系统正如图 6 中所示的并行处理器子系统的功能框图所示。子系统定序器通过被主机接口控制电路 413 控制的系统 I/O 端口起到与一个 SIMD 程序相似的作用，处理单元的功能由高函数宏加以控制。存储器寻址能对 8 位（即一个字节宽度）的数据流进行，楔 8 算术逻辑电路用于各种函数运算（逻辑、加、乘和除）。对于将点格式以及独立“睡眠（sleep）”的自主阵元运算，忽略状态及分离寻址，也作了规定。

图 7 中示出了子系统控制器的结构。每块处理器阵列电路卡 512（本子系统中示出了 4 块，也可能减少到 2 块 SEM E 卡）都与定序器 CRP 703 相连，CRP 703 又与子系统控制器 702 相连，子系统控制器 702 再通过芯片 705 接入一个主存储器系统或者接入配置中的另一个系统，从而接至相应的微通道总线 706。在本最佳实施例中，该子系统控制器是 IBM 公司生产的 PS/2（IBS 的商标）通用微处理器单元，采用英特尔（Intel）386 处理芯片，有 4 兆存储器。个人计算机微处理器 MP 702 通过子系统中的微通道型总线 705、706 与定序器电路卡相连。

很显然，根据上面的示教，对本发明还可以作许多变型和改进，因此，申请人相信后面的权利要求书将实际地而不是具体地描述本发明。

说明书附图

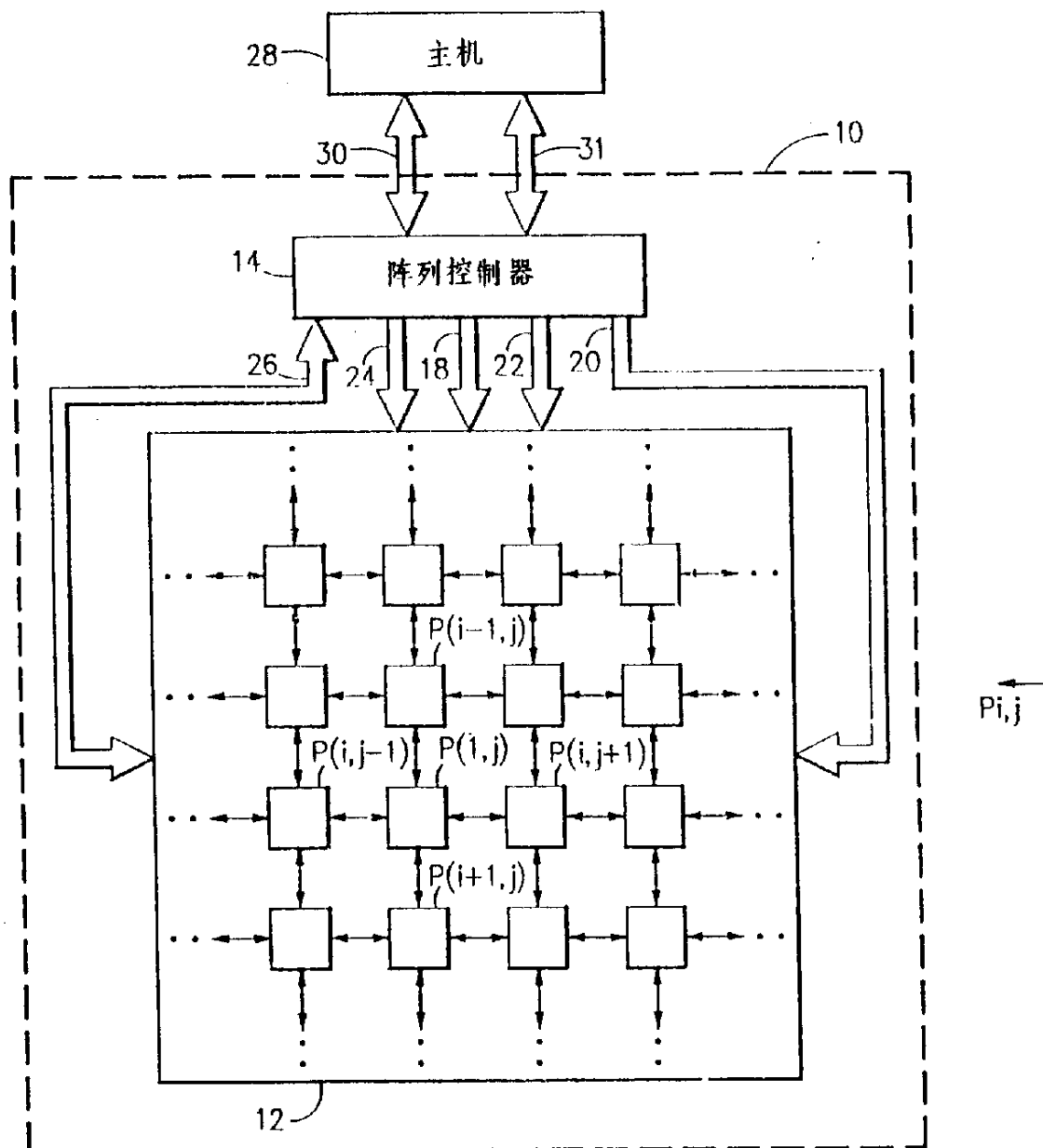


图.1

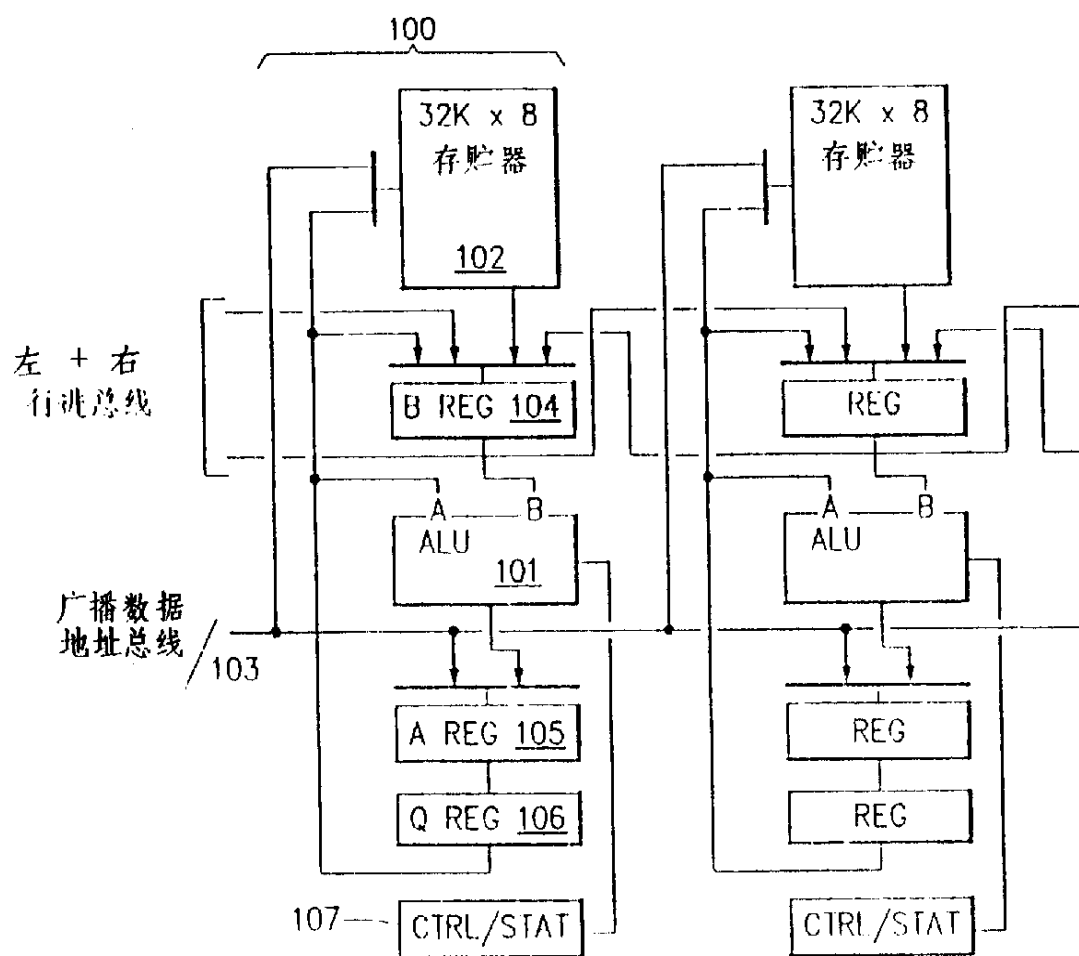


图.2

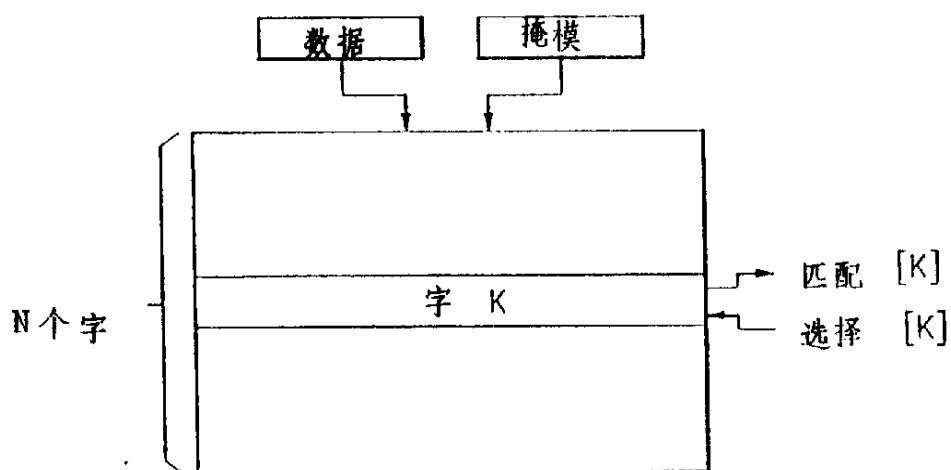


图.3

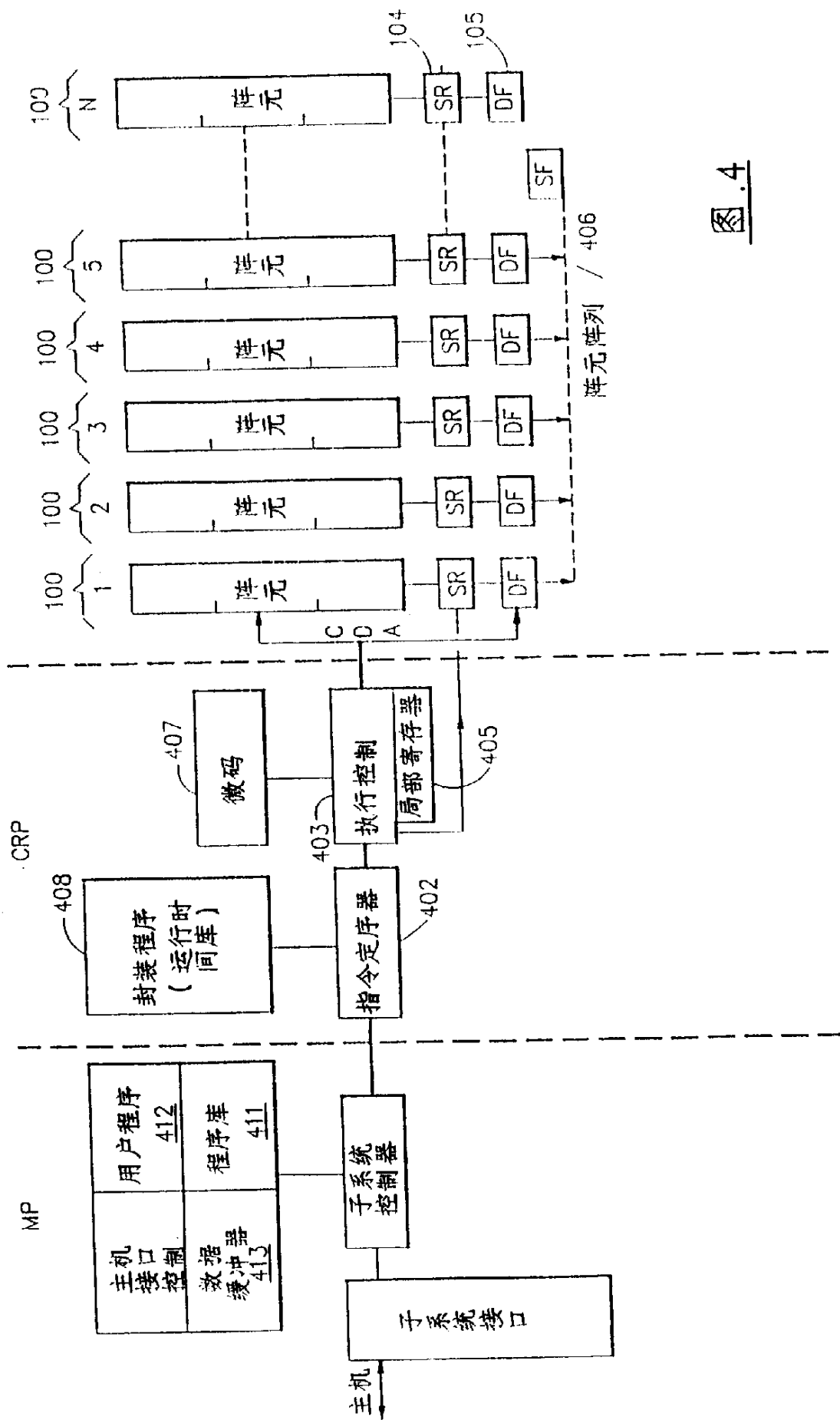


图.4

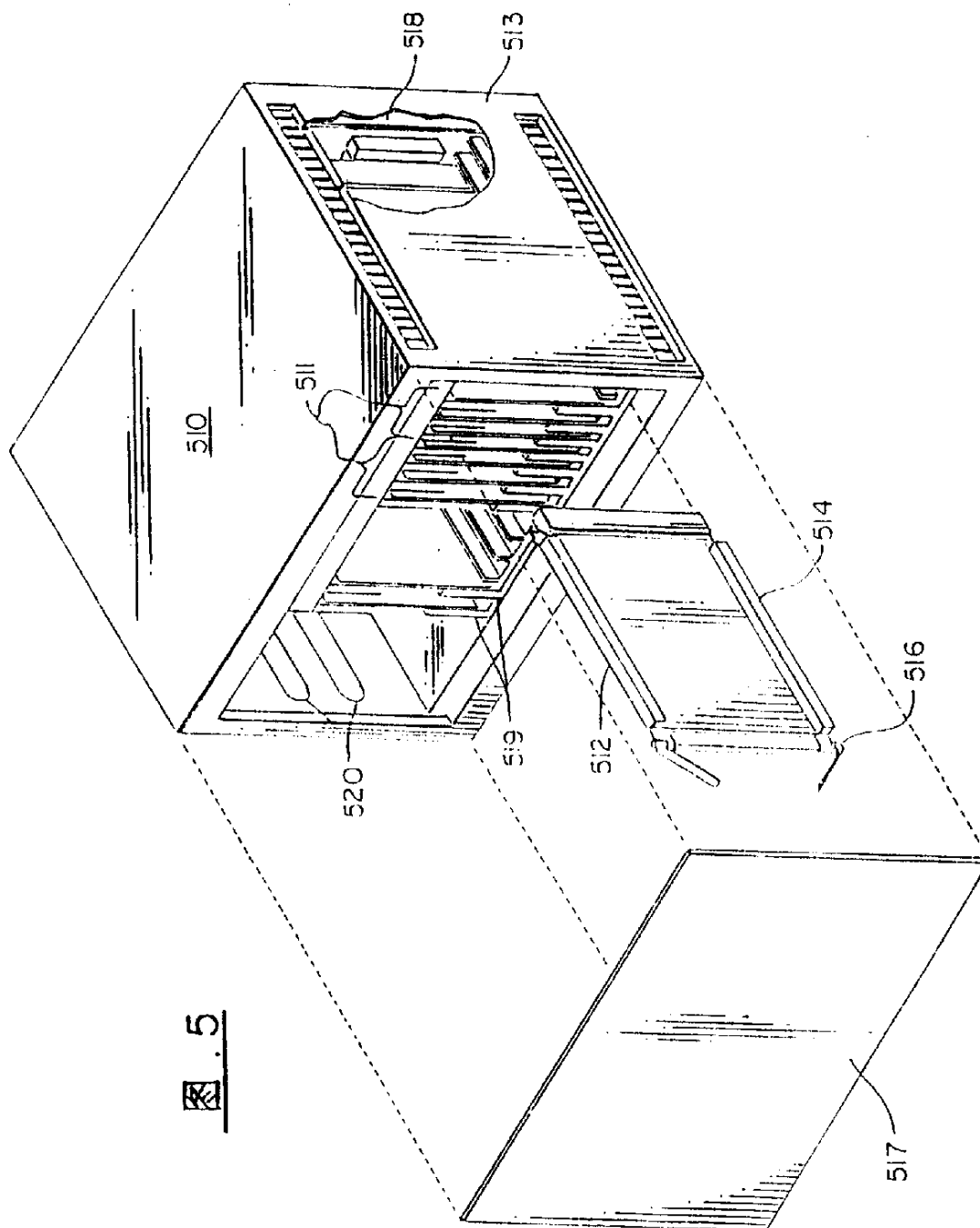


图. 5

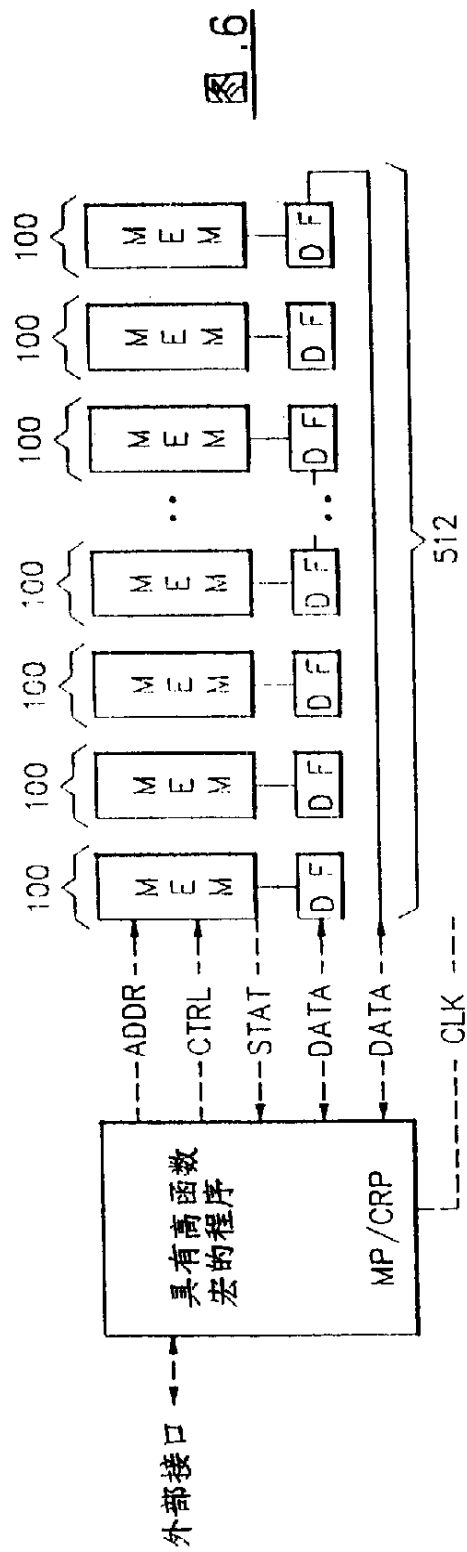


图.6

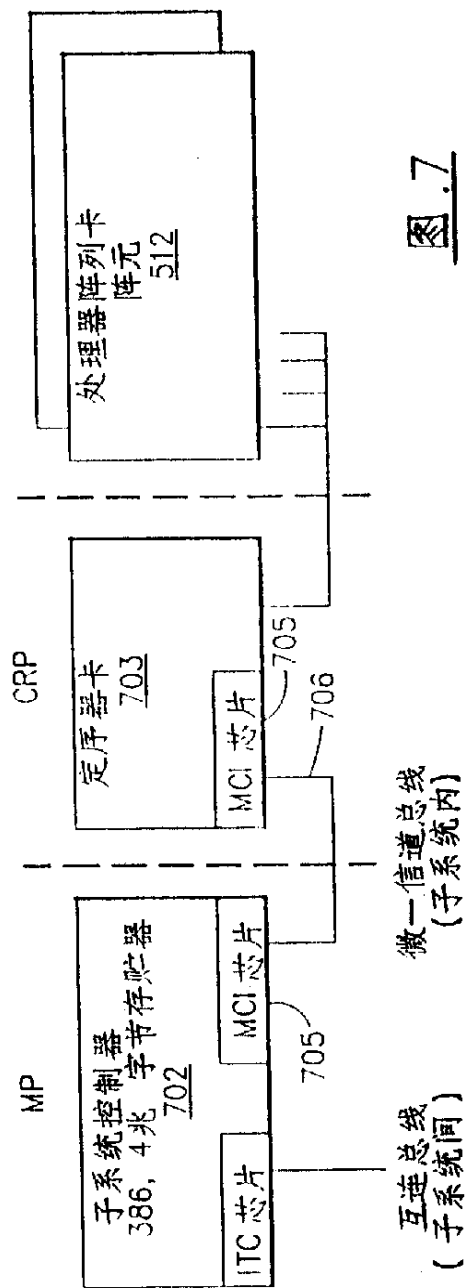


图.7