



[12] 发明专利申请公开说明书

[21]申请号 93104086.8

[51]Int.Cl⁵

G06F 12/00

[43]公开日 1994年10月12日

[22]申请日 93.4.8

[30]优先权

[32]92.4.10 [33]US[31]07 / 866,934

[71]申请人 国际商业机器公司

地址 美国纽约

[72]发明人 瑞查德·E·玛屈科

斯泰勒·E·理斯特

[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

G11C 5/02

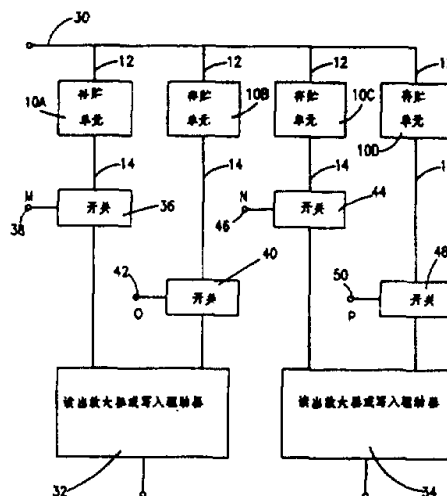
说明书页数:

附图页数:

[54]发明名称 用于电子计算机存贮器的位线开关阵列

[57]摘要

一个高速缓冲存贮器，具有多排存贮单元，每一排至少具有第一和第二块存贮单元。每个存贮单元存贮一个数据信号，并具有至少一个字线输入端和至少一个位线输入端/输出端。第一和第三存贮单元被包含在第一块中，而第二和第四存贮单元被包含在第二块中。第一和第二开关将第一和第二存贮单元的位线输入端/输出端分别连接到第一读出放大器/写入驱动器。第三和第四开关将第三和第四存贮单元的位线输入端/输出端分别连接到第二读出放大器/写入驱动器。



权 利 要 求 书

1. 一种电子计算机存贮器,其特征在于:

一个存贮单元阵列,每个存贮单元存贮一个数据信号,每个存贮单元具有至少一个字线输入端和至少一个位线输出端;

至少一个字线被连接到至少第一,第二,第三和第四存贮单元的字线输入端;

一个第一读出放大器,用于从存贮单元读取数据;

一个第二读出放大器,用于从存贮单元读取数据;

一个第一开关,用于将第一存贮单元的位线输出端交换地连接到第一读出放大器,所说的第一开关具有一个控制输入端;

一个第二开关,用于将第二存贮单元的位线输出端交换地连接到第一读出放大器,所说的第二开关具有一个控制输入端;

一个第三开关,用于将第三存贮单元的位线输出端交换地连接到第二读出放大器,所说的第三开关具有一个控制输入端;

一个第四开关,用于将第四存贮单元的位线输出端交换地连接到第二读出放大器,所说的第四开关具有一个控制输入端;

其中第一、第二,第三和第四开关的控制输入能够被独立地驱动。

2. 如权利要求1所述的电子计算机存贮器,其特征在于:地

址解码装置用于独立驱动第一和第三开关或第一和第四开关。

3. 如权利要求 1 所述的电子计算机存贮器,其特征在於每个存贮单元有两个位线输出端。

4. 一种电子计算机存贮器,其特征在於:

一个存贮单元阵列,每个存贮单元存贮一个数据信号,每个存贮单元具有至少一个字线输入端和至少一个位线输入端;

至少一个字线被连接到至少第一,第二,第三和第四存贮单元的字线输入端;

一个第一写入驱动器,用于将数据写入到存贮单元;

一个第二写入驱动器,用于将数据写入到存贮单元;

一个第一开关,用于将第一存贮单元的位线输入端交换地连接到第一写入驱动器,所说的第一开关具有一个控制输入端;

一个第二开关,用于将第二存贮单元的位线输入端交换地连接到第一写入驱动器,所说的第二开关具有一个控制输入端;

一个第三开关,用于将第三存贮单元的位线输入端交换地连接到第二写入驱动器,所说的第三开关具有一个控制输入端;和

一个第四开关,用于将第四存贮单元的位线输入端交换地连接到第二写入驱动器,所说的第四开关具有一个控制输入端;

其中第一、第二、第三和第四开关的控制输入端能够被独立地驱动。

5. 如权利要求 4 所述的电子计算机存贮器,其特征在於:地

址解码装置用于独立地驱动第一和第三开关或第一和第四开关。

6. 如权利要求 4 所述的电子计算机存贮器, 其特征在于: 每个存贮单元具有两个位线输入端。

7. 一种电子计算机存贮器, 其特征在于:

一个高速缓冲存贮器, 包括多排存贮单元, 每排存贮单元包括至少第一块和第二块存贮单元, 每个存贮单元存贮一个数据信号, 每个存贮单元具有至少一个字线输入端和至少一个位线输出端;

一个字线被连接到一排高速缓冲存贮器中的至少第一, 第二, 第三和第四存贮单元的字线输入端, 第一和第三存贮单元被包含在第一块中, 第二和第四存贮单元被包含在第二块中;

一个第一读出放大器, 用于从存贮单元中读取数据;

一个第二读出放大器, 用于从存贮单元中读取数据;

一个第一开关, 用于将第一存贮单元的位线输出端交换地连接到第一读出放大器, 所说的第一开关具有一个控制输入端;

一个第二开关, 用于将第二存贮单元的位线输出端交换地连接到第一读出放大器, 所说的第二开关具有一个控制输入端;

一个第三开关, 用于将第三存贮单元的位线输出端交换地连接到第二读出放大器, 所说的第三开关具有一个输入控制端; 和

一个第四开关, 用于将第四存贮单元的位线输出端交换地连接到第二读出放大器, 所说的第四开关具有一个输入控制端;

其中第一, 第二, 第三和第四开关的控制输入端能够独立地被

驱动。

8. 如权利要求 7 的电子计算机存贮器,其特征在於:选择装置用于独立地驱动第一和第三开关,以将仅仅在第一块中的存贮单元连接到读出放大器,或者独立地驱动第一和第四开关,以将在第一和第二两个块中的存贮单元连接到读出放大器。

9. 如权利要求 8 所述的电子计算机存贮器,其特征在於:选择装置另外独立地驱动第二和第四开关,以将仅在第二块中的存贮单元连接到读出放大器,或者独立地驱动第二和第三开关,以将在第一和第二两个块中的存贮单元连接到读出放大器。

10. 如权利要求 8 的电子计算机存贮器,其特征在於:

每一排存单元包括 M 块存贮单元;

每块存贮单元包括 N 个存贮单元;

计算机存贮器包括 N 个用于从存贮单元读取数据的读出放大器。

计算机存贮器包括 $M \times N$ 个开关,每个开关将每排仅仅一个存贮单元的位线输出端交换地连接到一个读出放大器,每个开关具有一个控制输入端;

该开关的控制输入端能够被独立地驱动;和

选择装置独立地驱动一个第一组开关,以将一个块中的 N 个存贮单元连接到读出放大器,或者独立地驱动一个第二组开关,以将 M 块的每一块中 N/M 个存贮单元连接到读出放大器。

说 明 书

用于电子计算机存贮器的位线开关阵列

本发明涉及电子计算机存贮器。电子计算机存贮器存贮一个中央处理单元使用的数据和指令。为有优良性能,一个电子计算机存贮器应具有大容量,高速度,低成本的特性。

为了获得优良性能,在一个多存贮级的存贮器分级体系中构成电子计算机存贮器是已知的。例如,与中央处理单元直接联系的第一存贮级(L1)可以包括一个小容量的、每个存贮单元为高成本的高速电子计算机存储器。与第一存贮级联系而不直接与中央处理单元联系的第二存贮级(L2)可以包括一个较大容量的、但每个存贮单元为较低成本的较低速的电子计算机存储器。如果需要,与第二存贮级联系的、可选择性地与第一存贮级联系但不直接与中央处理单元联系的第三存贮级可以包括一个更大容量的、但每个存贮单元为更低成本的更慢的电子计算机存储器。如果需要可以提供另外的存贮级。

第一存贮级一般是一个成组相连的高速缓冲存贮器。为了获得高速度,该高速缓冲存贮器包括有大量读出放大器和写入驱动器,它们可以扩大到跨越两列或更多列存贮单元(即,可以扩大到跨越两条

或更多条位/读出线)。在一个成组相连的高速存贮器中,利用中央处理单元的存取包括对其中只有一个含有所需要的数据或指令的两个或更多个高速缓冲存贮器数据块部分的一组存贮单元进行存取。从第一存贮级向更高存贮级的存取包括对含有所需要的数据或指令的整个高速缓冲存贮器数据块进行存取。

对于一个用于快速中央处理单元存取的成组相连高速缓冲存贮器的设计可能降低更高存贮级的存取速度,或者是可能要求附加的而且是更昂贵的电路以避免降低存取更高级存贮级的速度。

本发明的目的是提供一种电子计算机存贮器,它能够以高速度实现对中央处理单元的存取或对更高级存贮级的存取。

本发明的另一个目的是提供一种电子计算机存贮器,它能够以高速度实现对中央处理单元的存取或对更高级的存贮级的存取,而且能够利用一种简单的、低成本电路设计实现。

根据本发明,一个电子计算机存贮器包括一个存贮单元阵列。每个存贮单元存贮一个数据信号,而且每个存贮单元具有至少一个字线输入端和至少一个位线输出端。至少一条字线连接到至少第一,第二,第三和第四存贮单元的字线输入端。第一和第二读出放大器被用于从存贮单元中读取数据。

本发明中也提供有第一,第二,第三和第四开关,每一个都有一个控制输入端。第一开关可交换地将第一存贮单元的位线输出连接到第一读出放大器。第二开关可交换地将第二存贮单元的位线输出

连接到第一读出放大器。第三开关可交换地将第三存贮单元的位线输出连接到第二读出放大器。第四开关可交换地将第四存贮单元的位线输出连接到第二读出放大器。第一,第二,第三和第四开关的控制输入端能够独立地被进行驱动。

本发明的另一个方面中,电子计算机存贮器包括一个存贮单元阵列,每个存贮单元至少具有一个位线输入端。第一和第二写入驱动器用于将数据写入存贮单元。第一开关可交换地将第一存贮单元的位线输入连接到第一写入驱动器。第二开关可交换地将第二存贮单元的位线输入连接到第一写入驱动器。第三开关可交换地将第三存贮单元的位线输入连接到第二写入驱动器,第四开关可交换地将第四存贮单元的位线输入连接到第二写入驱动器。第一,第二,第三和第四开关能够被独立地进行驱动。

该电子计算机存储器还可以包括地址解码装置,用于独立地驱动第一和第三开关或者第二和第四开关。

每个存贮单元可以具有(例如)两个位线输入端和/或两个位线输出端。

本发明的另一个方案是,电子计算机存贮器包括含有其中包括多排存贮单元的一个高速缓冲存贮器。每排存贮单元至少包括存贮单元的第一和第二数据块。每个存贮单元存贮一个数据信号,而且每个数据单元具有至少一个字线输入和至少一个位线输出。一条字线被连接到高速缓冲存贮器一排中的至少第一,第二,第三和第四存贮

单元中的字线输入端。第一和第三存贮单元包含在第一数据块中,而第二和第四存贮单元包含在第二数据块中。第一和第二读出放大器被用于从存贮单元读取数据。

本发明中提供有每个都有一个控制输入端的第一,第二,第三和第四开关。第一开关将第一存贮单元的位线输出可交换地连接到第一读出放大器。第二开关将第二存贮单元的位线输出可交换地连接到第一读出放大器。第三开关将第三存贮单元的位线输出可交换地连接到第二读出放大器。第四开关将第四存贮单元的位线输出可交换地连接到第二读出放大器。该第一,第二,第三和第四开关能够被独立地驱动。

该电子计算机存贮器还可以包括选择装置,用于独立地驱动第一和第三开关,以将仅仅在第一高速缓冲存贮器数据块中的存贮单元连接到读出放大器,或者独立地驱动第一和第四开关,以将在第一和第二高速缓冲存贮器数据块中的存贮单元都连接到读出放大器。

该选择装置也可以独立地驱动第二和第四开关,以将仅仅在第二数据块中的存贮单元连接到读出放大器,或者独立地驱动第二和第三开关,以将在第一和第二数据块中的存贮单元都连接到读出放大器。

本发明的另一种方案,存贮单元的每一排包括 M 块存贮单元(典型地,当存在一个高速缓冲存贮器存取错误时,该高速缓冲存贮器“数据块”或高速缓冲存贮器“线”由一个更高级存贮级代替该高速

缓冲的存贮器的单元)。每块存贮单元包括 N 个存贮单元,计算机存贮器包括用于从存取单元中读取数据的 N 个读出放大器。计算机存贮器也包括 $M \times N$ 个开关。每个开关将每一排的仅仅一个存贮单元的位线输出可交换地连接到一个读出放大器。每个开关有一个控制输入端。该开关的控制输入端能够被独立地驱动。选择装置独立地驱动第一组开关,以将在一个数据块中的 N 个存贮单元连接到读出放大器,或者独立地驱动第二组开关,以将在 M 个数据块的每一个数据块中的 N/M 个存贮单元连接到读出放大器。

根据本发明的电子计算机存贮器通过利用能够被独立地驱动的开关,将存贮单元连接到读出放大器(即连接到写入驱动器)的方法,利用一个简单的和低成本的设计,能够实现向中央处理单元进行高速存取以及向更高的存贮级进行高速存取。

图 1 是根据本发明的电子计算机存贮器的一个范例的方框图。

图 2 用示意图表示了存贮单元的一个范例。

图 3 用示意图表示一个开关的范例。

图 4 用示意图表示一个读出放大器的范例。

图 5 用示意图表示一个写入驱动器的范例。

图 6 用示意图表示一个地址解码器的范例。

图 7 是根据本发明的另一个电子计算机存贮器的一个范例的方框图。

图 8 用示意图表示一个拉丁方格变址存贮单元阵列(Latin

square mapped memmory cell array)一部分的范例。

图 9 用示意图表示一个拉丁方格变址存贮单元阵列一部分的更详细的范例。

图 10 用示意图表示图 9 的拉丁方格变址存贮单元阵列的缩简图。

图 11 用示意图表示用于一个电子计算机存贮器的 18 个位地址的一个范例。

图 12 用示意图表示用于一个电子计算机存贮器的一个位线开关阵列一部分的一个范例。

图 13 用示意图表示用于一个电子计算机存贮器的一个地址解码器一部分的一个范例。

图 14 用示意图表示用于一个电子计算机存贮器的一个地址解码器另一部分的一个范例。

图 15 用示意图表示用于一个电子计算机存贮器的一个最新选取地址解码器一部分的范例。

图 16 用示意图表示在一个拉丁方格变址存贮单元阵列中的数据排列的一个范例。

图 17 用示意图表示将由一个拉丁方格变址存贮器单元阵列读出或写入一个拉丁方格变址存贮器单元阵列的数据进行重新组合所需要的数据移位的范例。

图 18 用示意图表示用于实施图 17 中所示的重新组合的一个数

据移位器/校准器的范例。

图 19 是根据本发明的另一个电子计算机存贮器范例的方框图。

图 1 是根据本发明的一个电子计算机存贮器范例的一个方框图。该电子计算机存贮器包括一个存贮单元阵列 10A, 10B, 10C 和 10D。每个存贮单元 10 存贮一个数据信号。每个存贮单元至少有一个字线输入端和至少一个位线输出和/或输入端 14。

图 2 用示意图表示包含有 6 个场效应晶体管的一个存贮单元 10 的一个范例。场效应晶体管 16 可以是(例如)N 通道的, 而场效应晶体管 18 可以是 P 通道的。存贮单元以第一状态或第二状态的形式存贮一个数据信号, 第一状态即在节点 20 具有高电压而在节点 22 具有低电压, 第二状态即在节点 22 有高电压而在节点 20 具有低电压。当一个合适的信号加到字线输入端 29 时, 开关场效应晶体管 24 就分别将节点 20 和 22 接到位线输入端/输出端 26 和 28。

回到图 1, 至少一个字线 30 被连接到至少第一, 第二, 第三和第四存贮单元 10A, 10B, 10C 和 10D 的字线输入端 12。第一读出放大器或写入驱动器 32 用于从存贮单元读取数据或将数据写入存贮单元。第二读出放大器或写入驱动器 34 用于从存贮单元读取数据或将数据写入存贮单元。

第一开关 36 用于将第一存贮单元 10A 的位线输出/输入端 14 转换地连接到第一读出放大器或写入驱动器 32。第一开关 36 具有一个控制输入端 38。第二开关 40 用于将第二存贮单元 10B 的位线

输出/输入端 14 转换地连接到第一读出放大器或写入驱动器 32。第二开关 40 具有一个控制输入端 42。第三开关 44 用于将第三存储单元 10D 的位线输出/输入端 14 转换地连接到第二读出放大器或写入驱动器 34。第三开关 44 具有一个控制输入端 46。第四开关 48 用于将第四存储单元 10D 的位线输出 / 输入端 14 转换地连接到第二读出放大器或写入驱动器 34。第四开关具有一个控制输入端 50。如图 1 所示,第一,第二,第三和第四开关 36,40,44 和 48 都能够被独立地进行驱动。

图 3 用示意图表示出开关 36,40,44 或 48 的一个范例,在该范例中,开关包括一个单个的场效应晶体管 52,该场效应晶体管有一个构成控制输入端的栅极 53,也可以使用其它的合适的开关。

图 4 用示意图表示出读出放大器 32 或 34 的一个范例。在该范例中,读出放大器包括交叉耦合的场效应晶体管 54。将数据信号读入该读出放大器,而放大的数据信号在节点 56 从该读出放大器读出。

图 5 用示意图表示写入驱动器 32 或 34 的一个范例。在该范例中,写入驱动器包括 N 沟道场效应晶体管 58 和 P 沟道场效应晶体管 60。“实”和“补”输出端 62 和 64 分别提供相应表示在输入端 66 的数据和在输入端 66 的数据的补码。

图 6 用示意图表示一个地址解码器的范例。在该范例中,地址解码器包括输入端 68,70,72 和 74,或门 76,以及输出端 M,N,O 和

P。输出端 M, N, O 和 P 分别连接到图 1 的开关 36, 44, 40 和 48 的控制输入端 38, 46, 42 和 50。通过在地址解码器的输入端 68 提供一个信号, 就可启动第一和第三开关 36 和 44。通过在地址解码器的输入端 70 提供一个信号, 就可启动第一和第四开关 36 和 48。在地址解码器输入端 72 提供一个信号启动第二和第四开关 40 和 48, 而在地址解码器输入端 74 提供一个信号启动第二和第三开关 40 和 44。

当图 1 的存贮单元 10 形成一个高速缓冲存贮器的整排或部分排时, 第一和第三存贮单元 10A 和 10C 可能包含在该存贮单元的第一数据块中, 第二和第四存贮单元 10B 和 10D 可能包含在存贮单元的第二数据块中(如前所述, 当存在一个高速存贮器存取错误时, 典型地一个高速缓冲存贮器“数据块”由一个更高存贮级替代高速缓冲存贮器的该单元)。在这种情况下, 图 6 的地址解码器形成一个选择装置, 用于独立地启动第一和第三开关 36 和 44, 以便只将第一数据块中的存贮单元连接到读出放大器 32 和 34。或者独立地启动第一和第四开关 36 和 48, 以便将第一和第二数据块二者中的存贮单元连接到读出放大器 32 和 34。

图 7 是根据本发明的一个电子计算机存贮器优选实施例的一个范例的方框图。在该优选的存贮器分级体系中, 第一存贮级(L1)最接近于中央处理单元(CPU)并由中央处理单元(CPU)存取, 第一存贮级 L1 包括一个静态随机存取存贮器(SRAM)高速缓冲存贮器,

构成一个最新选择的四路成组相联阵列。第二存贮级(L2)包括一个动态随机存取存贮器(DRAM),该存贮器(DRAM)是一个后援存贮器,第一存贮级在高速缓冲存贮器错误时从其上重新加载。在该范例如中,第一存贮级或第二存贮级可以从一个形成第三存贮级(L3)的一个主存贮器重新加载。第一和第二存贮级最好集成在一块芯片上(虽然不是必须的)。随着所希望的容量而定,一个典型的存贮器由一片或更多的这种芯片构成。

所希望的这种存贮器分级体系描述在1992年1月24日申请的系列号为826,306的美国专利申请中,该申请名称为具有多宽度,高速度通信缓冲放大器的电子计算机存贮器(*Electronic Computer Memory System Having Multiple Width, High Speed Communication Buffer*)。”

为了更清楚地理解本发明,并且表示本发明的总的性能,将利用一组特定参数说明最佳实施例。

参看图7,第一存贮级由一个256,000(256k)字节静态随机存取存贮器单元阵列78构成,它是利用八个相同的芯片得到的。每个芯片包含有作为整个芯片一部分的一个256k静态随机存取存贮器阵列。该静态随机存取存贮器可以具有例如类似于在下述文章中所描述的特性,即名为“*A 2-ns Cycle, 3.8-ns Access 512-kb CMOS ECL SRAM with a Fully Pipelined Architecture*”,该文章作者是T. Chappel, B. Chappel, S. Schuster, J. Allan, S. Klepner, R.

Joshi 和 *R. Franch*(*IEEE* 固体电路年报,第 26 卷第 11 期,1991 年 11 月,第 1577—1585 页,*ISSCC91* 技术论文摘要,1991 年,2 月 13—15,美国旧金山市,第 50—51 页)。

利用每个 256 字节的高速缓冲存贮器数据块,将高速缓冲存贮器构成为一个四路成组相联的、最新选择阵列。每个高速缓冲存贮器数据块是该高速缓冲存贮器的一部分,在该高速缓冲存贮器中存贮有一组来自例如主存贮器的一个更高级存贮器数据的一组相关字节(在该范例中,256 个相关字节)。该设计是一个高速存入缓冲存贮器。一个存贮通过(*Storu—thru*)高速缓冲存贮器是一个可行的选择,但在该系统中不是典型,而且也不是最佳实施例。假设由中央处理单元存取的希望的逻辑字(*LW*)是一个由八位字节组成的双字(*DW*),该八位字节由来自八个芯片的每个芯片形成。

静态随机存取存贮器单元阵列 78 构成为每一排的 1024 个存贮单元的 256 排的一个阵列。每一排存贮单元被提供有一条字线 80,每一列存贮单元被提供有一条位/读出线(或者一条位/读出线“实”/“补”对)。在图 7 中,每条线 82 代表八个位/读出线(即,八个位/读出线“实”/“补”对)。为了获得高速度和高密度、非常大、集成优良的读出放大器,就需要跨越 n 个单元(位/读出线)。因此,适用的读出放大器的数量将是在每排存贮单元数量的四分之一至八分之一的范围内。我们将选择(例如)具有 1024 个存贮单元的每排 256 个读出放大器的设计。

为了获得一个四路成组相联的、最新选择高速缓冲存储器设计(该设计具有快速重新加载能力),在组相关性、数据块容量、逻辑字规格、以及阵列结构之间必须存在确定的关系。存储位的变换则必须以一种恰当的方式进行以达到该设计。对于某些确定的条件,一个非常简单的变换和阵列结构是可能的。这些条件是:如果我们希望在2周期中的 N_c 芯片上加载和卸载一个 B_{bl} 字节的整个数据块,则每一排的数目(在每个实字线上的位)必须满足:

$$b_r \geq \frac{8 S B_{bl}}{\alpha N_c} = \frac{S b_{bl}}{\alpha N_c} \quad [1]$$

其中 $b_{bl} = 8B_{bl}$,而读出放大器可用于阵列的顶部或底部,为了从一个相合级读出四个双字,或从重新加载通路读出一个数据组,则必须等于或大于等式1中的 b_r 的最小值。

在我们的最佳实施例中,取 $S=4, N_c=8$ 芯片, $B_{bl}=256$ 字节,和 $\alpha=1$ 周期,该简单的变换将需要 $b_r \geq$ 每排1024字节以及1024个读出放大器的最小值。我们的阵列具有每排1024位的最小数目,但只有256个读出放大器。因此,简单的变换是不可能的。所以,就需要一个不同的而且更复杂的称为“拉丁方格”的变址类型以达到具有成组相联的,最新选择设计的快速重新加载带宽。(参看,例如,Matick, R. E等人的“*Functional cache for improved system performance*”。*IBM Journal of Research and Development*,第33卷

第一期,1989年1月,第15—32页)。

以下的描述使图8所示的拉丁方格变换扩大应用。在高速缓冲存贮器数据块A,B,C和D上的符号0,1,2和3表示逻辑字边界,在该情况中,它是八字节的一个双字。

定义两种相同级,一种用于排,一种用于列。用于典型的成组相联的、最新选择高速缓冲存贮器设计的普通相同级的标记是排相同级RC,由图8中的排组成。它们被标为RC0,RC1,RC2和RC3,如图所示。每一排的相同级包含来自四个高速缓冲存贮器每一个数据块的一个双字。每个双字具有相同的双字标记数。因此,第一排包含第一个双字DW0,它来自在高速缓冲存贮器一个排上的四个高速缓冲存贮器数据块中的每一个。第二排包含第二个双字DW1,它来自在高速缓冲存贮器一个排上的四个高速缓冲存贮器数据块中的每一个。第三排包含第三个双字DW2,它来自在高速缓冲存贮器一个排上的四个高速缓冲存贮器数据块中的每一个。第四排包含第四个双字DW3,它来自在高速缓冲存贮器一个排上的四个高速缓冲存贮器数据块中的每一个。

由图8的每一列所表示的一种新型相合级,也只包含来自四个高速缓冲存贮器数据块中的每一个的一个双字。然而,在该情况中,每个双字都有一个不同的双字标记号。它被称为拉丁方格列相同级(LCC)。例如,在图8的变址中,每个拉丁方格列相同级包含逻辑字0,1,2和3,它们来自四个不同的高速缓冲存贮器数据块A,B,C

和 D 。

由于来自一个拉丁方格变址的一排中的每个高速缓冲存储器数据块的每个双字有相同的标记,通过对每一列提供一个存取端口,就能通过选择的列 I, II, III 和 IV 及一个单排存取具有来自四个高速缓冲存储器数据块的每一个的相同标记的双字。这就构成了一个用于最新选择高速缓冲存储器设计的一个普通中央处理单元存取。

为了重新加载,所有端口都应进入一个高速缓冲存储器数据块。由于每一列只包含来自每个高速缓冲存储器数据块的一个双字,所以这对于拉丁方格变址是可能的。例如,高速缓冲存储器数据块 A 通过选择在 0 排上的列 I , 在 1 排上的列 II , 在 2 排上的列 III , 在 3 排上的列 IV 进行重新加载。以类似的方式,高速缓冲存储器数据块 B, C 或 D 通过排与列的适当的组合就能够进行选择。执行两种不同类型的寻址(一种用于普通存取,一种用于重新加载)的能力是由以下描述的特殊的寻址解码器完成的。

作为最佳实施例,利用一个交叉的拉丁方格型存同单元变址,产生图 9 所示的存贮单元阵列布局。每个存贮单元具有形式 $X_{LW, BIT}$ 的变换,此处 x 是数据块 A, B, C 或 D 中的一个, LW 是序号,从 D 至 32 的逻辑字标引(每个逻辑字是一个八字节的双字),而此处的 BIT 是双字中的位数 0 至 63。每个读出放大器 84 有一个形式为 $SA_{LCC, BIT}$ 的标记,此处 LCC 是相关的拉丁方格列相同级,而 BIT 是在相关拉丁方格列相同级中的双字内的位数 0 至 63。

读出放大器间距为四,意即每个读出放大器(SA)84 为四个位/读出线服务,并要求对四个位/读出线 82 的一个输出进行解码。这是通过位开关 86 以下述方式完成的。读出放大器精确地匹配四组相关性,并提供一种交叉拉丁方格变换的方法,以将用于数据调整的移位长度减至最小,如将看到的那样。除了位单元由每个连续的芯片上的一个进行增加外,所有的八个芯片具有一致的变换,如图所示。因此,八字节逻辑字由沿着八个芯片上的一个排的相邻位组成。

为了简化变换图并使它们与拉丁方格变换排序相关,将以下方方式表示该变换。注意沿着图 9 中的任意排(字线 80)的第一个四位被连接到相同的读出放大器,并与图 8 中的第一列对应,即 LCC-I。这样,这些四位即使属于一条单根的字线,也被表示为垂直地沿字线 88。另外要注意图 9 中的第一组八个读出放大器都属于相同的双字(逻辑字)并具有相同的双字标引数。因此,为了变换的目的,我们可以认为八个读出放大器的每一组都属于一个拉丁方格列相同级,如图 9 下部所示。

由于中央处理单元逻辑字 LW 总数为八字节,即每个芯片八位,则任何高速缓冲存储器数据组将由在八个芯片上的每一个上的 256 位组成。每个芯片上的一组四个高速缓冲存储器数据块需要有 $4 \times 256 = 1024$ 位。由于这精确地等于每个芯片上的每个字线的比特,则在八个对应的芯片上的每一排存储了整个相同级,即四个完整的数据块。由于任何给定的数据块只需要每个芯片上的 256 位,而且有

256 个读出放大器,因此,在一个周期中,可以将一个全数据块从一个重新加载缓冲器重新加载到高速缓冲存储器阵列中。如果读出放大器间距是在八个中的一个而不是四个中的一个,则只给出 128 个读出放大器而不是 256 个读出放大器,那么一个全数据块卸载或重新加载将需要两个周期(参看,例如 *Matick, R. E* 等人的上述的 1989 年 1 月的文章)。

为了能够将最新选择的八字节模拟字存取到中央处理单元,则拉丁方格标引必须被完成。如图 9 所示,在双字(八字节)边界上标引为一个最小量。利用这个构形,就能够在相同的芯片上,以下述的有效方式提供所有需要的移位,寻址以及最新选择解码。

图 9 的全部位和单个位变换可以依所示的方式进行简化,以对每块芯片给出图 10 所示的简缩的拉丁方格变址。鉴于八块芯片作为一个单元,则每条字线(排跨接八块芯片)包含来自四个数据块中每一个数据块的 32 个双字(逻辑字)。第一字线包含来自四个数据块 *A, B, C* 和 *D* 一个组的 32 个双字 (*DW0* 到 *DW31*)。第二字线包含来自四个数据块 *A, B, C* 和 *D* 的一个不同的组的 32 个双字 (*DW0* 到 *DW31*), 第三字线包含来自四个数据块的一个不同组的 32 个双字,等等。

图 10 所示的每个拉丁方格变换(跨接八块芯片)包含来自四个数据块中每一个数据块的四个双字,即一个完整的 16 个双字,如图所示。因此,每条字线具有八个这样的拉丁方格变换:在字线 0 上的

$M(0,0)$ 至 $M(0,7)$,在字线 1 上的 $M(1,0)$ 至 $M(1,7)$ 等等,其中符号表示法是 $M(\text{字线},\text{变换})$ 。我们将利用这个变换接着讨论,以表示出所需要的各种功能和一些可能的实现方法。

分配给每个拉丁方格地址变换的双字(逻辑字)如图 10 所示,而且它们如下:

$M(W,0)$:双字 0,1,2 和 3

$M(W,1)$:双字 4,5,6 和 7

$M(W,2)$:双字 8,9,10 和 11

$M(W,3)$:双字 12,13,14 和 15

$M(W,4)$:双字 16,17,18 和 19

$M(W,5)$:双字 20,21,22 和 23

$M(W,6)$:双字 24,25,26 和 27

$M(W,7)$:双字 28,29,30 和 31

这些分配将用于获得用于以下各种情况的适宜的解码器。

对于图 10 的变换,静态随机存取存储器的高速缓冲寄存器的十八位地址示于图 11 中。

字线位, W_0 至 W_7 ,选取 256 条字线中一条,每一条包含用于四个数据块中每一个数据块的双字。位 D_0 至 D_5 选取来自四个数据块中每一个数据块的 32 个双字之一。位 D_0 和 D_1 选取在每个变换地址中的四个双字标引中的一个, D_2 至 D_4 选取八个可能的拉丁方格变换中的一个。位 S_0 和 S_1 选取的每条字线上的四个数据块中的一

个,并由高速缓冲存储器控制器提供。它规定,例如,数据被读为存储返回,或被写为重新加载。位 B0 至 B2 选取在一个双字中的八字节中的一个字节。

如图 7 所示,只提供有一组 256 个读出放大器 84,它们被提供在存储单元阵列的下部。该读出放大器 84 被表示为每组八个读出放大器的 32 组。读出放大器组被标记为 SAG0 至 SAG31。

对于一个正常的读存取,解码器 90 和位开关 86 选取 32 个可能的双字,从四个数据块的每一个数据块取 8 个。正确的双字和数据块的剩余解码由最新选取的图形逻辑 92 完成,这将进行描述。最新选取的图形逻辑 92 通过驱动在每个芯片上的八个三态驱动器 98 选取双字中的一个,如图 12 所示。

对于一个要求高速缓冲存储器数据块中的一个数据块存储返回的错误,解码器 90 和位开关 86 从在完成解码之前现在必须被确定的数据块中选取所有的 32 个双字。通常知道该数据块来自错误产生的在前的周期。通过位于图 7 靠上部的移位器/校准器 96 对这些 32 个双字(每块芯片 256 位)进行选通(*gate up*),然后进入存储返回缓冲器 98。

要想执行选取的 32 个双字,从四个数据块的每一个数据块选 8 个,或者所有的 32 个双字选自一个数据块两种功能,解码器 90 必须具有一个至每个变换的每个位开关 86 的单独输入。每个变换有 16 个位开关 86,因此需要 16 个启动信号,如图 12 所示。所有的八个

交叉的拉丁方格地址变换都如其中所示的那样进行并联连接。图 12 中表示的每一个位开关 86 是一组并联连接的八位开关(参看图 9)。然而,由于只用一个单个的启动信号触发八个并联位开关的每一组,所以只需要 16 个启动信号。

图 13 用示意图表示用于产生触发位开关 86 的 16 个启动信号的解码器 90 的一部分。解码器 90 包括与门 100 和或门 102,如图所示。对于中央处理单元通路,解码器 90 选取 32 个双字总量,从数据块的每个数据块的双字取 8 个。对一个字线中的八个拉丁方格变换中的每一个,解码器 90 通过只选取四排相同级 $RC0$ 到 $RC3$ 中的一个来完成这项工作。解码器 90 并不要求数据块完全相同,因此这种解码和位开关选择就能够与阵列存取和变换平行进行。换言之,拉丁方格变换并不将任何附加延迟引入该通路的这一部分。

图 24 用示意图表示出也构成最新选取图形逻辑 92 的解码器 90 的另一部分。通过与门 100 和非门 104,地址位 $D0$ 和 $D1$ 产生一个适于四排相同级 $RC0, RC1, RC2$ 或 $RC3$ 之一的信号。

在由读出放大器 84 读出位之后,最新选取图形逻辑 92 必须从所希望的数据块中选取八个双字之一。但是,由于该数据是排列成行的拉丁方格,所以它必须被分解。最新选取的图形逻辑 92 分解该拉丁方格排列并利用图 10 中的八个可能的变址 M ,对四排相同级 $RC0$ 至 $RC3$ 中的 32 个指令的一个输出进行解码。该解码适合于四个数据块 A, B, C 和 D ,如图 15 所详细描绘的那样。最新选取图形逻

辑 92 包括与门 100 和或门 102。最新选取图形逻辑 92 接收来自图 14 所示逻辑的排相同级信号和变换号码信号,并从高速缓冲存贮器控制器接收一个最新数据块选取信号。从这些输入信号中,最新选取图形逻辑 92 产生启动信号 E_0 至 E_{31} 中的一个。生成的启动信号选取在每块芯片上的每组八个读出放大器的 32 组中的一组,如图 12 所示。这就向中央处理单元提供一个双字(跨接八块芯片)。但是,由于位总是在任意双字中排列成行,所以就不需要再排列。

如从图 15 所看到的,用于选取正确双字的这个解码的初始部分能够立即起动。由拉丁方格变换引入的附加延迟基本上就是在选择正确的双字以驱动中央处理单元之前的一个附加的与门的延时。

为了将数据从高速缓冲存贮器返存到第二存贮级,解码器 90 和位开关 86 从如图 12 和 13 所示的一个单个数据块 A, B, C 或 D 中选取 32 个双字。在八块芯片的每块芯块上的 256 个读出放大器恰好足以保持来自四个数据块的任一数据块的一个完整的数据块。

对于交叉的拉丁方格变换,在读出放大器中的地址将不必从左到右排列成行。进而,特别生成的排列将依赖于所选取的数据块进行变化。因此,在解码器 90 起动适当的位开关 86 以利用数据选取信号从一个确定的数据块中选取所有的 32 个双字之后,以及在 32 个双字从 256 个读出放大器中被读出之后,在读出放大器中的数据必须进行移位或被排列到正确的双字边界。

当解码器 90 和位开关 86 选取双字的整个数据块时,在读出放

大器中的数据将被排列到由图 10 中的交叉拉丁方格变换确定的边界上。对于由解码器 90 选取的数据块中的每一个,其排列将如图 16 所示。数据块 A 排列很好而无需移位。数据块 13 排列到八位的一个读出放大器组(SAG)的右边,数据块 C 排列到两个读出放大器组的右边,数据块 D 排列到三个读出放大器组的右边。为了获得合适的排列,则数据块 B, C 和 D 将必须分别向左移位一个,二个和三个组,如图 17 所示。

由于拉丁方格变换的这种交叉构形,在任何数据上的最大移位距离是三个读出放大器组,即一个 $3 \times 8 = 24$ 位单元的实际距离。如果我们不使用交叉变换,而是使用一个跨接在一排的全部 1024 位上的直接拉丁方格变换,则移位距离将是 $31 \times 8 = 248$ 个位。这将增加线长和延迟,而且将给出一个很复杂的布线图,这对于一个高速的设计都是有坏处的。交叉变换极大地简化和减少了布线通路。

在数据被寄存之后,在读出放大器输出端上就进行数据排列。移位器/校准器可以包括,例如如图 8=18 所示的一个简单的位开关网络。输入启动信号简单地就是来自高速缓冲存储器控制器的一般数据块选取信号。数据总是被移动规定的量以用于任何给定的数据块。如前所述,移位是在四个双字边界上进行的,而且存在八个这样的相同图形,其中的一个都用于图 10 中的 M 个变换中的每一个。

回到图 7,为了从中央处理单元(CPU)或者从第二存贮级 110 写入存贮单元阵列 78,假设在写入周期开始之前就完成了转换。为

了从中央处理单元写入存贮单元阵列 78, 将八个写入驱动器 108 连接到位/读出线的顶端。为了从第二存贮级 110 写入存贮单元阵列 78, 在高速缓冲存贮器重新加载缓冲器 112 中的 256 个写入驱动器利用移位器/校准器 96 和一组重新加载启动开关 114 连接到位/读出线 82 的顶端。芯片在底端上的输出通路的反转必须在顶端上完全再现。这就要求解码器 116 和位开关 118 与解码器 90 和位开关 86 完全一样, 最新选取图形逻辑 120 与最新选取图形逻辑 92 完全一样, 以及相关的选择开关 122 将写入驱动器 108 连接到位/读出线 82。

在第一存贮级的静态随机存取存贮器与第二和第三存贮器的动态随机存取存贮器之间的接口需要执行一定的功能。这些功能的精确方法和定位可能稍微变化。首先, 利用交拉丁方格变换, 使进入静态随机存取存贮器和从静态随机存取存贮器读出的数据必须进行移动/排列。第二, 从第一存贮级的静态随机存取存贮器至第二存贮级的动态随机存取存贮器的接口是每个周期 256 比特, 而第一和第三存贮级之间的接口是每周期 32 比特。然而, 静态随机存取存贮器的存贮返回和重新加载通路由图 7 所示的存贮返回缓冲器 98 和重新加载缓冲器 112 进行缓冲。

移位器/校准器 96 可位于静态随机存取存贮器阵列和存贮返回及重新加载缓冲器之间, 如图 7 所示, 而且以双向模式工作, 对输出数据和输入数据都进行移位。

移位器/校准器 96 的另一种位置可以在第二和第三存贮级 L2/L3 二者及存贮返回缓冲器 98 和高速缓冲存贮器重新加载缓冲器 112 二者之间,如图 19 所示。移位器/校准器仍然以双向模式工作。有时只有 32 比特通过移位器/校准器 96,而在其它时间则有 256 比特,这要根据操作而定。移位器/校准器只是一组无源开关,因此,不存在有源的问题。对于送至或来自第二存贮级 110、引出或者引入存贮返回缓冲器 98 或高速缓冲存贮器重新加载缓冲器 112 的数据,则多路转换器 124,126 或 128 将选择 256 位的通路。

对于送到或来自第三存贮级 L3 以及引出或引入存贮返回缓冲器 98 或高速缓冲存贮器重新加载缓冲器 112 的数据,多路转换器 124,126 或 128 选择 256 行的 32 行,其余行则处于高阻状态。对于此情况,整个数据块的传输要求 8 个周期,而且多路转换器 124,126 或 128 将必须控制增量地址以及 32 位连续组的选择。(参看,例如 Matick, R. E 等人的“*Architecture, design, and operating Characteristics of a 12-ns CMOS functional cache Chip*”, *IBM Journal of Research and development*, Vol. 33, No. 5, September 1989, Page 524—539)。

如果电路设计、布局、和/或速度要求限定一个单向移位器,可以通过使图 19 中的移位器/多路转换器“对”96/128 基本完全一样的电路来实现,并利用“一对”作为向高速缓冲存贮器重新加载缓冲器 112 的输入,而另“一对”作为存贮返回缓冲器 98 的一个输出来实

现。存在很多种选择方案,其中的一些将通过技术和综合说明来限定,然而基本原理没有变化。

按照总的系统操作的角度来看,在第二存贮级 110 的动态随机存取存贮器上的理想输入/输出接口应该是图 7 所示的可选 32/256 位通路。在第二存贮级 110 的动态随机存取存贮器和静态随机存取存贮器(存贮返回缓冲器 98 和高速缓冲存贮器重新加载缓冲器 112)之间的转换通路将是 256 位,而第一和第三存贮级(存贮返回缓冲器/高速缓冲存贮器重新加载缓冲器)之间的转换通路将是 32 位,而不论其它通路情况如何。对于从第三存贮级 L3 向第一和第二存贮级的一个漏掉的数据块的重新加载,第一组 32 字节包含产生该遗漏的双字。第一组 32 字节被选通到高速缓冲存贮器重新加载缓冲器 112,而负载通过(*load-thru*)通路被启动。被当前存取的双字由负载通过(*load-thru*)解码器 130 进行解码并通入中央处理单元。如果希望的话,可选择在随后周期上执行相同的功能。

眼前主要的问题是将每周期 32 位的八个周期转换成第二存贮级 110 的动态随机存取存贮器。如果第二存贮级 110 具有一个用以适于解码、选择的 32 位输入通路 和一个输入缓冲寄存器(*IBR*)132, (类似于高速缓冲存贮器重新加载缓冲器 112),则在每一个周期这 32 位既可以输入高速缓冲存贮器重新加载缓冲器 112,也可以输入缓冲寄存器 132。在八个周期结束时,可以将输入缓冲寄存器 132 加载到动态随机存取存贮器。

如果第二存贮级只有 256 位输入能力,则可以按以下方法对第二存贮级 110 加载。在高速缓冲存贮器重新加载缓冲器 112 满载后,可以在一个机器周期中将数据复制到存贮返回缓冲器 98,然后从该处作为 256 位加载到第二存贮级 110 中。如果第二存贮级 110 有一个输入缓冲寄存器 132,该寄存器 132 具有用于排列位的 256 位输入通路,则需要将一个附加周期用于存贮返回缓冲器 98,以对输入缓冲寄存器 132 加载,而用于该输入缓冲寄存器 132 的大约八个周期被写入动态随机存取存贮器。然而,在对输入缓冲寄存器 132 加载之后,存贮返回缓冲器 98 被释放。

利用图 7 和图 19 所示的高速缓冲放大器结构,当一个数据块从第二存贮级 110 被重新加载到第一存贮级时,引起遗漏的双字将在高速缓冲存贮器重新加载缓冲器满载的同时被加载在下一个周期。如果数据需要来自高速缓冲存贮器重新加载缓冲器 112,则可以将整个高速缓冲存贮器重新加载缓冲器加载到静态随机存取存贮器,而所希望的双字就可以同时进行存取。

为了从第三存贮级 L3 向静态随机存取存贮器重新加载,在高速缓冲存贮器重新加载缓冲器 112 中的数据能够以相同的方式进行存取,除非是全数据块可能不存在而且仍然要求高速缓冲存贮器重新加载缓冲器 112 向存贮单元阵列 78 的连续加载。

以使高速缓冲存贮器重新加载缓冲器 112 中的数据对中央处理单元成为可存取的方式是另一种设计方案(参看,例如 Matick,

R. E. et al. (*Architectural implication in the design of microprocessors*) *IBM system Journal*, Vol. 23, No. 3, 1984, Page 264—280 ;
Matick , R . E . et al, January 1989, 上面已提到过; 以及 Radin, G.
“The 801 minicomputer” *IBM Journal of research and Development*, Vol. 27, No. 3 1983, page 237—246)。

以下是最糟情况下的高速缓冲存储器重新加载的一个范例。在该范例中, 第一和第二存储级二者都有一个遗漏而且二者都有一个被存储返回的数据块, 即, 第一存储级有一个到第二存储级的存储返回, 第二存储级有一个到第三存储级的存储返回, 而该重新加载的数据块必须既输入到第一也输入第二存储级。对于图 19 的构形, 完成该过程的步骤将如下所述。

步骤 1: 周期 T_1

存取静态随机存取存储器和动态随机存取存储器目录; 二者都给出遗漏及舍去指示, 即, B_3 出自第一存储级, B_9 出自第二存储级, 而数据块 B_6 则从第三存储级被重新加载。

步骤 2: 周期 T_2

将 B_3 输入到静态随机存取存储器并寄存在存储返回缓冲器中。对于 B_9 启动动态随机存取存储器的存储。对于 B_6 启动第三存储级的存储。

步骤 3: 周期 T_3

将 B_3 从存储返回缓冲器转换到输入缓冲寄存器 (一个周期)。

步骤 4: 周期 $T_2 + TD$

(TD = 在机器周期中的动态随机存取存储器存取时间)

数据块 9 进入输出缓冲寄存器。

步骤 5a: 周期 $T_2 + TD + 1$

启动输入缓冲寄存器(包含 B_3)至动态随机存取存储器的一个写入周期。将 B_9 从输出缓冲寄存器转换到高速缓冲存储器重新加载缓冲器(一个周期)。

步骤 5b: 周期 $T_2 + TD + 2$

将高速缓冲存储器重新加载缓冲器(B_9)转换至存储返回缓冲器(一个周期)。

步骤 6: 周期 $T_2 + TD + 1 + TD$

现在 B_3 在动态随机存取存储器中 (B_9 写入存储返回缓冲器)。

步骤 7: 周期 $T_2 + TM$ (TM 为第三存储级存取时间)

首先将 B_6 的 32 字节(包含产生遗漏的双字)转换到输入缓冲寄存器并输入到高速缓冲存储器重新加载缓冲器,将产生遗漏的双字加载通过中央处理单元。从第三存储级来的重新加载继续用于七个附加周期。

步骤 8: 周期 $T_2 + TM + 7$

数据块 B_6 被满载地重新加载到输入缓冲寄存器并输入到高速缓冲存储器重新加载缓冲器。

步骤 9: 周期 $T_2 + TM + 8$

起动 $B6$ (在输入缓冲寄存器中) 至阵列的动态随机存取存储器写入周期。起动存储器返回缓冲器的存储器返回至第三存储级。完整的存储器返回到第三存储级将需要 T_M+7 周期。

步骤 10: 周期 $T_2+T_M+8+(T_M+7)$

存储器返回缓冲器 ($B9$) 满载地存储器返回到第三存储级。

在上述过程中, 内含地假设有:

$$T_M \geq 2 \times T_D + 1 \text{ 周期} \quad (9)$$

因此, 步骤 6 将自动地在步骤 7 之前产生。如果不是这种情况, 则重新加载控制器将必须提供必要的控制, 以便防止 $B3$ 和 $B6$ 的加载之间的矛盾进入到动态随机存取存储器上的输入缓冲寄存器之中。另一方面, 来自第三存储级的 32 位数据通路仅仅能够被加载到高速缓冲存储器重新加载缓冲器, 而不直接输入到第二存储级。其后, 当来自第三存储级的全数据块转换完成时, 第二存储级可利用存储器返回缓冲通路进行重新加载。

说明书附图

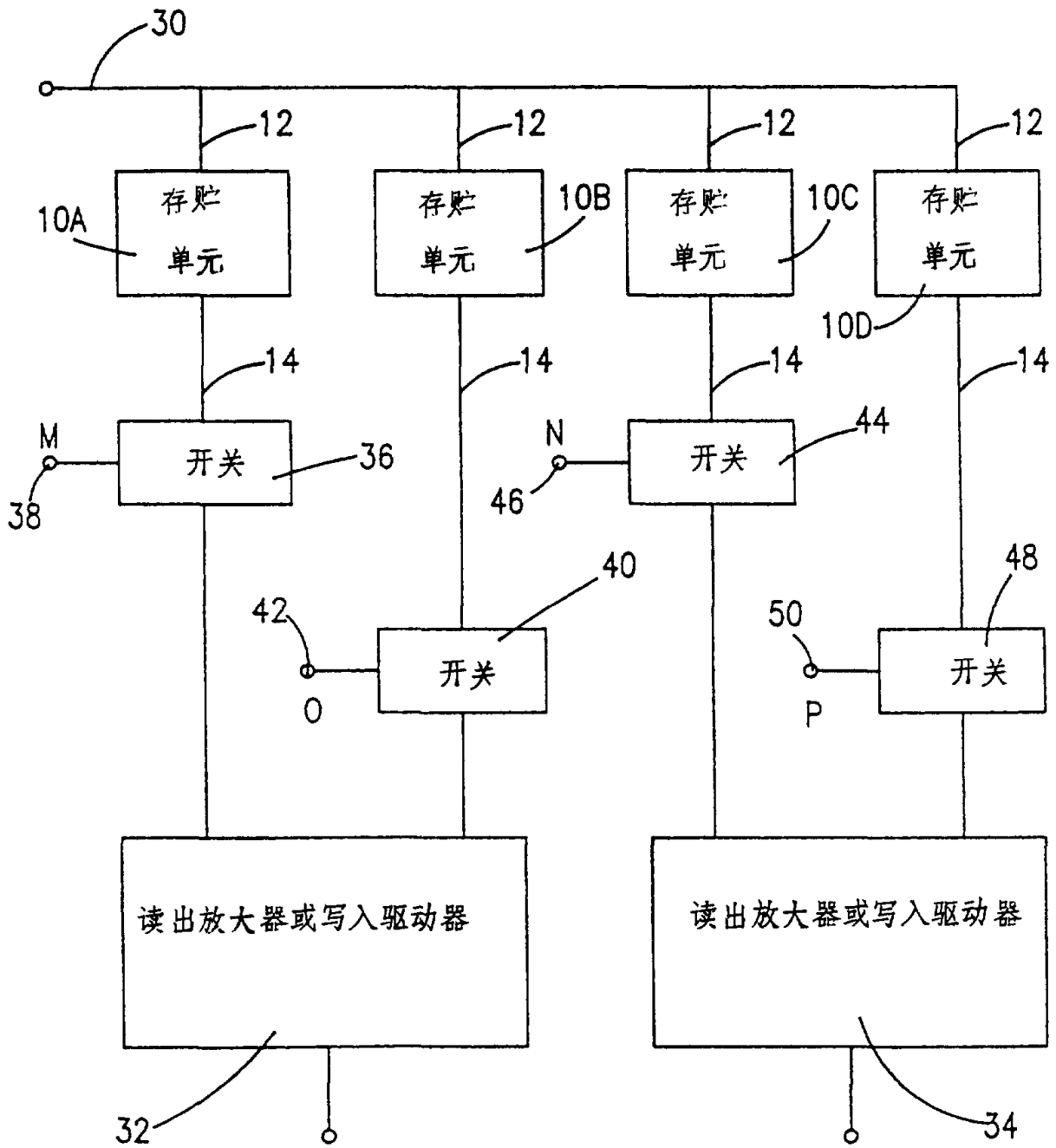


图1

图2

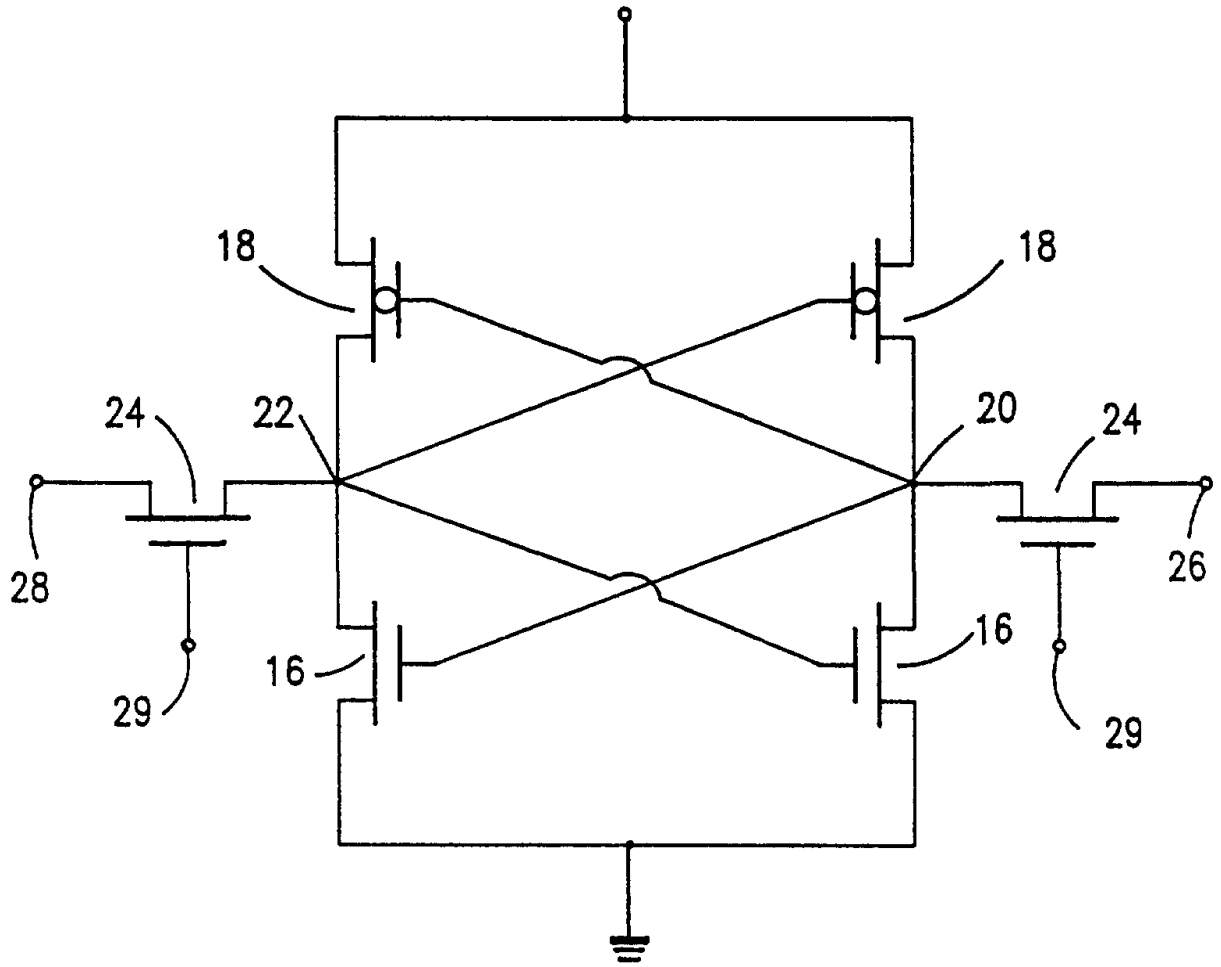
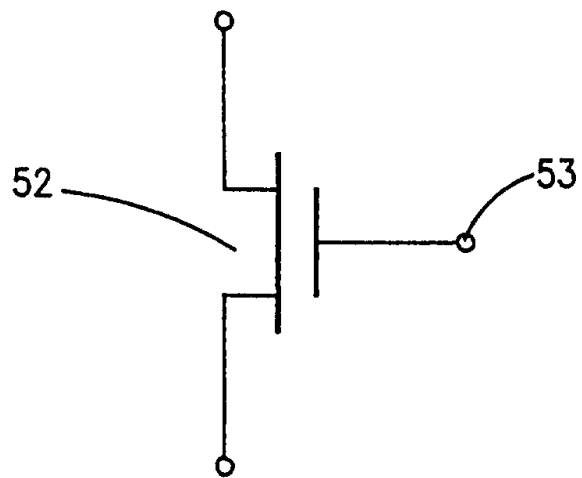


图3



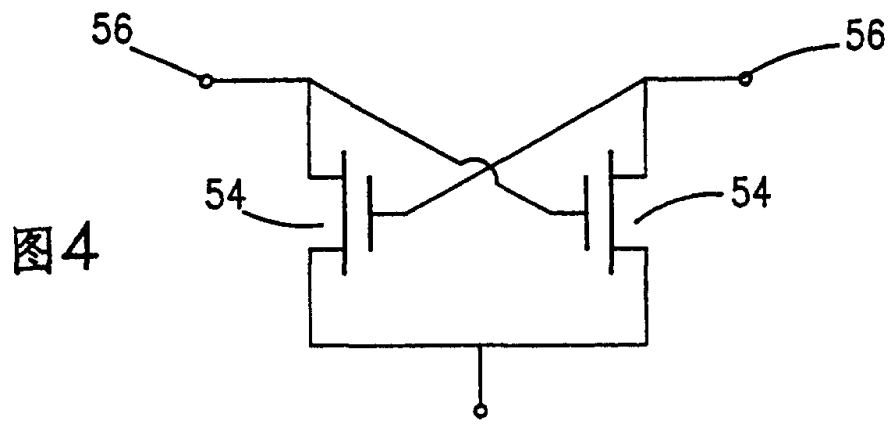


图4

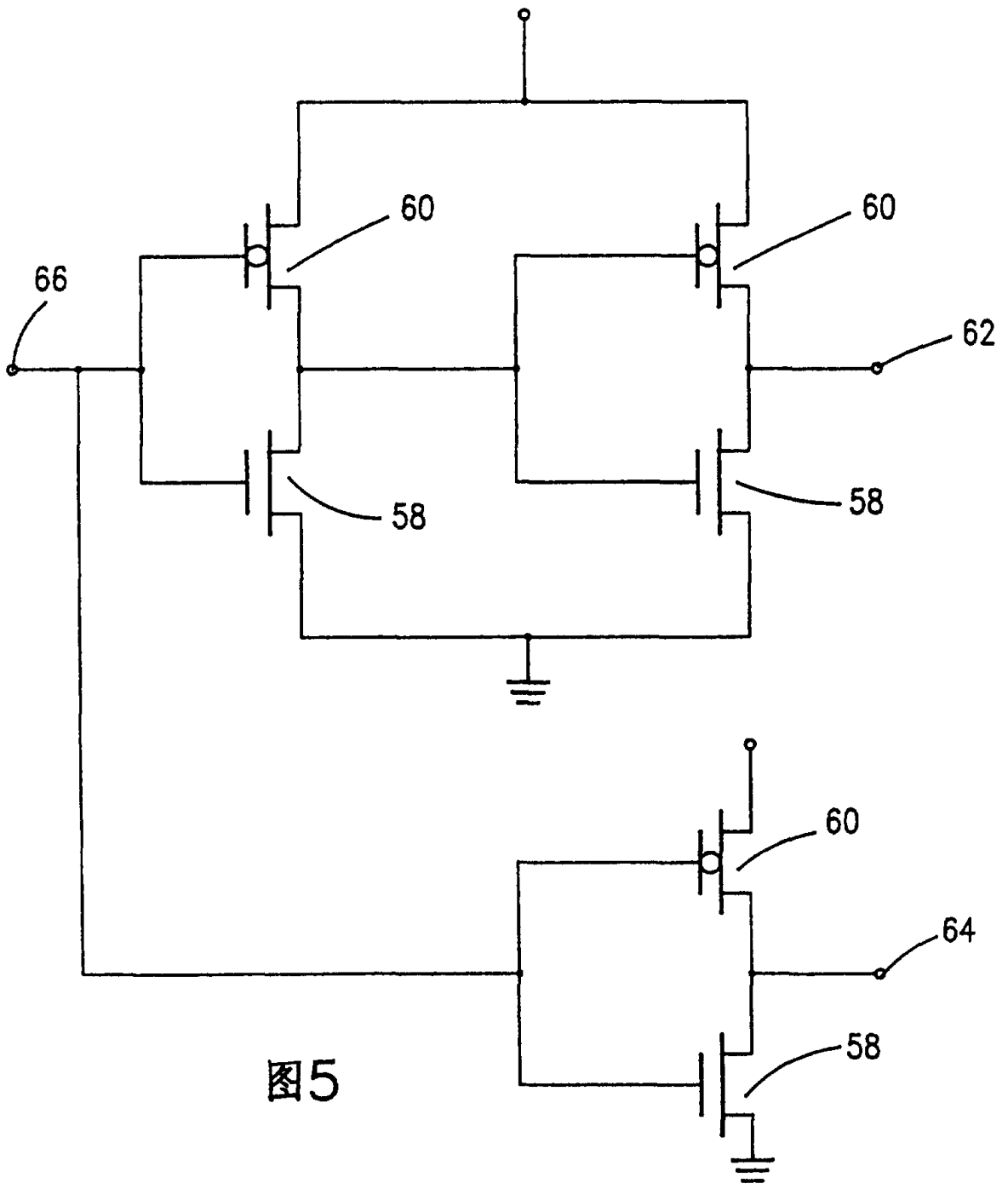


图5

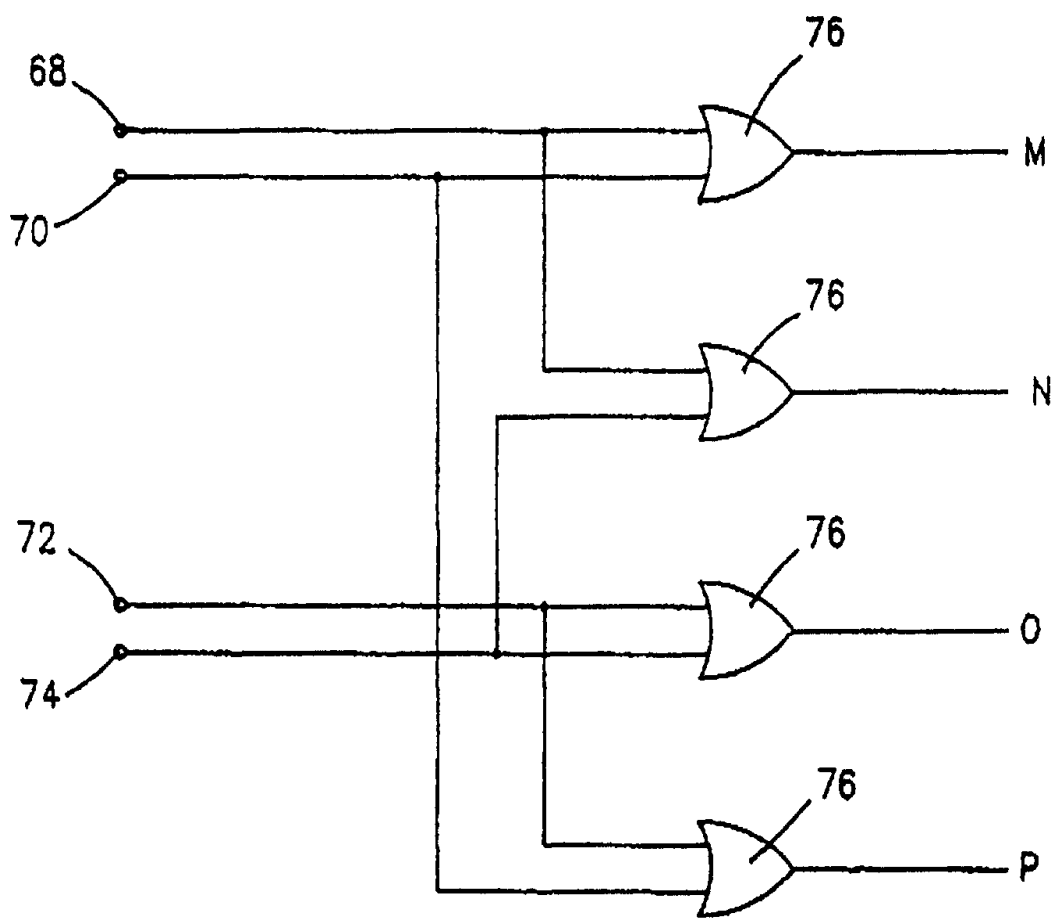


图6

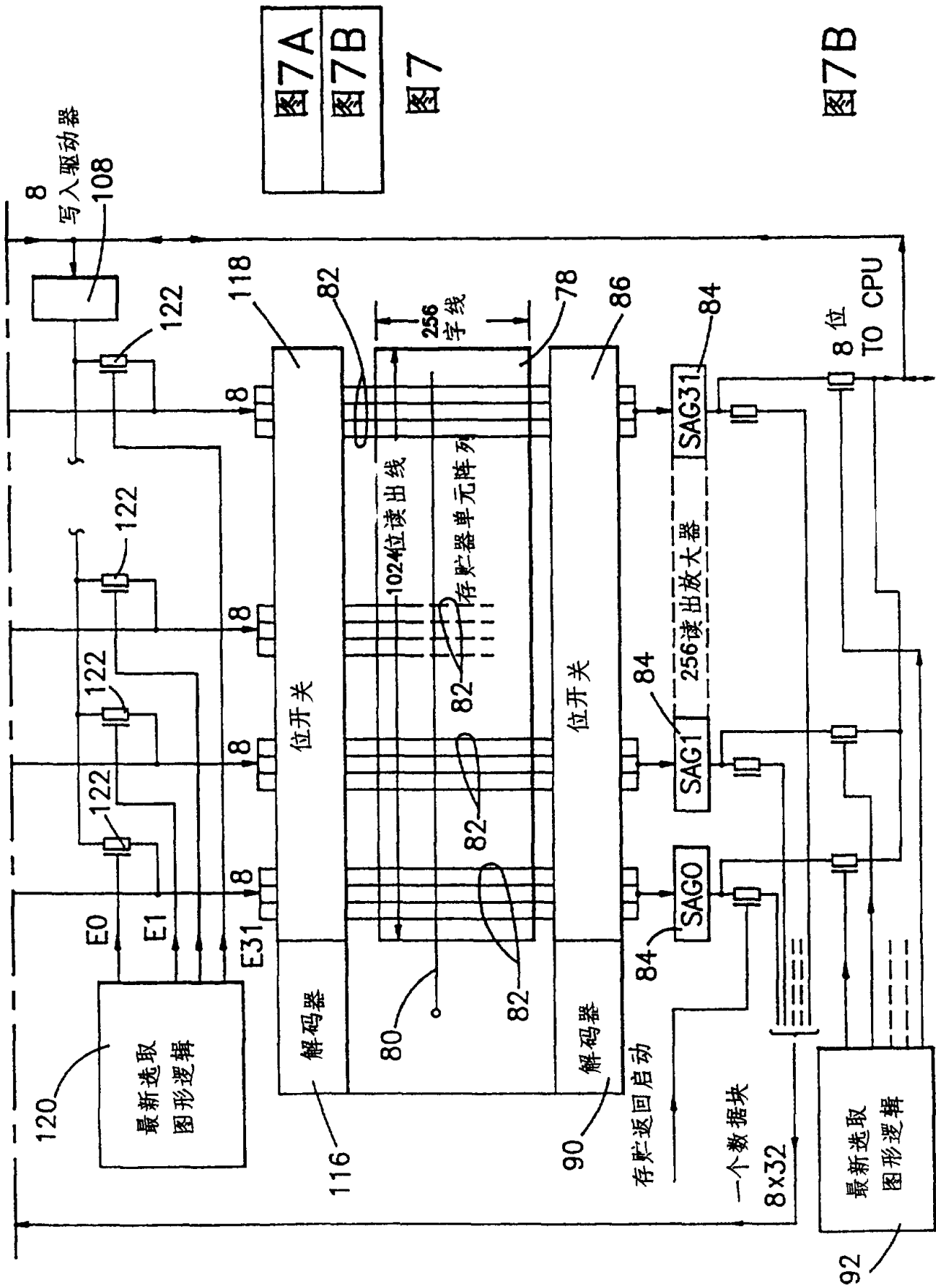


图7A
图7B

图7

图7B

在每个数据块上的符号是
逻辑字的号数

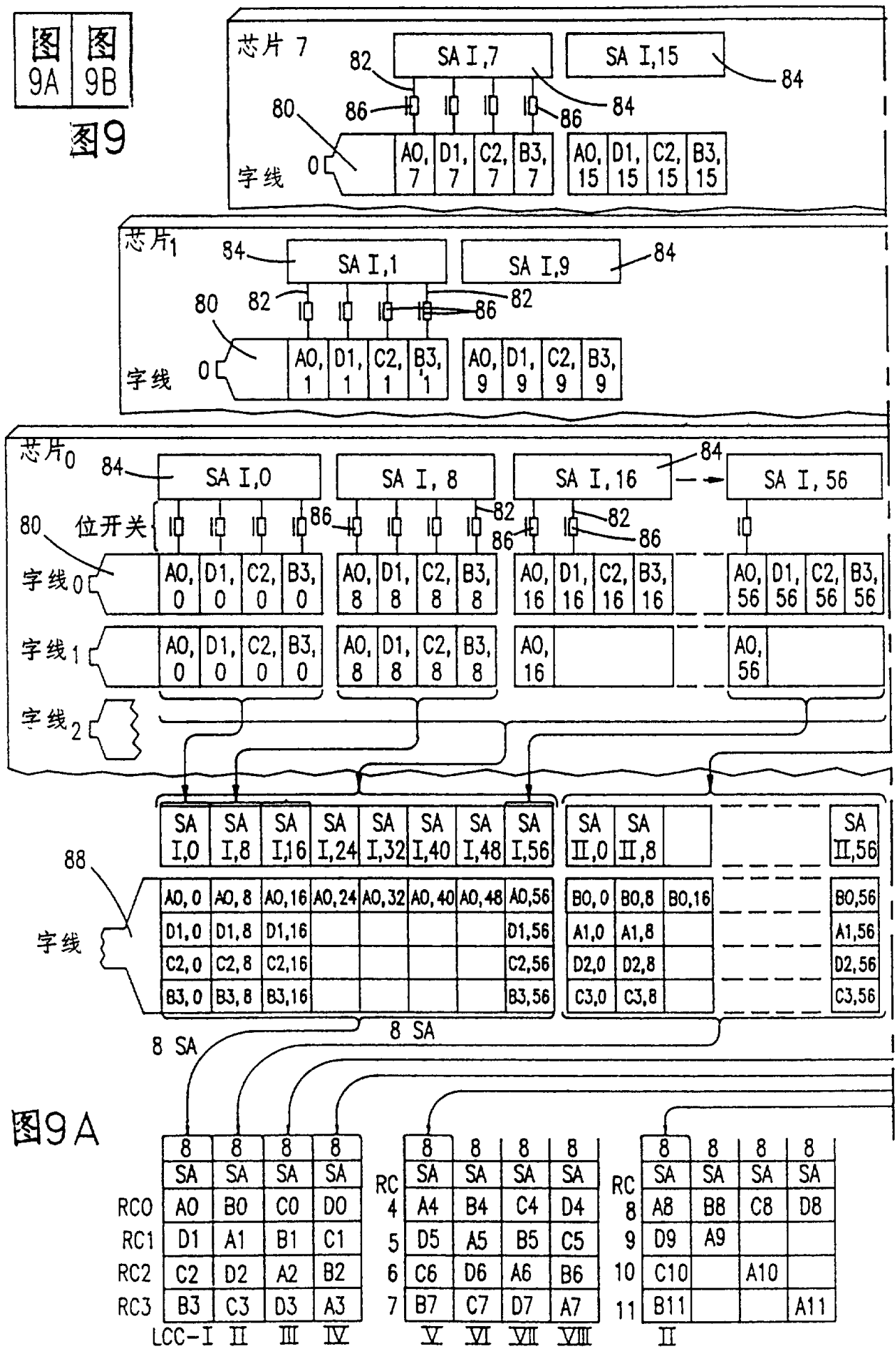
				排相同级	
	A ₀	B ₀	C ₀	D ₀	RC0
	D ₁	A ₁	B ₁	C ₁	RC1
	C ₂	D ₂	A ₂	B ₂	RC2
	B ₃	C ₃	D ₃	A ₃	RC3
LCC-I	II	III	IV		

拉丁方格列相同级

图8

图 9A 图 9B

图 9



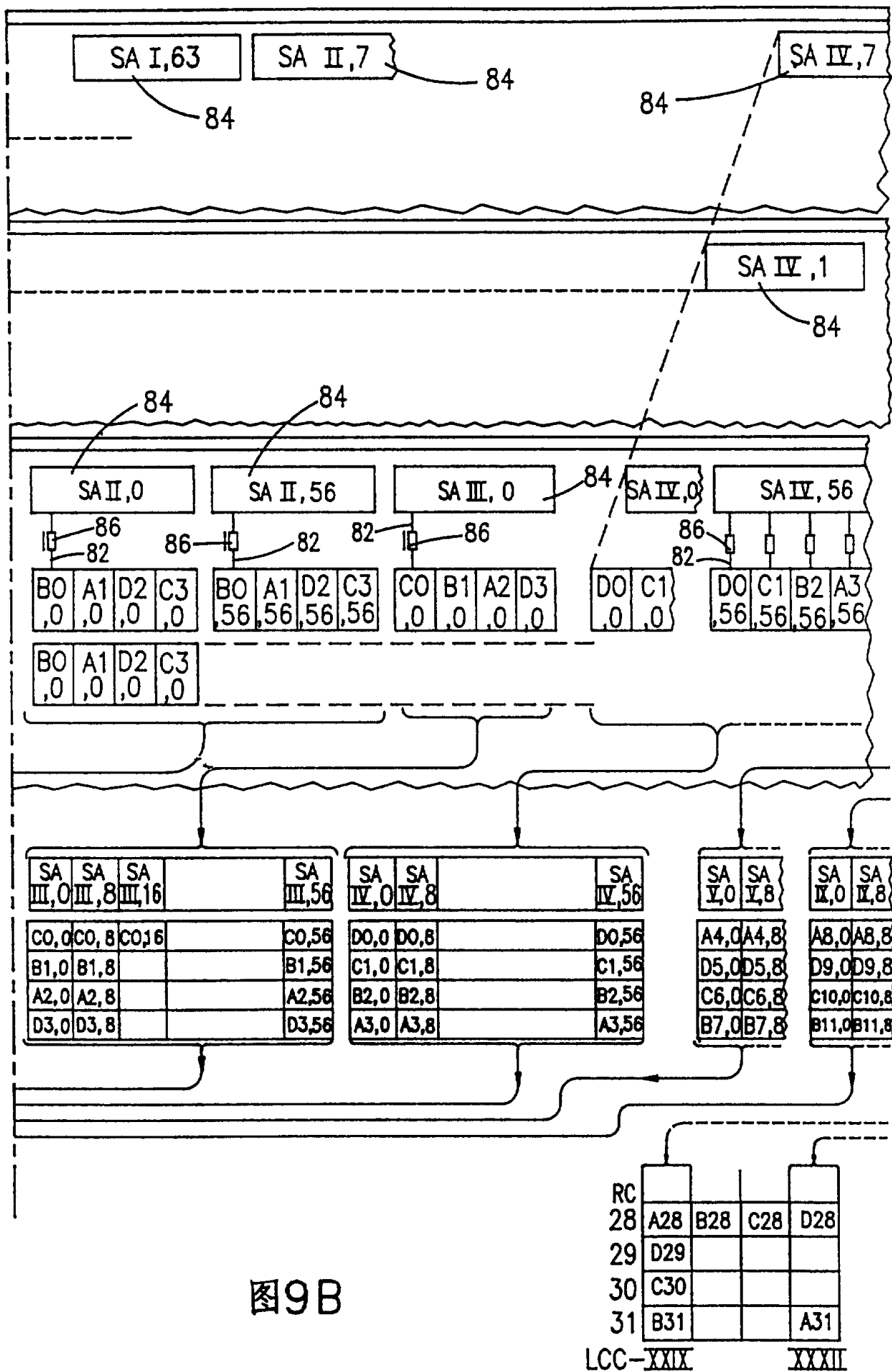


图9B

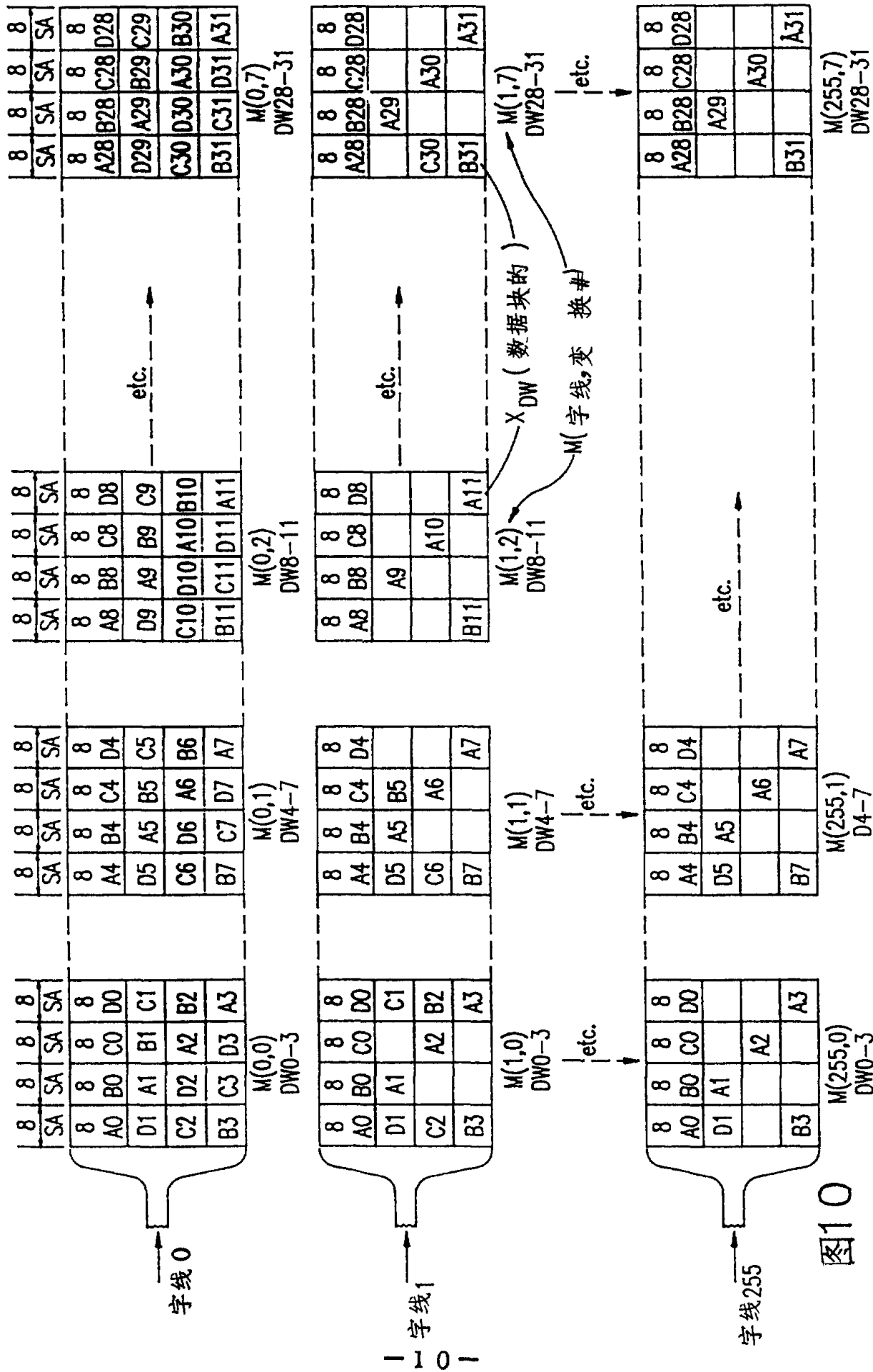


图10

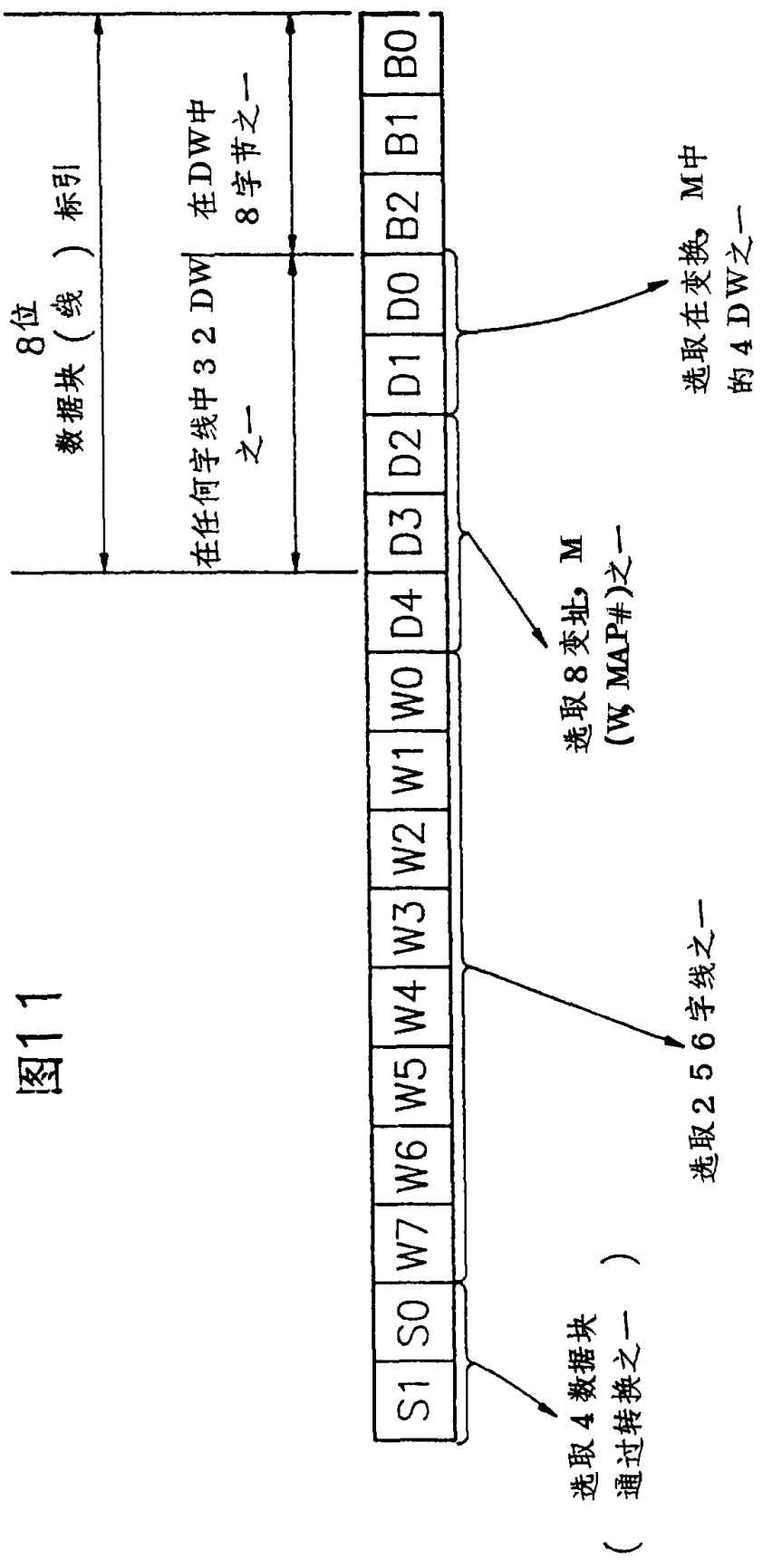


图11

图1 2A

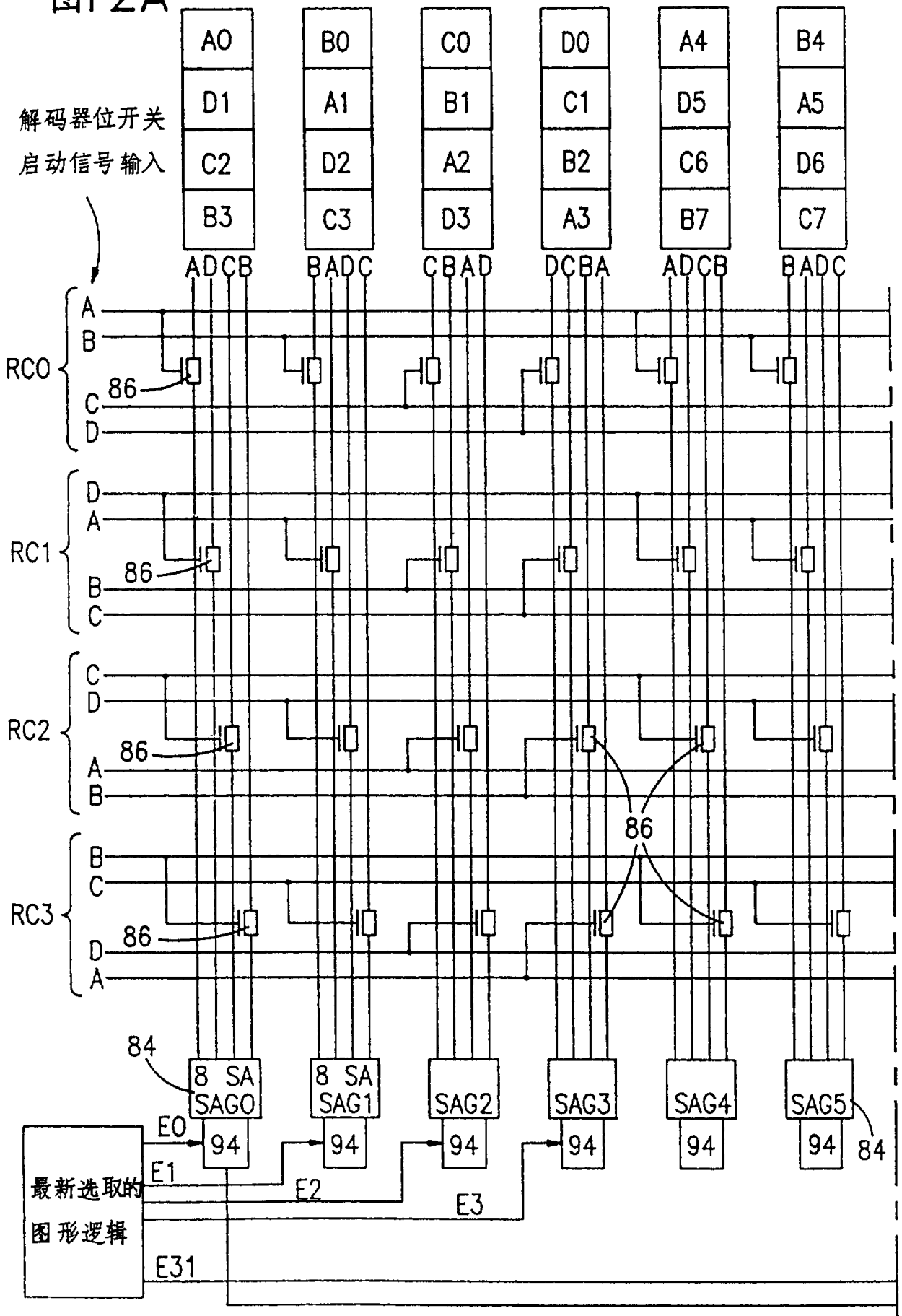
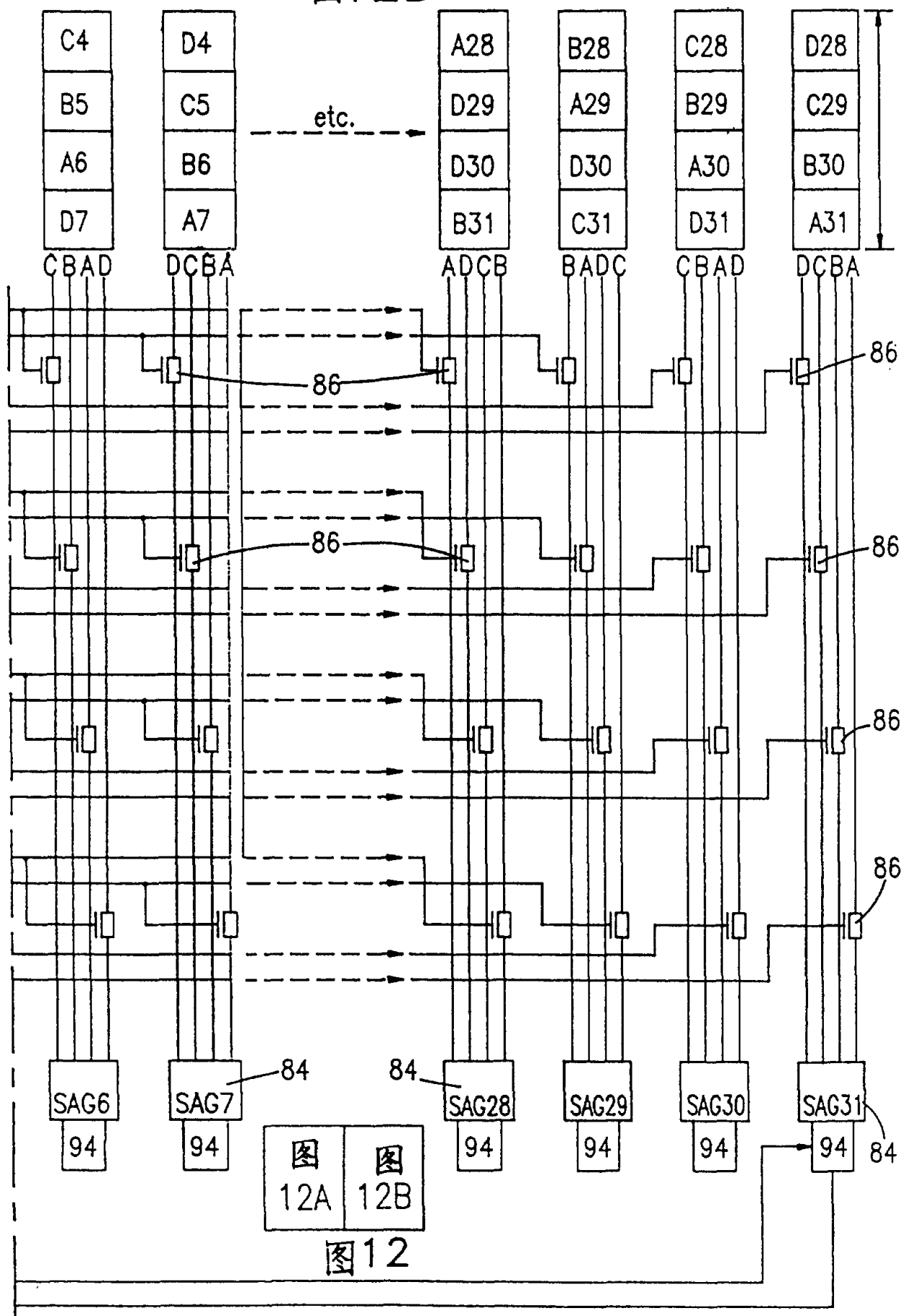


图12B



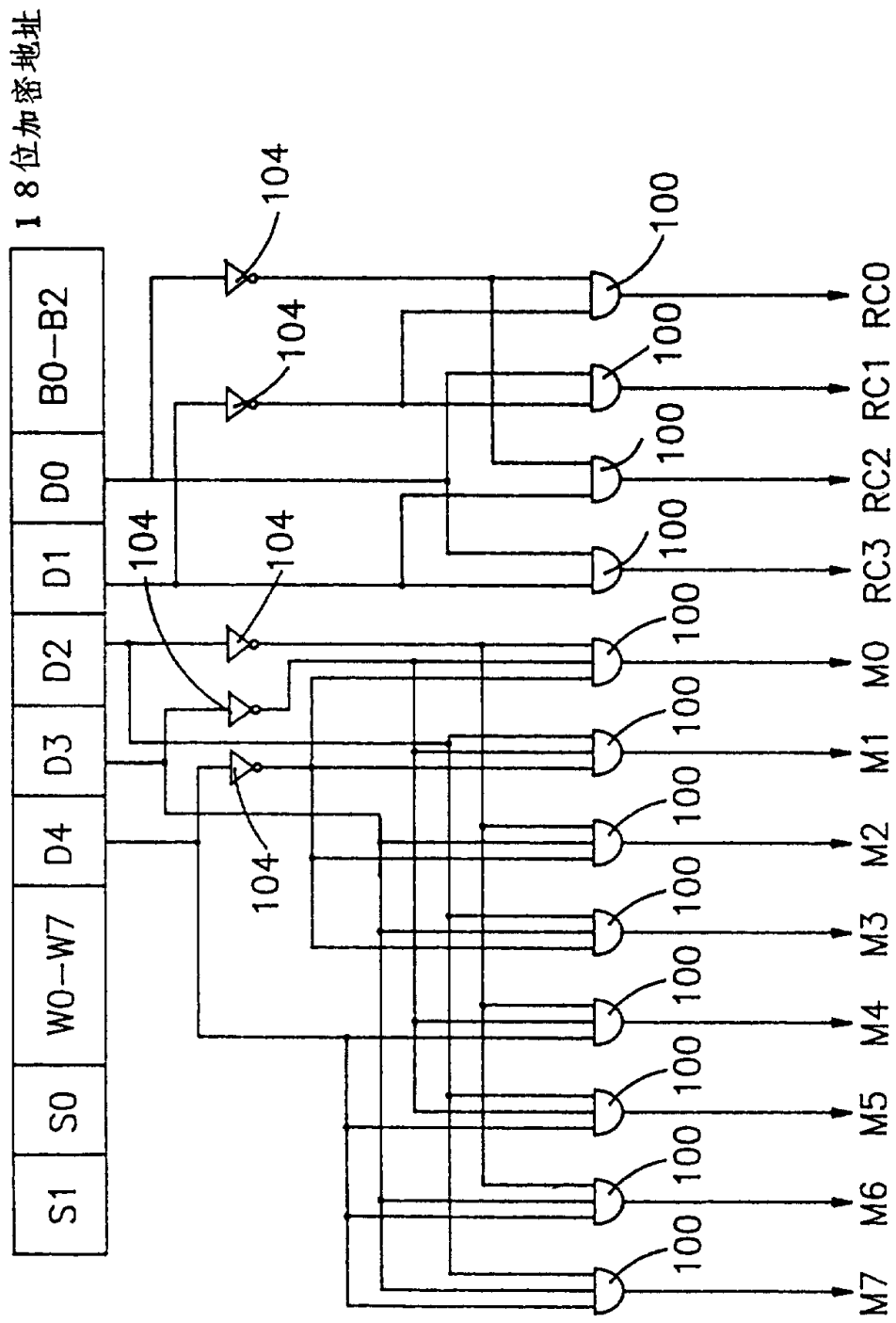


图14

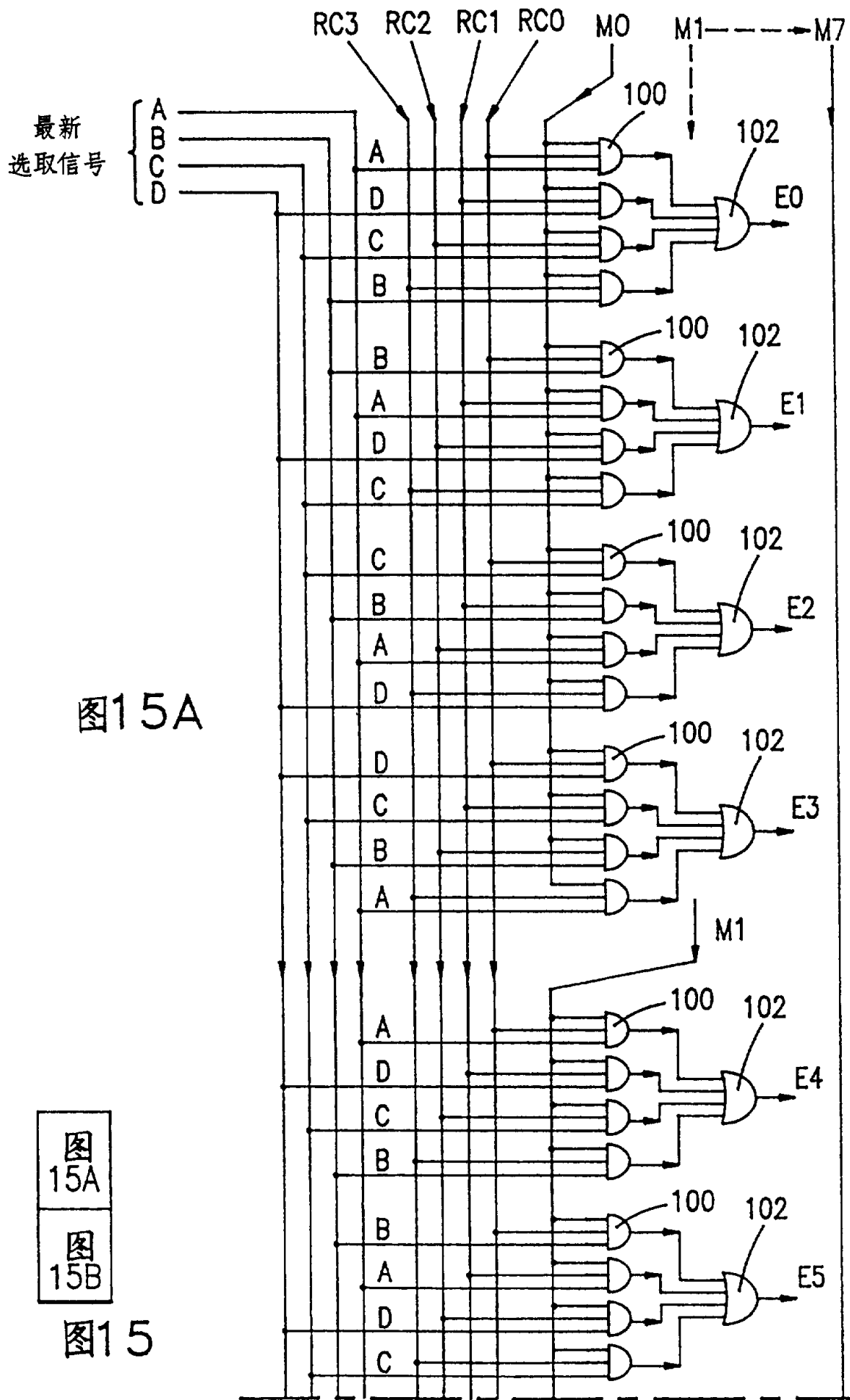


图15B

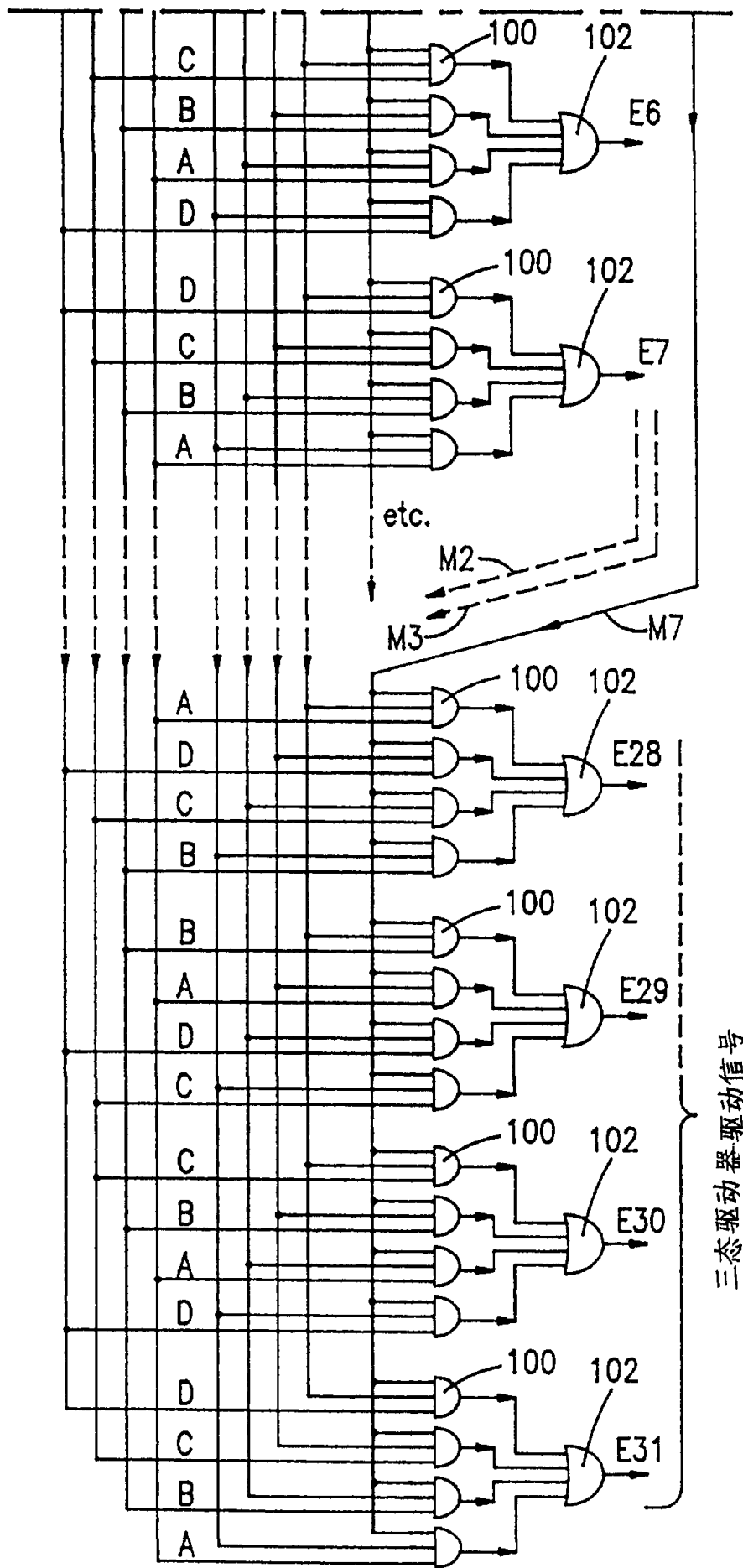
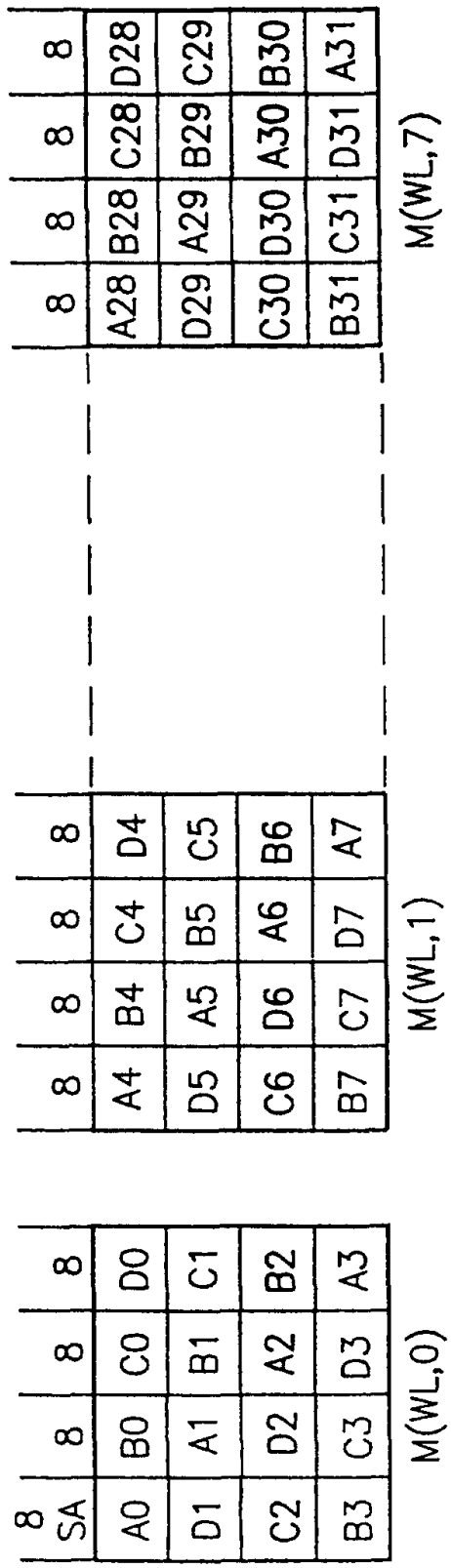
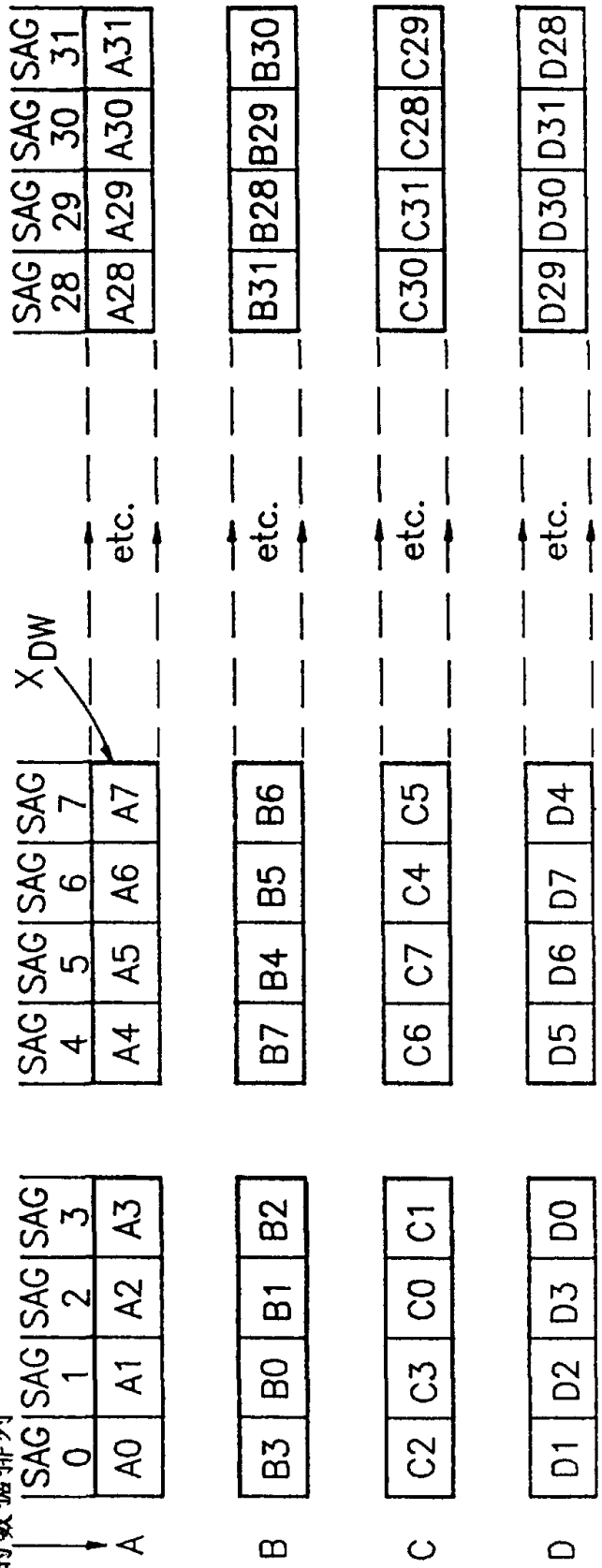


图16



在用于数据块的
S A 中的数据排列



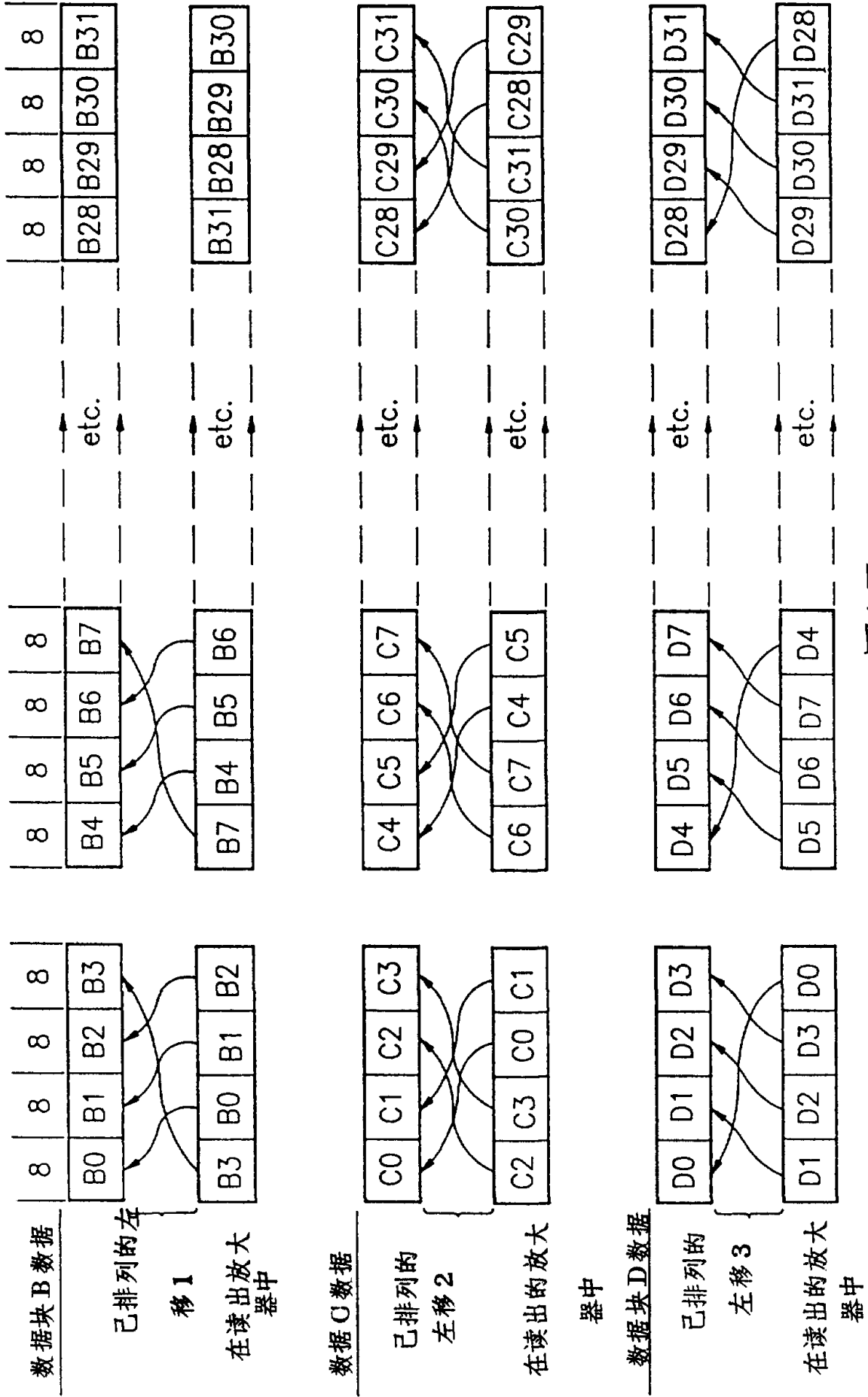


图17

图18

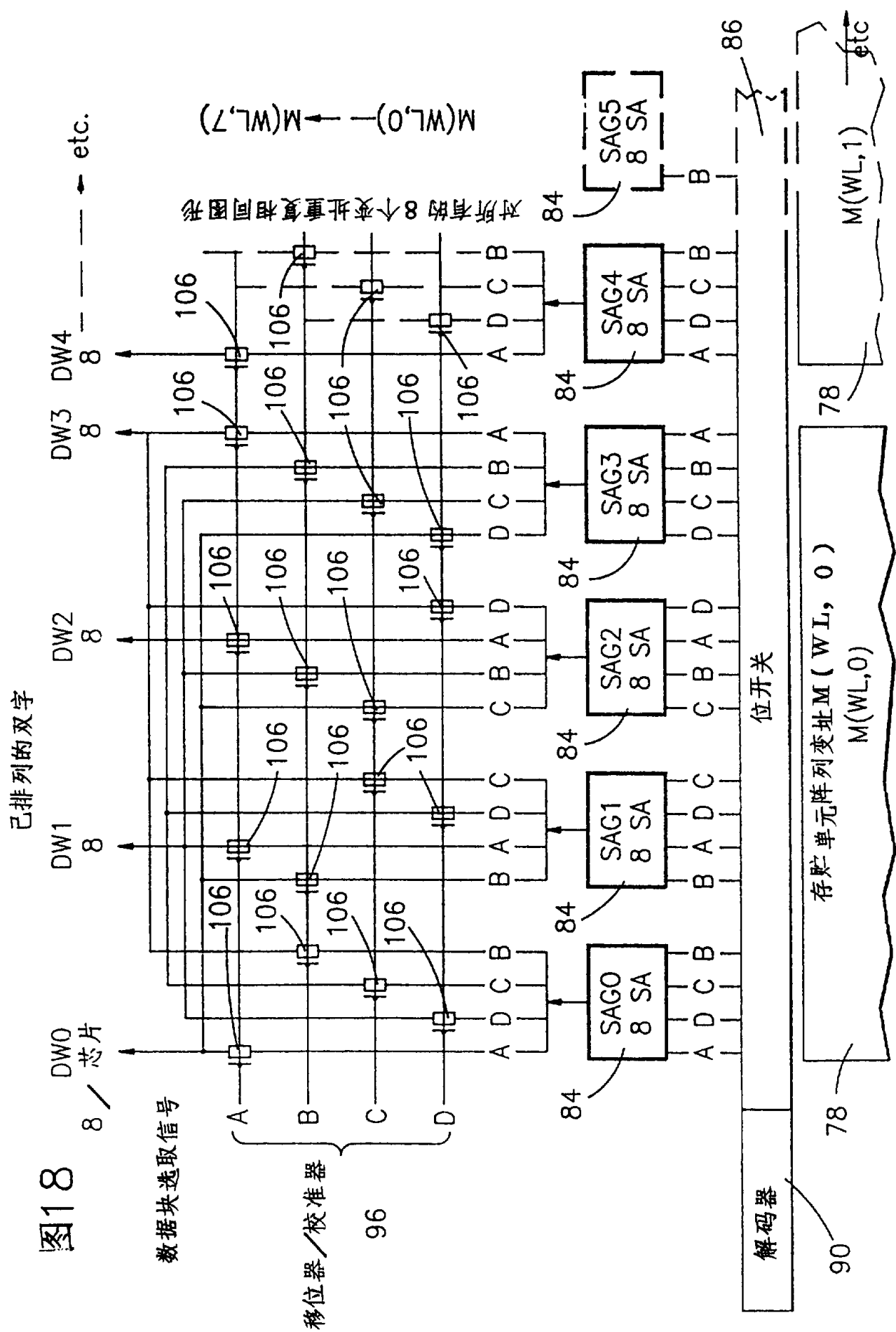


图19A

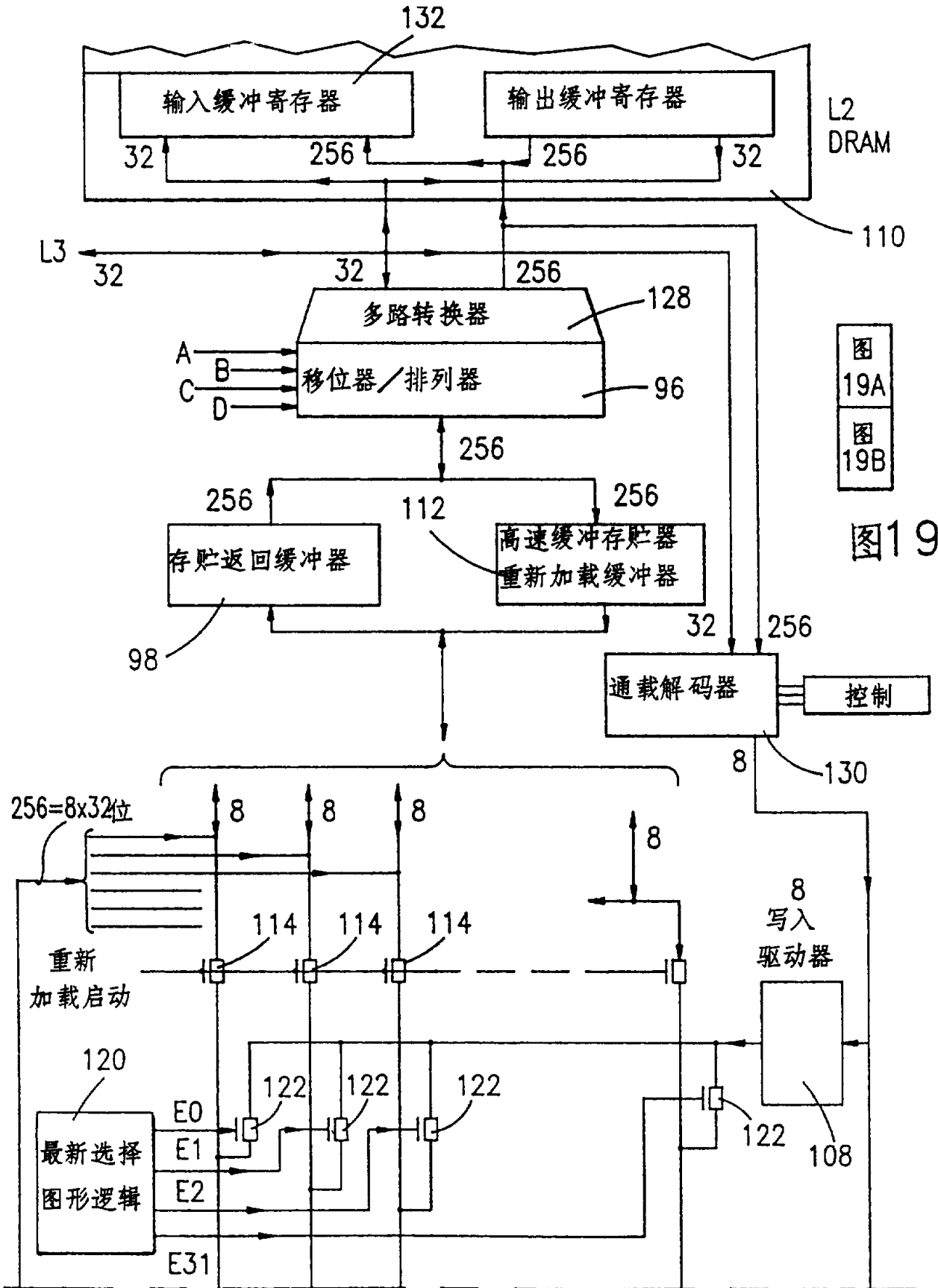


图
19A
图
19B

图19

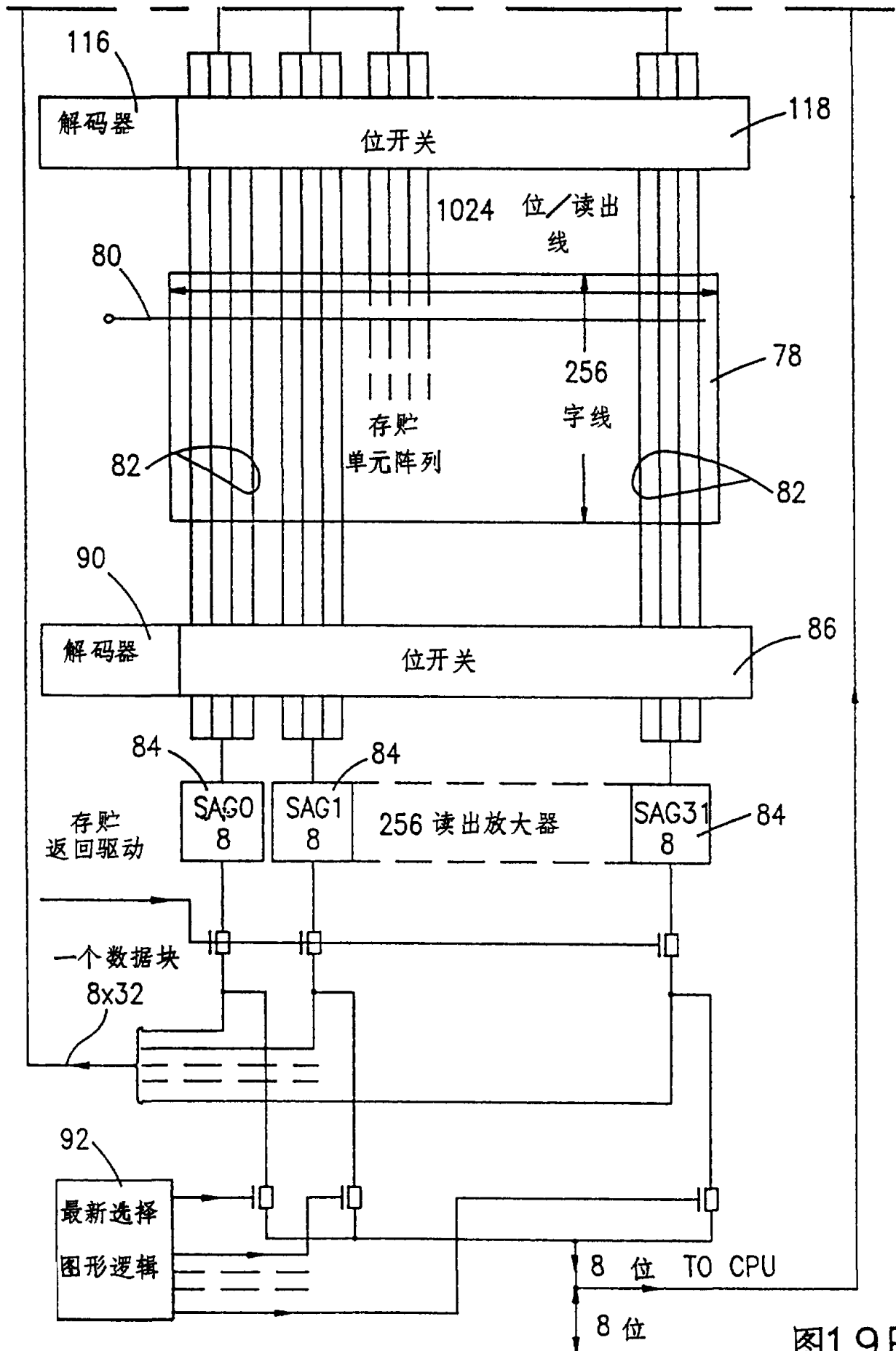


图19B