



# [12] 发明专利申请公开说明书

[21] 申请号 02826091.0

[43] 公开日 2005 年 4 月 20 日

[11] 公开号 CN 1608392A

[22] 申请日 2002.12.19 [21] 申请号 02826091.0  
 [30] 优先权  
 [32] 2001.12.27 [33] US [31] 10/034,340  
 [86] 国际申请 PCT/US2002/041311 2002.12.19  
 [87] 国际公布 WO2003/059004 英 2003.7.17  
 [85] 进入国家阶段日期 2004.6.25  
 [71] 申请人 英特尔公司  
 地址 美国加利福尼亚州  
 [72] 发明人 M·鲁伊赫尔

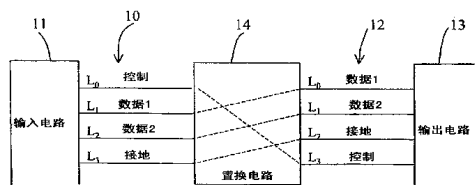
[74] 专利代理机构 上海专利商标事务所有限公司  
 代理人 李家麟

权利要求书 3 页 说明书 13 页 附图 9 页

[54] 发明名称 信号置换

[57] 摘要

一种构建配置网络的方法。根据要置换的信号数 N 的一组整数因数以及预先选择类型的开关来为置换网络的层选择一种配置。根据所选择的配置通过使用预先选择类型的开关构建层中的置换网络。



1. 一种方法，其特征在于，包括：  
根据要置换的信号数  $N$  的一组整数因数以及预先选择类型的开关，为置换网络  
5 的层选择一种配置；以及  
根据所选择的配置，利用预先选择类型的开关来按层构建置换网络。
2. 如权利要求 1 所示的方法，其特征在于，每种类型的开关都能够从一定数量的  
信号中选择一个信号，对于不同类型的开关，该数量是不同的。
3. 如权利要求 2 所述的方法，其特征在于，每个整数因数对应于一种类型的开  
10 关可以从其中进行选择信号的数量。
4. 如权利要求 1 所述的方法，其特征在于，还包括选择所述一组整数因数  $w_1$ ，  
 $w_2$ ，...， $w_D$  ( $D$  为整数)，以使  $N=w_1 \times w_2 \times \dots \times w_D$ 。
5. 如权利要求 4 所述的方法，其特征在于，所述置换网络被配置成具有  $2D-1$   
层的开关，这些开关包括  $w_1: 1$ ， $w_2: 1$ ，...，和  $w_D: 1$  开关或者由  $w_1: 1$ ， $w_2: 1$ ，...，  
15 和  $w_D: 1$  开关构成。
6. 一种方法，其特征在于，包括：  
接收  $N$  个信号；以及  
使用置换网络将这  $N$  个信号重新排序，其中该置换网络是由开关层构成的，且  
这些开关层具有基于  $N$  的一组整数因数以及预先选择类型的开关的配置。
- 20 7. 如权利要求 6 所述的方法，其特征在于，每层都具有  $N$  个相同类型的开关，  
每种开关都具有预定数量的输入端子和一个输出端子。
8. 如权利要求 7 所述的方法，其特征在于，置换网络的每一层都将所述  $N$  个信  
号分成信号子集并置换信号子集的排序，子集中信号的数量等于层中每个开关拥有  
的输入端子数。
- 25 9. 如权利要求 7 所述的方法，其特征在于，通过将多维坐标分配给所述开关来  
构建所述置换网络，层中的每个开关都具有不同的坐标，并配置所述开关从而当信  
号从一个层中的第一开关传播到下一个层中的第二开关时，这两个开关的坐标仅在  
一个维数上不同。
10. 如权利要求 6 所述的方法，其特征在于，所述整数因数是  $w_1$ ， $w_2$ ，...， $w_D$  ( $D$   
30 是整数)，以使  $N=w_1 \times w_2 \times \dots \times w_D$ ，且预先选择类型的开关包括  $w_1: 1$ ， $w_2: 1$ ，...，  
和  $w_D: 1$  开关。
11. 一种装置，其特征在于，所述装置包括：

N 个输入端子, N 是整数;

N 个输出端子; 以及

5 置换网络, 它被配置成形成以任意顺序将输入端子连接到输出端子的非阻断信号路径, 所述置换网络由不同类型的开关的层构成, 每层都具有同一类型的相同数量的开关, 每个类型的开关都能从预定数量的信号中选择出一个信号。

12. 如权利要求 11 所述的装置, 其特征在于, 每个开关都具有输入端子和输出端子, 第一层中开关的输入端子耦合到所述装置的 N 个输入端子, 最后一层中开关的输出端子耦合到所述装置的 N 个输出端子, 且对于最后一层以外的所有层, 开关的输出端子都连接到下一层中开关的输入端子。

10 13. 如权利要求 12 所述的装置, 其特征在于, 层的数量和连续层的开关之间的连接是基于 N 的一组整数因数以及所使用的开关的类型。

14. 如权利要求 13 所述的装置, 其特征在于, 所述整数因数是  $w_1, w_2, \dots, w_D$  (D 是整数), 以使  $N = w_1 \times w_2 \times \dots \times w_D$ , 且所使用开关的类型包括  $w_1: 1, w_2: 1, \dots,$  和  $w_D: 1$  开关。

15 15. 如权利要求 14 所述的装置, 其特征在于, 所述置换网络被配置成具有  $2D-1$  层的开关, 每一层都置换信号路径的不同子集的排序。

16. 如权利要求 15 所述的装置, 其特征在于, 对于每个开关的第 p 层, p 的范围从 1 到 D,  $w_p: 1$  开关被配置成构成能置换  $w_p$  个信号路径顺序的  $w_p$  乘  $w_p$  置换器, 而对于每个开关的第 q 层, q 的范围从  $D+1$  到  $2D-1$ ,  $w_{2D-q}: 1$  开关被配置成构成能  
20 置换  $w_{2D-q}$  个信号路径顺序的  $w_{2D-q}$  乘  $w_{2D-q}$  置换器。

17. 如权利要求 16 所述的装置, 其特征在于, 第二层到第  $(2D-1)$  层中每个置换器的每个输入端子都连接到前一层中不同置换器的输出端子。

18. 一种装置, 其特征在于, 包括:

第一装置, 它被配置成产生具有第一排序的 N 个信号;

25 第二装置, 它被配置成接受第二排序的 N 个信号; 以及

置换网络, 它被配置成接收具有第一排序的 N 个信号并将这 N 个信号重新排序, 从而所述 N 个信号具有第二装置可接受的第二排序, 由不同类型的开关层构成所述置换网络, 每个层都具有相同类型的相同数量的开关, 每个类型的开关都能从预定数量的信号中选出一个信号。

30 19. 如权利要求 18 所述的装置, 其特征在于, 层的数量和连续层的开关之间的连接是基于 N 的一组整数因数和所使用开关的类型。

20. 如权利要求 19 所述的装置, 其特征在于, 所述第二装置是存储器。

21. 如权利要求 20 所述的装置, 其特征在于, 所述第一装置是计算机主板。

22. 如权利要求 19 所述的装置, 其特征在于, 所述整数因数是  $w_1, w_2, \dots, w_D$  ( $D$  是整数), 以使  $N=w_1 \times w_2 \times \dots \times w_D$ , 且所使用开关的类型包括  $w_1: 1, w_2: 1, \dots,$  和  $w_D: 1$  开关。

5 23. 如权利要求 22 所述的装置, 其特征在于, 所述置换网络被配置成具有  $2D-1$  层的开关, 每一层都具有  $N$  个相同类型的开关, 每一层都置换  $N$  个信号的不同子集的顺序。

24. 一种存储在计算机可读媒介上的计算机程序, 其特征在于, 所述计算机程序用于使得计算机系统执行以下功能:

10 将多维坐标分配给一组  $N$  个信号中的每一个,  $N$  是整数;

在连续操作中, 在每次操作期间改变这  $N$  个信号的坐标的特定维数, 从而在每次操作后没有两个信号具有相同的坐标, 从而在所述连续操作后, 这  $N$  个信号的坐标与一组目标坐标相匹配。

15 25. 如权利要求 24 所述的计算机程序, 其特征在于, 还使得计算机系统执行以下功能: 找到  $N$  的整数因数  $w_1, w_2, \dots, w_D$ , 以使  $N=w_1 \times w_2 \times \dots \times w_D$ , 其中  $D$  是整数。

26. 如权利要求 25 所述的计算机程序, 其特征在于, 所述多维坐标是  $[x_1, x_2, \dots, x_D]$ , 其中对于每个  $k, x_k$  的范围 1 到  $w_k, k$  从 1 到  $D$ 。

20 27. 如权利要求 26 所述的计算机程序, 其特征在于, 通过交换一对信号的坐标来改变所述  $N$  个信号的坐标。

28. 一种用于将  $N$  个信号重新排序的装置, 其特征在于, 包括:

( $2D-1$ )层的开关,  $D$  为整数, 第  $n$  层和第  $(2D-n)$  层具有  $w_n$  乘  $w_n$  个开关,  $n$  的范围从 1 到  $D$ , 且  $w_1$  到  $w_D$  是  $N$  的整数因数从而  $N=w_1 \times w_2 \times \dots \times w_D$ , 第一层开关将  $N$  个信号的顺序重新排序以产生第一组重新排序的信号, 第  $i$  层开关将第  $(i-1)$  组的重新排序信号重新排序以产生第  $i$  组的重新排序信号,  $i$  的范围从 2 到  $2D-1$ , 第 25 ( $2D-1$ ) 组的重新排序信号与  $N$  个信号的目标排序相匹配。

29. 如权利要求 28 所述的装置, 其特征在于,  $N$  个信号中的每一个都被分配了  $D$  维坐标, 第  $n$  坐标的范围从 1 到  $w_n$ , 第  $p$  层开关被配置成交换仅第  $p$  个坐标不同而其它维数中的坐标相同的信号,  $p$  的范围是从 1 到  $D$ , 且第  $q$  层开关被配置成交换仅第  $(2D-q)$  个坐标不同而其它维数中的坐标相同的信号,  $q$  的范围从  $D+1$  到  $2D-1$ 。

## 信号置换

## 5 技术领域

本发明涉及信号置换。

## 背景

10 信号置换(signal permuting)用于改变信号的路由通路以连接具有不同顺序的信号线组的两个接口。例如,如图 1A 所示,如果连到输入电路 11 的接口 10 具有承载控制、数据 1、数据 2 和接地信号的线 L0 到 L3,而连到输出电路 13 的第二接口 12 具有承载数据 1、数据 2、接地和控制信号的线 L0 到 L3,则通过 4 乘 4 的置换网络 14 耦合输入和输出接口,该置换网络 14 将 [L0, L1, L2, L3] 映射到 [L3, L0, L1, L2]。

15 可以采用开关实现置换网络。在图 1B 中,一  $n:1$  开关 16 的输出可以反映其  $n$  个输入中的任何一个。通过采用在其输入处接收同组  $n$  个信号的大量  $n:1$  开关,就可以在其输出端子处将  $n$  个信号置换成任意顺序。图 1C 示出一个使用两个  $2:1$  开关 20、22 的  $2$  乘  $2$  的置换网络,它置换两个信号  $S_0$  和  $S_1$  的顺序。

20 可以使用  $n:1$  的多路复用器、写入其内容以反映单个地址输入的  $2^n \times 1$  的随机访问存储器或者被编程为在输出处反映任何一个输入的现场可编程序的门阵列中的  $n$  输入查找表实现一个  $n:1$  开关。对于每一种技术,可以最经济地实现的开关类型可能是不同的。

## 附图概述

25 图 1A-1C 是开关和置换网络的示意图。

图 2-3A 和 3B 是置换网络的示意图。

图 4A 是输入和输出端子之间映射的示图。

图 4B 是示出信号的位置坐标转换的示图。

图 5A 是具有置换网络的计算机的示意图。

30 图 5B 是图 5A 的置换网络的输入和输出端子之间映射的示图。

图 6 是置换网络的示意图。

### 具体描述

本发明针对使用预定组构件块(building block)的置换网络的构建。构件块是一种或多种可编程开关,每一种都能从几个信号中选择一个信号。置换网络被配置成多维开关阵列置换器(MSAP),它可以通过编程开关来置换信号顺序以便根据置换  
5 算法选择信号。

在附图中,每个置换器都在左边具有输入端子并在右边具有输出端子。参考图2,可以使用 $2D-1$ 个层的开关(层 $L_1$ 到 $L_{2D-1}$ )实现可置换 $N=w_1 \times w_2 \times \dots \times w_D$ 个信号的多维开关阵列置换器(MSAP)200。数字 $w_1, w_2, \dots, w_D$ 是整数。每个层包含相同数目(N)的同类开关,但不同层中开关的类型可以不同。对于 $\lambda=1$ 到D,层 $L_\lambda$ 包  
10 括 $w_\lambda:1$ 开关,它们每一个都可以从 $w_\lambda$ 个信号中选择一个信号。例如,层 $L_1$ 包含 $w_1:1$ 开关,层 $L_2$ 包含 $w_2:1$ 开关,层 $L_D$ 包含 $w_D:1$ 开关。对于 $\lambda=D+1$ 到 $2D-1$ ,层 $L_\lambda$ 包括 $(2D-\lambda):1$ 开关。例如,层 $L_{D+1}$ 包括 $w_{D-1}:1$ 开关,而层 $L_{2D+1}$ 包括 $w_1:1$ 开关。通过将一层中的开关输出端子适当地连接到下一层中的开关输入端子,并且适当地将开关编程以便选择输入信号,MSAP200可以将其输入处的一组信号的  
15 顺序置换成其输出处的信号的任意顺序。

层 $L_1$ 中开关的输入信号是MSAP的N个输入信号。层 $L_1$ 中的 $w_1:1$ 开关被配置成形成大量 $w_1$ 乘 $w_1$ 置换器202,数量是 $N/w_1$ 。每个 $w_1$ 乘 $w_1$ 置换器202包括大量 $w_1:1$ 开关,数量是 $w_1$ 。每个 $w_1$ 乘 $w_1$ 置换器202从输入端子204接收 $w_1$ 信号,并在输出端子206上输出一组重新排序的 $w_1$ 信号。置换器202的输出端子206被发送到  
20 层 $L_2$ 中的置换器输入端子。层 $L_2$ 中的 $w_2:1$ 开关被配置成形成大量 $w_2$ 乘 $w_2$ 置换器208,数量是 $N/w_2$ 。每个 $w_2$ 乘 $w_2$ 置换器208在输入端子210处接收 $w_2$ 信号,并在输出端子212处输出一组重新排序的 $w_2$ 信号。置换器208的输出信号被发送到层 $L_3$ 中置换器的输入端子,等等。

通常,层 $L_\lambda$ ( $\lambda=1$ 到D)中的开关被分组成 $w_\lambda$ 乘 $w_\lambda$ 置换器,每个置换器都从前一层接收 $w_\lambda$ 信号并在其输出处产生一组重新排序的 $w_\lambda$ 信号。层 $L_\lambda$ ( $\lambda=D+1$ 到  
25  $2D-1$ )中的开关被分组成 $w_{2D-\lambda}$ 乘 $w_{2D-\lambda}$ 置换器,每个置换器都从前一层接收 $w_{2D-\lambda}$ 信号并在其输出处产生一组重新排序的 $w_{2D-\lambda}$ 信号。最末层 $L_{2D-1}$ 中的开关输出是MSAP的输出,它是一组重新排序的N信号。

一层的输出端子和下一层的输入端子之间的连接是固定的(例如,硬连接的)。该连接可以通过虚线中围绕的区域214表示。配置每个层之间的连接,从而置换器的  
30 每一个输出信号都被发送到下一层中的不同置换器。因此,每个 $w_\lambda$ 乘 $w_\lambda$ 置换器从前一层中的 $w_\lambda$ 不同置换器接收 $w_\lambda$ 输入信号,置换该 $w_\lambda$ 信号的顺序,并将重新

排序的  $w_\lambda$  信号发送到下一层中的  $w_\lambda$  不同置换器。

本发明的优点在于可以根据哪种类型是可行的来使用各种类型的开关设计和构建 MSAP。例如，为了设计可以置换  $N$  个信号的 MSAP，可以首先确定  $N$  的因数  $w_i$ 。通常，可以以许多方法将数量  $N$  化为因数，每一种都对应于不同的设计。例如，假定选择因数  $w_i$  从而  $N=w_1 \times w_2 \times \dots \times w_D$ ，随后，可以设计具有  $D$  维的 MSAP，其中  $w_\lambda$  是第  $\lambda$  维的宽度。通过以下的描述将使“宽度”和“维”的意思变得显而易见。

例如，有几个可以置换 48 个信号的 MSAP 结构：

● 可以通过选择  $N=48$ ， $D=2$ ， $w_1=6$  和  $w_2=8$  来设计二维 MSAP。通过采用 6:1 开关的第一层、8:1 开关的第二层和 6:1 开关的第三层来构成该 48 乘 48 的 MSAP。

● 可以通过选择  $N=48$ ， $D=3$ ， $w_1=3$ 、 $w_2=4$  和  $w_3=4$  来设计三维 MSAP。通过采用 3:1 开关的第一层、4:1 开关的第二、第三、第四层和 3:1 开关的第五层来构成该 48 乘 48 的 MSAP。

可以通过选择  $N=48$ ， $D=5$ ， $w_1=2$ 、 $w_2=2$ 、 $w_3=2$ 、 $w_4=2$  和  $w_5=3$  来设计五维 MSAP。通过采用 2:1 开关的第一到四层、3:1 开关的第五层和 2:1 开关的第六到九层来构成该 48 乘 48 的 MSAP。

当不同环境(例如，用于实现开关的不同技术，或者相同技术的不同商标的产品)提供不同类型的开关，或者允许最经济地实现不同类型的开关时，优点是明显的。例如，在一种环境中，基本构件块可以是 2:1 开关，且所有其它  $n$ :1 开关都可以由 2:1 开关构成。在另一种环境中，基本构件块可以包括 2:1 和 3:1 两种开关，且更大的开关可以由这两种类型的开关构成。可以使用对特定环境最经济的构件块来构成  $N$  乘  $N$  的 MSAP。

接着之前的实例，为了置换 48 个信号，还可能选择 MSAP 的结构，其中  $N$  被选为大于 48，从而某些 MSAP 输入和输出是不使用的。该结构使用更多逻辑电路，但可以使特定 MSAP 结构可得。例如，通过选择  $N=64$ ，以下的三种配置可以用于构成 64 乘 64 的 MSAP：

●  $N=64$ ， $D=2$ ， $w_1=8$ ， $w_2=8$ 。

●  $N=64$ ， $D=3$ ， $w_1=4$ ， $w_2=4$ ， $w_3=4$ 。

●  $N=64$ ， $D=6$ ， $w_1=2$ ， $w_2=2$ ， $w_3=2$ ， $w_4=2$ ， $w_5=2$ ， $w_6=2$ 。

通常，对于特殊环境，如果必须从更小的开关构建更大的开关，则为了置换相同数量( $N$ )的信号，使用较小开关的较高维 MSAP 将使用更少的逻辑电路。在上述实例中，如果必须通过 2:1 开关构建 8:1 和 4:1 的开关，则在这三种配置中，使

用 2: 1 开关的六维 MSAP 是更加经济的(即, 将使用最少数目的全部开关)。因为维宽度  $w_k$  与开关的大小成比例, 通过将  $w_1, w_2, \dots, w_D$  选为数值上尽可能接近将可以使用较小的开关和较少的逻辑电路。例如, 当  $N=n^D$  时, 最经济的配置是选择  $w_1 = w_2 = \dots = w_D = n$ 。例如, 如果必须通过 2: 1 的开关构成更大的开关, 则 MSAP 的最有效设计是选择  $n=2$  和使用 2: 1 的开关作为 MSAP 的基本构件块。参见以下的表 1 以便比较用各种大小的开关构成 MSAP 的有效性。

在图 3A 所示的实例中, 12 乘 12 的 MSAP 300 由两层的 3: 1 开关和一层的 4: 1 开关构成。MSAP 300 包括层  $L_1$  到  $L_3$ 。层  $L_1$  包括 3: 1 的开关, 它被配置成形成 3 乘 3 的置换器 302、304、306 和 308。层  $L_2$  包括 4: 1 开关, 它被配置成构成 4 乘 4 置换器 310、312 和 314。层  $L_3$  包括 3: 1 开关, 它被配置成构成 3 乘 3 置换器 316、318、320 和 322。在每个置换器内, 不同的开关选择不同的输入信号, 从而在它们的选择中没有重叠, 导致置换器输出端子处出现输入信号的重新排序。

层  $L_1$  中的每个 3 乘 3 置换器在其输入端子处接收三个信号, 置换这三个信号的顺序, 并将重新排序的三个信号发送到层  $L_2$  中的三个不同的置换器。例如, 置换器 302 的第一输出端子连接到置换器 310 的第一输入端子。置换器 302 的第二输出连接到置换器 312 的第一输入。置换器 302 的第三输出连接到置换器 314 的第一输入, 等等。在这种方式中, 层  $L_1$  中置换器的三个输出信号中的每一个都将由层  $L_2$  中的不同置换器重新排序。

同样, 层  $L_2$  中置换器的每个输出都连接到层  $L_3$  中不同置换器的输入。因此, 层  $L_2$  中置换器的四个输出信号中的每一个都将由层  $L_3$  中的不同置换器重新排序。通过用每个置换器将信号合适地重新排序, 12 乘 12 的 MSAP 可以将 12 个输入信号重新排序成任一顺序的 12 个输出信号。

作为置换器如何将信号重新排序的说明, 在每个置换器内示出虚线, 表示从输入端子到输出端子的信号路由。具有顺序  $[S_1, S_2, \dots, S_{12}]$  的 12 个输入信号在 MSAP 300 的输出处被重新排序成顺序  $[S_7, S_5, S_3, S_9, S_8, S_1, S_{10}, S_2, S_6, S_{12}, S_{11}, S_4]$ 。以下说明信号  $S_3$  如何从图左边的输入端子处的线#3 被重新排序成图右边的输出端子处的线#6。置换器 302 将信号  $S_3$  从输入#3 重新排序到输出#1, 该输出#1 连接到置换器 310 的输入#1。置换器 310 将信号  $S_3$  从输入#1 重新排序到输出#2, 该输出#2 连接到置换器 318 的输入#4。随后, 置换器 318 将信号  $S_3$  从输入#4 重新排序到输出#6。图中从上到下将每一层中置换器的输入和输出编号为#1 到#12。

以下描述一个层中开关的输出端子和下一个层中开关的输入端子之间的连接。这些连接是固定的(例如, 硬连接的)。为了描述的目的, 假定 MSAP 每一层中的开



关形成  $D$  维阵列。层  $L_\lambda$  中的开关被表示成  $S[\lambda][x_1, x_2, \dots, x_D]$ , 其中  $1 \leq x_k \leq w_k$ ,  $k$  从 1 到  $D$ 。数字  $x_1$  表示开关的第一维坐标, 数字  $x_2$  表示第二维坐标, 等等。通过在开关名称的末端添加下标来表示这些开关的输入。例如, 开关  $S[1][2, 3, 4]$  具有输入  $S[1][2, 3, 4]_1, S[1][2, 3, 4]_2$  等等。通过添加星号表示输出(例如  $S[1][2, 3, 4]^*$ )。

MSAP 的输出信号是最末层  $L_{2D-1}$  的输出, 即对于  $x_1, x_2, \dots, x_D$  的各种组合, MSAP 输出是  $S[2D-1][x_1, x_2, \dots, x_D]^*$ 。为了简化开关符号的描述, 使用虚构层  $L_0$ , 从而 MSAP 输入是  $L_0$  输出。因此, 对于  $x_1, x_2, \dots, x_D$  的各种组合, MSAP 输入是  $S[0][x_1, x_2, \dots, x_D]^*$ 。虽然不存在开关  $S[0][1, \dots, 1]$ , 但有信号  $S[0][1, \dots, 1]^*$ , 它是 MSAP 输入。

对于  $\lambda = 1, 2, \dots, D$ , 层  $L_\lambda$  中置换器仅沿着维数  $\lambda$  任意地置换前一层中置换器的输出。这意味着层  $L_\lambda$  中置换器中的开关仅从具有不同维数  $\lambda$  的标记的前一层中的开关接收信号, 并仅置换这些信号。层  $L_\lambda$  中 ( $\lambda \leq D$ ) 每个  $w_\lambda: 1$  开关, 表示成  $S[\lambda][x_1, x_2, \dots, x_D]$ , 具有  $w_\lambda$  输入, 即  $S[\lambda][x_1, x_2, \dots, x_D]_1, S[\lambda][x_1, x_2, \dots, x_D]_2, \dots, S[\lambda][x_1, x_2, \dots, x_D](w_\lambda)$ 。这些输入映射到前一层的所有输出, 其中开关坐标仅有维数  $\lambda$  不同:

$$\begin{aligned} S[\lambda][x_1, x_2, \dots, x_\lambda, \dots, x_D]_1 &= S[\lambda-1][x_1, x_2, \dots, 1, \dots, x_D]^* \\ S[\lambda][x_1, x_2, \dots, x_\lambda, \dots, x_D]_2 &= S[\lambda-1][x_1, x_2, \dots, 2, \dots, x_D]^* \\ &\dots \\ S[\lambda][x_1, x_2, \dots, x_\lambda, \dots, x_D](w_\lambda) &= S[\lambda-1][x_1, x_2, \dots, w_\lambda, \dots, x_D]^* \end{aligned}$$

( $1 \leq \lambda \leq D$ ; 任何  $x_1, x_2, \dots, x_D$ )

对于  $\lambda = D+1, D+2, \dots, 2D-1$ , 层  $L_\lambda$  仅沿维数  $2D-\lambda$  任意置换前一层输出。这意味着层  $L_\lambda$  中置换器中的开关仅从具有不同维数  $2D-\lambda$  的标记的前一层中的开关接收信号, 并仅置换这些信号。层  $L_\lambda$  中 ( $\lambda > D$ ) 每个  $w_{2D-\lambda}: 1$  开关, 表示成  $S[\lambda][x_1, x_2, \dots, x_D]$ , 具有  $w_{2D-\lambda}$  输入, 即  $S[\lambda][x_1, x_2, \dots, x_D]_1, S[\lambda][x_1, x_2, \dots, x_D]_2, \dots, S[\lambda][x_1, x_2, \dots, x_D](w_{2D-\lambda})$ 。这些输入映射到前一层的所有输出, 其中开关坐标仅有维数  $2D-\lambda$  不同:

$$\begin{aligned} S[\lambda][x_1, x_2, \dots, x_{2D-\lambda}, \dots, x_D]_1 &= S[\lambda-1][x_1, x_2, \dots, 1, \dots, x_D]^* \\ S[\lambda][x_1, x_2, \dots, x_{2D-\lambda}, \dots, x_D]_2 &= S[\lambda-1][x_1, x_2, \dots, 2, \dots, x_D]^* \\ &\dots \\ S[\lambda][x_1, x_2, \dots, x_{2D-\lambda}, \dots, x_D](w_{2D-\lambda}) &= S[\lambda-1][x_1, x_2, \dots, w_{2D-\lambda}, \dots, x_D]^* \end{aligned}$$

( $D+1 \leq \lambda \leq 2D-1$ ; 任何  $x_1, x_2, \dots, x_D$ )

参考图 3B, MSAP 300 中的开关被标为  $S[\lambda][x_1, x_2]$ ,  $\lambda$  从 1 到 3,  $x_1$  从 1 到 3 且  $x_2$  从 1 到 4。为了符号的方便, MSAP 输入信号被描述成层  $L_0$  的输出信号, 并标记成  $S[0][i, j]^*$ ,  $i$  从 1 到 3 而  $j$  从 1 到 4。层  $L_1$  沿第一维数置换信号。这意味着层  $L_0$  中开关的输出连接到层  $L_1$  中开关的输入, 该层  $L_0$  中开关具有第一维数中的不同坐标而第二维数中相同坐标的标记。例如, 开关  $S[1][2, 3]$  的三个输入连接到层  $L_0$  中开关的输出, 具体如下:

$$S[1][2, 3]_1 = S[0][1, 3]^*$$

$$S[1][2, 3]_2 = S[0][2, 3]^*$$

$$10 \quad S[1][2, 3]_3 = S[0][3, 3]^*$$

层  $L_2$  沿第二维数置换信号。这意味着层  $L_1$  中开关的输出连接到层  $L_2$  中开关的输入, 该层  $L_1$  中开关具有第二维数中的不同坐标而第一维数中相同坐标的标记。例如, 开关  $S[2][2, 3]$  的四个输入如下连接:

$$S[2][2, 3]_1 = S[1][2, 1]^*$$

$$15 \quad S[2][2, 3]_2 = S[1][2, 2]^*$$

$$S[2][2, 3]_3 = S[1][2, 3]^*$$

$$S[2][2, 3]_4 = S[1][2, 4]^*$$

层  $L_3$  沿第一维数置换信号。例如, 开关  $S[3][2, 3]$  的三个输入如下连接:

$$S[3][2, 3]_1 = S[2][1, 3]^*$$

$$20 \quad S[3][2, 3]_2 = S[2][2, 3]^*$$

$$S[3][2, 3]_3 = S[2][3, 3]^*$$

层 MSAP 300 的输出信号是来自层  $L_3$  的 12 个开关输出, 即  $S[3][1, 1]^*$  到  $S[3][3, 4]^*$ 。

MSAP 开关拓扑具有递归结构。如果从维数  $D > 1$  的 MSAP 中除去开关的第一和最末层 ( $L_1$  和  $L_{2D-1}$ ), 则剩余部分是大量维数  $D-1$  的更小 MSAP, 数量是  $w_1$ 。通常, 在维数  $D > 1$  的 MSAP 中, 对于第一维中 1 和  $w_1$  之间的每一个固定坐标  $c$ , 开关的子集  $\{S[\lambda][c, x_2, x_3, \dots, x_D]\}$  ( $\lambda$  从 2 到  $2D-2$  且每个  $x_k$  从 1 到  $w_k$ ) 形成维数  $D-1$  的子 MSAP, 其维宽度是  $w_2, w_3, \dots, w_D$ 。每个子 MSAP 都具有其自己的子 MSAP 等等, 直到一维 MSAP 的水平。在以下描述的 MSAP 切换算法中反映了该 MSAP 的递归结构。

30 根据切换算法配置 MSAP 中的开关, 从而 MSAP 可以以任何选择的顺序置换信号。为了便于描述该算法, MSAP 的每个输入信号都被描述成具有起始位置和目标位置。特殊层中信号的位置写作  $[x_1, x_2, \dots, x_D]$  并对应于开关的标记, 它选择信号作为

输出。例如，图 3B 中，输入线#3 处的信号具有[3, 1]的起始位置，因为该信号由开关  $S[0][3, 1]$  选择并在其输出处反映。如前所述，层  $L_0$  是虚构层并仅为了描述方便。当信号传播到层  $L_1$  时，其位置变成[1, 1]因为该信号由开关  $S[1][1, 1]$  选择。当信号传播到层  $L_2$  时，其位置变成[1, 2]因为该信号由开关  $S[2][1, 2]$  选择。

5 随后在层  $L_3$  中，信号的位置变成[3, 2]因为该信号由开关  $S[3][3, 2]$  选择。

在三维 ( $D=3$ ) MSAP 中，假定输入信号具有起始位置[2, 5, 3]和目标位置[4, 1, 6]。这意味着必须以一定的方式设置 MSAP 开关从而输入信号  $S[0][2, 5, 3]^*$  被路由到输出信号  $S[5][4, 1, 6]^*$ 。当信号通过 MSAP 传播时，在每个开关层中调整信号位置，反映信号传播经过的开关。因为每层仅在一个维数中置换信号，所以在一层中仅调整信号位置的一个坐标。例如，在  $D=3$  的 MSAP 中，信号可以遵循以下位置路径：

$L_0$ : [2, 5, 3]  
 $L_1$ : [8, 5, 3]  
 $L_2$ : [8, 7, 3]  
 15  $L_3$ : [8, 7, 6]  
 $L_4$ : [8, 1, 6]  
 $L_5$ : [4, 1, 6]

该位置路径确定如何设置每个层中的一个开关。为使信号从  $L_0$  位置[2, 5, 3]传播到  $L_1$  位置[8, 5, 3]，开关  $S[1][8, 5, 3]$  必须被设定为位置 2。为使信号从  $L_1$  位置[8, 5, 3]传播到  $L_2$  位置[8, 7, 3]，开关  $S[2][8, 7, 3]$  必须被设定为位置 5，等等。当用于 MSAP 输入信号的所有位置路径都确定下来时，就可以设置 MSAP 中的所有开关，从而所有的输入信号都被路由到 MSAP 的合适输出端子。因此， $N$  位置路径(或信号路径)的集合可以用于确定 MSAP 的开关设置配置。

### 25 MSAP 切换算法

MSAP 切换算法提供了一种配置 MSAP 以便在输出端子处实现所有可能的信号置换的方法。但是，切换算法提供的配置不是仅有的配置 MSAP 以置换信号的方法。还可以使用“强力(brute-force)”方法，通过以某些智能(例如，递归)检查所有可能的切换组合以找到解决方案。

30 MSAP 切换算法包括算法 1，它调用算法 2。在以下算法 1 的描述中，在信号位置路径而非 MSAP 切换设置的方面描述 MSAP 切换程序。有  $N$  个信号，每个信号都具有  $D$  维起始位置和  $D$  维目标位置。算法 1 为所有信号确定位置路径，以 MSAP 拓扑

允许的方式给出每个信号在从  $L_0$  到  $L_{2D-1}$  每层中的位置坐标。两个信号不能占据任一层中的相同位置。每个信号的  $L_0$  位置是其起始位置。通过算法 1 生成的每个信号的  $L_{2D-1}$  位置将与信号的目标位置相匹配。

### 5      算法 1

算法 1 递归地调用其自身。每次调用算法 1 时，表示输入信号子集的参数  $\Sigma$  就被传递给算法 1。当初始调用算法 1 时，参数  $\Sigma = \{\text{所有信号}\}$  被传递给算法 1。在随后的递归调用中，参数  $\Sigma$  将包含信号的连续的较小子集。算法 1 通过传递水平参数  $\lambda$  保持跟踪其递归的深度。当初始调用算法 1 时，将参数  $\lambda = 1$  提供给该算法。

10     当算法 1 再次调用其自身时，传递参数  $\lambda = 2$ ，等等。

对于任何给定的  $N, D, w_1, w_2, \dots,$  和  $w_D$  从而  $N = w_1 \times w_2 \times \dots \times w_D$ ，算法 1 用参数  $\lambda$  表示递归深度和用  $\Sigma$  表示信号的子集来执行以下步骤从而进行操作。开始，设置  $\lambda = 1$ 。

步骤 1: 如果  $\lambda = D$ ，跳到步骤 5。

15     步骤 2: 将  $\Sigma$  中每个信号的  $L_{\lambda-1}$  位置复制到其  $L_\lambda$  位置。

步骤 3: 调用算法 2 来置换  $\Sigma$  中信号的  $L_\lambda$  位置，仅改变每个信号的第  $\lambda$  位置坐标，其中具有相同第  $\lambda$  位置坐标的  $\Sigma$  中的两个信号不能具有相同的第  $(\lambda + 1)$  至第  $D$  目标坐标。

步骤 4: 对于每个  $c$ ，范围从 1 到  $w_\lambda$ ，递归地调用算法 1，传递参数：

20      $\Sigma = \Sigma$  中信号的子集，它具有等于层  $L_\lambda$  中的  $c$  的第  $\lambda$  位置坐标；以及  
 $\lambda = \lambda + 1$ 。

步骤 5: 通过将每个信号的第  $\lambda$  位置坐标变成其第  $\lambda$  目标坐标，将  $\Sigma$  中的信号从其  $L_{2D-\lambda-1}$  中的位置置换成  $L_{2D-\lambda}$  中的新位置。

25     算法 1 的步骤 3 和 5 中执行的置换仅沿该层允许的特定维数置换信号。例如，层  $L_k$  中的开关仅沿维数  $k$  置换信号。算法 1 还遵循两个信号不能占据给定层中同一位置的规则。可以确保该规则，因为如下所述，算法 2 通过交换信号对移动信号从而防止了信号的冲突(即，两个信号由同一开关选择)。

### 30      算法 2

以下是算法 2 的描述，它由算法 1 的步骤 3 调用。由于算法 1 的递归性质，在层  $L_\lambda$  中， $\Sigma$  中的所有信号都将具有相同的第 1 到第  $(\lambda - 1)$  位置坐标。另一方面，它们的第  $\lambda$  到第  $D$  位置坐标将经过所有可能的组合。因此，虽然每个信号仍具有  $D$

位置坐标，但 $\Sigma$ 实际上是根据层 $L_\lambda$ 中位置的信号的 $(D-\lambda+1)$ 维阵列。 $\Sigma$ 中每个信号的第 $\lambda$ 位置坐标确定信号所在的阵列的“片(slice)”。每个信号的第 $(\lambda+1)$ 到第 $D$ 位置坐标确定信号所在的阵列的“列(column)”。“片”和“列”的定义在下面给出。具有相同第 $\lambda$ 位置坐标的两个信号处于相同片中，且具有相同第 $(\lambda+1)$ 到第 $D$ 位置坐标的两个信号处于相同列中。因此， $\Sigma$ 中的信号阵列被分成 $w_\lambda$ 个不同片，从1标号到 $w_\lambda$ 。该阵列还被分成 $w_{\lambda+1} \times w_{\lambda+2} \times \dots \times w_D$ 个不同列，不加以标号。算法1的步骤3仅改变每个信号的第 $\lambda$ 位置坐标，从而信号仅在其列中移动。

以下是术语“片”和“列”的描述。片是包含由算法2处理的信号 $\Sigma$ 子集的可得位置坐标的子集。例如，考虑 $D=5$ ， $w_1=6$ ， $w_2=5$ ， $w_3=4$ ， $w_4=3$ 和 $w_5=2$ 的WSAP。在这种情况下，总共 $6 \times 5 \times 4 \times 3 \times 2 = 720$ 个信号被置换，每个都具有5维的起始位置坐标和5维的目标位置坐标。例如，一个位置 $[1, 1, 1, 1, 1]$ 中开始的信号可以具有目标位置 $[6, 5, 4, 3, 2]$ 。假定当 $\lambda=3$ ，这表示算法1中递归的第三级时，调用算法2。在这种情况下， $\Sigma$ 是具有维数1和2中恒定 $L_2$ 位置坐标的信号的子集。例如， $\Sigma$ 中所有的信号都具有 $L_2$ 位置坐标，其形式是 $[4, 2, \_, \_, \_]$ ，从而 $\Sigma$ 正好包含 $4 \times 3 \times 2 = 24$ 个信号： $[4, 2, 1, 1, 1]$ ， $\dots$ ， $[4, 2, 4, 3, 2]$ 。

在算法1的步骤2中， $\Sigma$ 中信号的 $L_2$ 位置被复制到它们的 $L_3$ 位置，从而这24个信号的 $L_3$ 位置也是 $[4, 2, \_, \_, \_]$ 的形式。调用算法2以便重新排列 $L_3$ 位置内 $\Sigma$ 中的24个信号，即算法2改变 $\Sigma$ 中24个信号的 $L_3$ 位置坐标。算法2根据其位置坐标将 $\Sigma$ 中的24个信号分层4“片”和6“列”。4个片中的每一个都有6个信号，且6个列中的每一个都有4个信号。

4个片都包含具有给定的第三(由于 $\lambda=3$ ) $L_3$ 位置坐标的那些信号。存在4个片，因为 $\lambda=3$ 且 $w_3=4$ 。这些片被称作“片1”到“片4”。片 $k$ 包含 $\Sigma$ 中第三 $L_3$ 位置坐标是 $k$ 的那些信号。例如，片3包含6个信号，它们的 $L_3$ 位置坐标是 $[4, 2, 3, 1, 1]$ ， $[4, 2, 3, 1, 2]$ ， $[4, 2, 3, 2, 1]$ ， $[4, 2, 3, 2, 2]$ ， $[4, 2, 3, 3, 1]$ ，和 $[4, 2, 3, 3, 2]$ 。在调用算法2期间，片3中的信号具有这些六个 $L_3$ 位置，虽然占据这些位置的特定信号可以改变。

当算法2交换两个信号时，这两个信号将交换片，这意味着它们将交换它们的第三 $L_3$ 位置坐标。同时，6个列包含具有第四和第五 $L_3$ 位置坐标( $\lambda=3$ 以上的坐标)的给定组合的这些信号。因此，存在一列，其信号具有 $L_3$ 位置坐标 $[4, 2, 1, 1, 1]$ ， $[4, 2, 2, 1, 1]$ ， $[4, 2, 3, 1, 1]$ 和 $[4, 2, 4, 1, 1]$ 。存在另一个列，其信号具有 $L_3$ 位置坐标 $[4, 2, 1, 3, 2]$ ， $[4, 2, 2, 3, 2]$ ， $[4, 2, 3, 3, 2]$ 和 $[4, 2, 4, 3, 2]$ 。该实例中使用的术语“列”表示一组 $L_3$ 位置坐标，其第四和第

五维坐标是固定的。因此，具有  $L_3$  位置坐标  $[4, 2, 2, 3, 1]$  和  $[4, 2, 4, 3, 1]$  的信号被称为是“同一列”，因为它们的第四和第五坐标  $[3, 1]$  相匹配。在算法 2 中，列总是在置换后保持同组信号，因为交换操作仅交换同一列中的两个信号。

已经描述了术语“片”和“列”的定义，以下是算法 2 执行的步骤的描述：

1. FOR  $S = 1$  TO  $w_\lambda - 1$  DO:
  2.     WHILE slice  $S$  contains at least 2 signals with identical  $(\lambda+1)^{\text{th}}$  to  $D^{\text{th}}$  target coordinates DO:
  3.         SET  $X_0 =$  one of these signals
  4.         FOR  $K = 0, 1, 2, \dots$  DO:
  5.             Look for a signal  $Z$  in slices  $S+1$  through  $w_\lambda$  and in the same column as  $X_K$  (selected in Step 3 or 6) with the following property:  $Z$ 's  $(\lambda+1)^{\text{th}}$  to  $D^{\text{th}}$  target coordinates are not identical to the  $(\lambda+1)^{\text{th}}$  to  $D^{\text{th}}$  target coordinates of any signal in slice  $S$ . IF found THEN BREAK from the “FOR  $K$ ” loop.
  6.             Select a pair of signals signal  $Y_{K+1}$  and  $X_{K+1}$  with the following properties:  
 $Y_{K+1}$  is in slices  $S+1$  through  $w_\lambda$  and in the same column as  $X_0, X_1, \dots, \text{ or } X_K$ ;  
 $X_{K+1}$  is in slice  $S$ , and is different from  $X_0, X_1, \dots, \text{ and } X_K$ ;  
 $Y_{K+1}$  and  $X_{K+1}$  have identical  $(\lambda+1)^{\text{th}}$  to  $D^{\text{th}}$  target coordinates.  
There will always be such a pair of signals.
  7.             END FOR  $K$
  8.             LOOP:
  9.             Swap signal  $Z$  with signal  $X_K$
  10.            IF  $K = 0$  THEN BREAK from the LOOP
  11.            SET  $Z = Y_K$
  12.             $Z$  is now in the same column as some  $X_j$ . SET  $K$  to this  $J$ .
  13.            END LOOP
  14.     END WHILE
  15. END FOR  $S$
- (
1. FOR  $S=1$  TO  $w_\lambda - 1$  DO:
  2. WHILE 片  $S$  包含至少具有相同的第  $(\lambda + 1)$  至第  $D$  目标坐标 DO:
  - 10 3. SET  $X_0 =$  这些信号之一
  4. FOR  $K=0, 1, 2, \dots$  DO:
  5.     在片  $S+1$  至  $w_\lambda$  并且在与  $x_k$  相同的列中 (在步骤 3 或 6 中选择的) 查找具有

下列属性的信号  $Z$ :  $Z$  的第  $(\lambda + 1)$  至第  $D$  目标坐标与在片  $S$  中任何信号的第  $(\lambda + 1)$  至第  $D$  目标坐标相同. 如果找到则从"FOR  $K$ "循环中止.

6.       选择具有下列属性的一对信号  $Y_{K+1}$  和  $X_{K+1}$ :

$Y_{K+1}$  是在片  $S+1$  至  $w_{\lambda}$  并且在与  $X_0, X_1, \dots,$  或  $X_K$  相同的列中;

5        $X_{K+1}$  是在片  $S$  中, 并且不同于  $X_0, X_1, \dots,$  或  $X_K$ ;

$Y_{K+1}$  和  $X_{K+1}$  具有相同的第  $(\lambda + 1)$  至第  $D$  目标坐标.

将总是存在这样的一对信号.

7.       END FOR  $K$

8.       LOOP:

10      9.       将信号  $Z$  与信号  $X_K$  交换

11      10.      IF  $K=0$  THEN BREAK 从 LOOP

12      11.      SET  $Z=Y_K$

13      12.       $Z$  现在在与  $X_j$  相同的列中. 设置  $K$  为这个  $J$ .

14      13.      END LOOP

15      14.      END WHILE

15      15.      END FOR  $S$

)

通过每次“固定”一个片来进行算法 2, 即一旦固定一片, 在连续调用该算法时不改变其信号. 步骤 2 的 WHILE 条件检查当前片的属性, 该属性是算法 2 试图除去  
20 去的. 当不再发现该属性时, 就固定了该片. 经过步骤 3-13 的相当少的次数总是固定了一片。

步骤 3-13 的目的在于将具有“重复的”第  $(\lambda + 1)$  到第  $D$  目标坐标的信号移出层  $S$ , 用来自在层  $S$  上的某些其它信号代替之, 在层  $S$  中缺少其中所述的某些其它信号的第  $(\lambda + 1)$  到第  $D$  目标坐标. 在这两个信号处于同一列中的理想情况下, 算法  
25 2 仅交换这两个信号. 但是, 这对信号可以不在同一列中, 从而执行一交换链, 其中每一个都在不同的列中进行操作. 第一个交换操作将缺少第  $(\lambda + 1)$  到第  $D$  目标坐标的信号移动入层  $S$ , 而最末的交换操作将具有重复的  $(\lambda + 1)$  到第  $D$  目标坐标的信号移出层  $S$ 。

图 4A 示出 MSAP 300 的输入和输出端子之间映射的实例. 图 4B 示出 12 乘 12 MSAP  
30 的输入信号的位置坐标如何从其开始位置坐标变成其目标位置坐标的实例. 用以上的算法 1 和算法 2 确定位置坐标的变化. 层  $L1$  仅改变第一维的坐标, 层  $L2$  仅改变第二维的坐标, 而层  $L3$  仅改变第一维的坐标。

MSAP 的优点在于, 它有效地使用预先选择的开关作为基本构件块. 最简单且最有效的 MSAP 仅使用  $n:1$  开关 ( $n$  是整数) 来置换  $N=n^D$  个信号. 在这种情况下, 有  
35  $2D-1$  层, 每层都包含  $N$  个开关. 由于  $D=\log_n N$ , 总共有  $N \times (2\log_n N - 1)$  个  $n:1$  开关. 因此, 用预先选择的开关构建 MSAP 所需的逻辑电路量约为  $N \log N$ , 其中该预

先选择的开关具有固定的开关尺寸。

以下在表 1 中给出了使用各种尺寸的开关构建的 MSAP 的效率比较。这里，假定较大开关是由较小开关构成的。表 1 示出，使用较大开关尺寸的 MSAP 需要更多的逻辑电路。例如，如果 2:1 开关是可得的，则用 11 个 2:1 开关的 6 维层构建 N=64 的 MSAP 是最有效的。使用 5 个 4:1 开关的 3 维层多需要 36% 的逻辑电路。使用 3 个 8:1 开关的 2 维层多需要 91% 的逻辑电路。使用 1 个 64:1 开关的 1 维层多需要 473% 的逻辑电路。

表 1

N	n:1 开关 每层	层	总的 n:1 开关	组成每个 n:1 开关的 2: 1 开关(=n-1)	等效的总 2:1 开关	与 n=2 相对 的比率	与 n=2 相对 的差
2	64	11	704	1	704	100%	0%
4	64	5	320	3	960	136%	36%
8	64	3	192	7	1344	191%	91%
64	64	1	64	63	4032	573%	473%

MSAP 的应用是在计算机的主板和子板之间置换信号。参考图 5A，计算机 500 包括经由具有信号线#1 到#12 的接口 504 耦合到主板 502 的子板 506。子板 506 包括存储器 512，它可以由主板 502 访问。子板 506 和主板 502 由不同的公司制造，从而主板 502 在接口 504 的信号线上以与存储器 512 识别的顺序不同的顺序发送信号。

子板 506 包括置换网络 508，它具有连接到接口 504 的输入端子和连接到接口 510 的输出端子，它顺次连接到存储器 512。置换网络 508 将接口 504 的信号线中的信号路由到接口 510 的信号线，它具有存储器 512 可接受的新顺序。图 5B 示出接口 504 的信号线和接口 510 的信号线之间的映射。作为实例，接口 504 的线#1 上的信号 s1 映射到接口 510 的线#7，而接口 504 的线#12 上的信号 s12 映射到接口 510 的线#4。同样，从存储器 512 发送的信号由置换网络 508 重新排序，随后传递到主板 502 上。

使用现场可编程序的逻辑阵列(FPGA)实现置换网络 508。FPGA 包括可配置的逻辑部件，它们每一个都可以被编程以便接收大量输入并将任何一个输入传递到其输出，就象一个 n:1 开关。当首先引导计算机 500 时，子板 506 与主板 502 通信以便确定所需的置换。子板 506 上的处理器(未示出)执行 MSAP 切换算法以便将可配置逻辑部件编程从而构建实现所需置换的 MSAP。

在图 6 所示的实例中，18 乘 18 的 MSAP 600 由 4 层的 3:1 开关和 1 层 2:1 开



关构成。在该实例中， $N=18=3\times 3\times 2$ ， $D=3$ ， $w_1=3$ ， $w_2=3$ ，且  $w_3=3$ 。MSAP 600 包括层  $L_1$  到  $L_5$ 。层  $L_1$ ， $L_2$ ， $L_4$ ，和  $L_5$  包括被配置成形成 3 乘 3 置换器的 3: 1 开关。层  $L_3$  包括被配置成形成 2 乘 2 置换器的 2: 1 开关。在每个置换器内，不同的开关选择不同的输入信号，从而在它们的选择中没有重叠。这使得在置换器的输出端子处呈现重新排序的输入信号。

图 6 中，在置换器内示出开关标记。层  $L_1$  的置换器 602 中的三个开关具有相同的第二和第三坐标和不同的第一坐标。因此，置换器 602 在第一维中置换输入信号的位置坐标。同样，层  $L_1$  中的其它置换器也在第一维中置换输入信号的位置坐标。层  $L_2$  的置换器 604 中的三个开关具有相同的第一和第三坐标和不同的第二坐标。因此，置换器 604 在第二维中置换输入信号的位置坐标。同样，层  $L_2$  中的其它置换器也在第二维中置换输入信号的位置坐标。

以类似的方式，层  $L_3$  中的置换器在第三维中置换输入信号的位置坐标。层  $L_4$  中的置换器在第二维中置换输入信号的位置坐标。层  $L_5$  中的置换器在第一维中置换输入信号的位置坐标。层  $L_1$  中的置换器的每个输出信号都被发送到层  $L_2$  中的不同置换器，层  $L_2$  中的置换器的每个输出信号都被发送到层  $L_3$  中的不同置换器，等等。

MSAP 600 被设计成允许 18 个输入信号中的任何一个被路由到 18 个输出端子中的任何一个。在操作中，执行 MSAP 切换算法以便为 18 个输入信号确定转换路径。该转换路径经过 5 层开关，从而每个信号被路由到所需的输出端子且两个信号不会占据同一开关。随后，将开关编程以便根据 MSAP 切换算法确定的信号路径选择输入信号。

其它实施例也在以下权利要求书的范围之内。例如，可以使用 FPGA 以外的装置实现开关。可以通过主板上的处理器而非通过子板上的处理器将开关编程。开关的配置（即，从其输入选择哪个信号和在输出处反映哪个信号）可以是动态可编程的，或者是可编程一次随后是永久固定的。执行位置置换的 MSAP 可以被用作编码/解码消息的编码器/解码器的构件块。MSAP 可以用来构建远程通信网络以便以非块的方式将来自一个位置处的节点的信号路由到其它位置的节点。MSAP 还可以用于大量并行或超级计算应用中以在不同的处理器间路由信号。

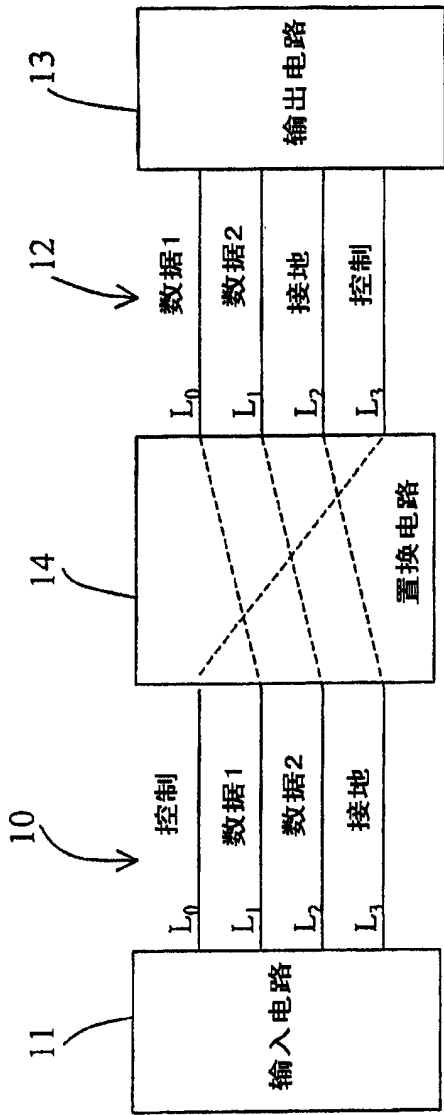


图 1A

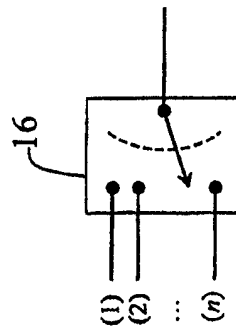


图 1B

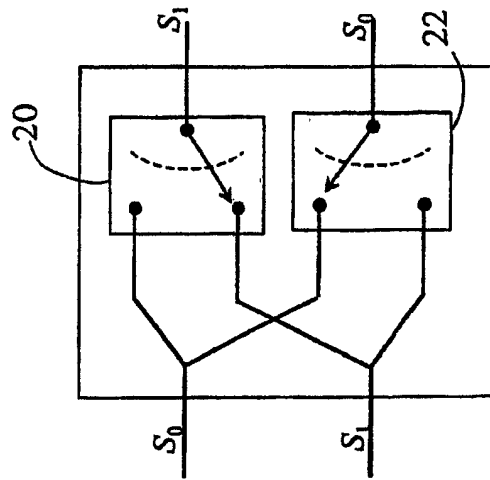
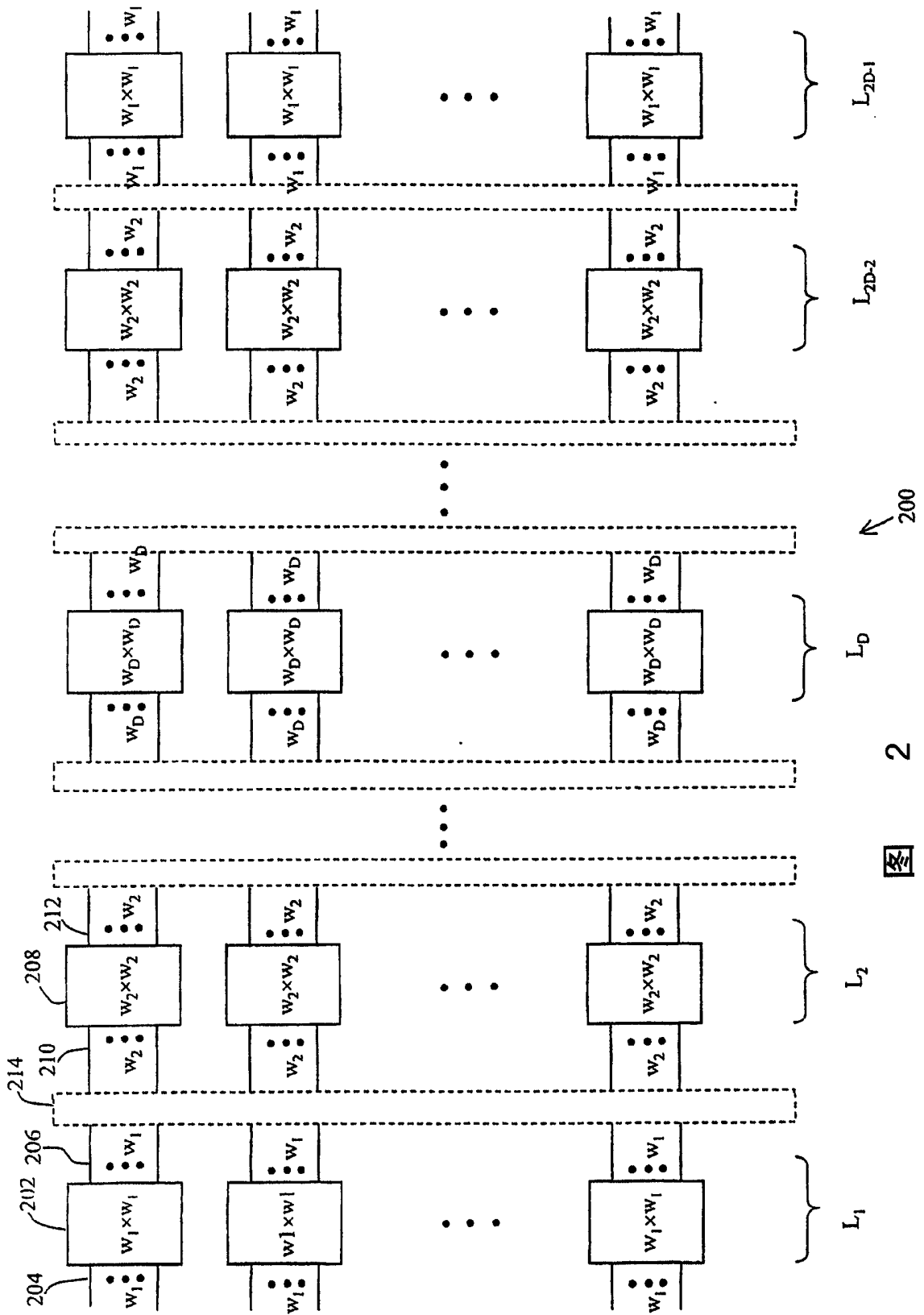


图 1C



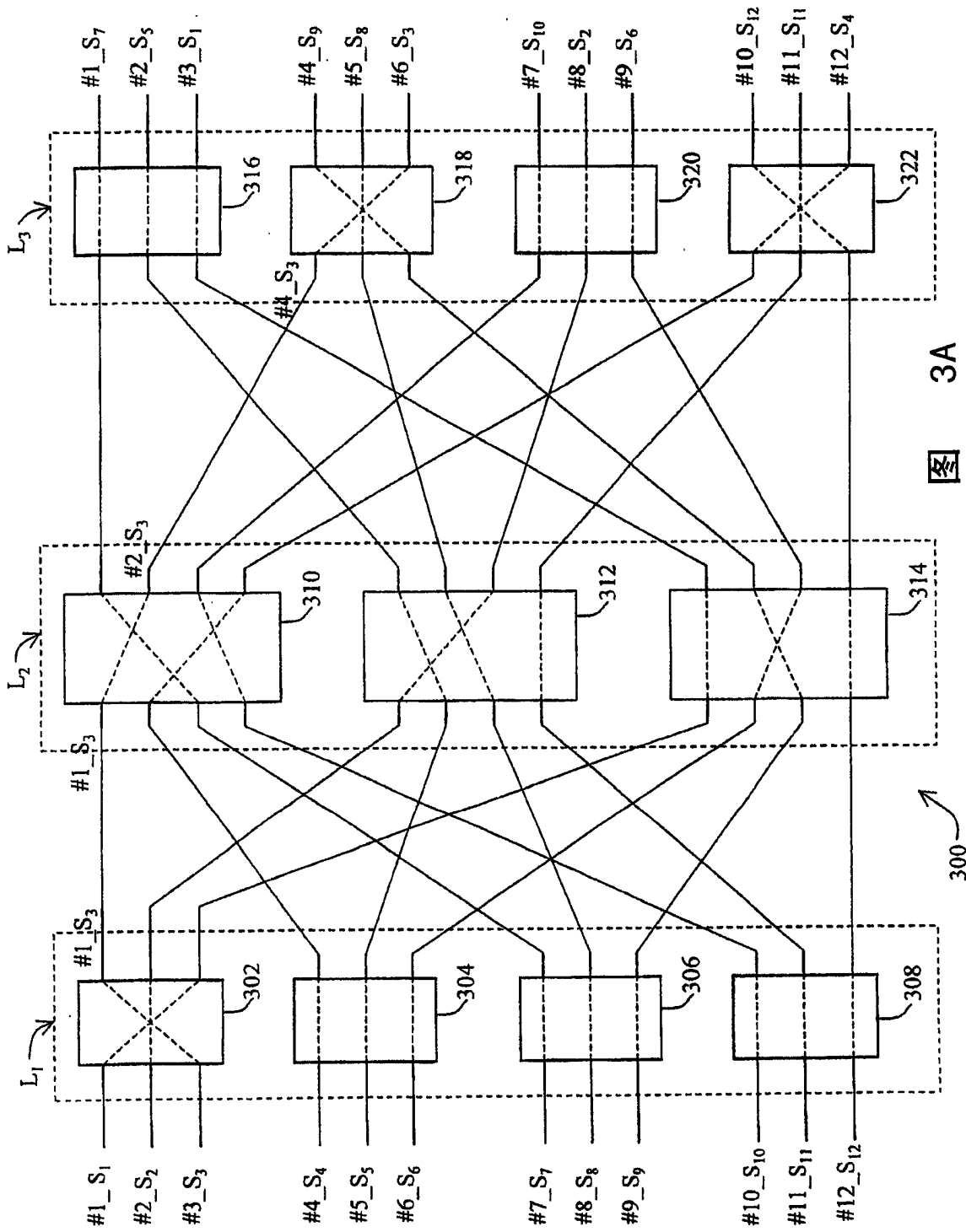


图 3A

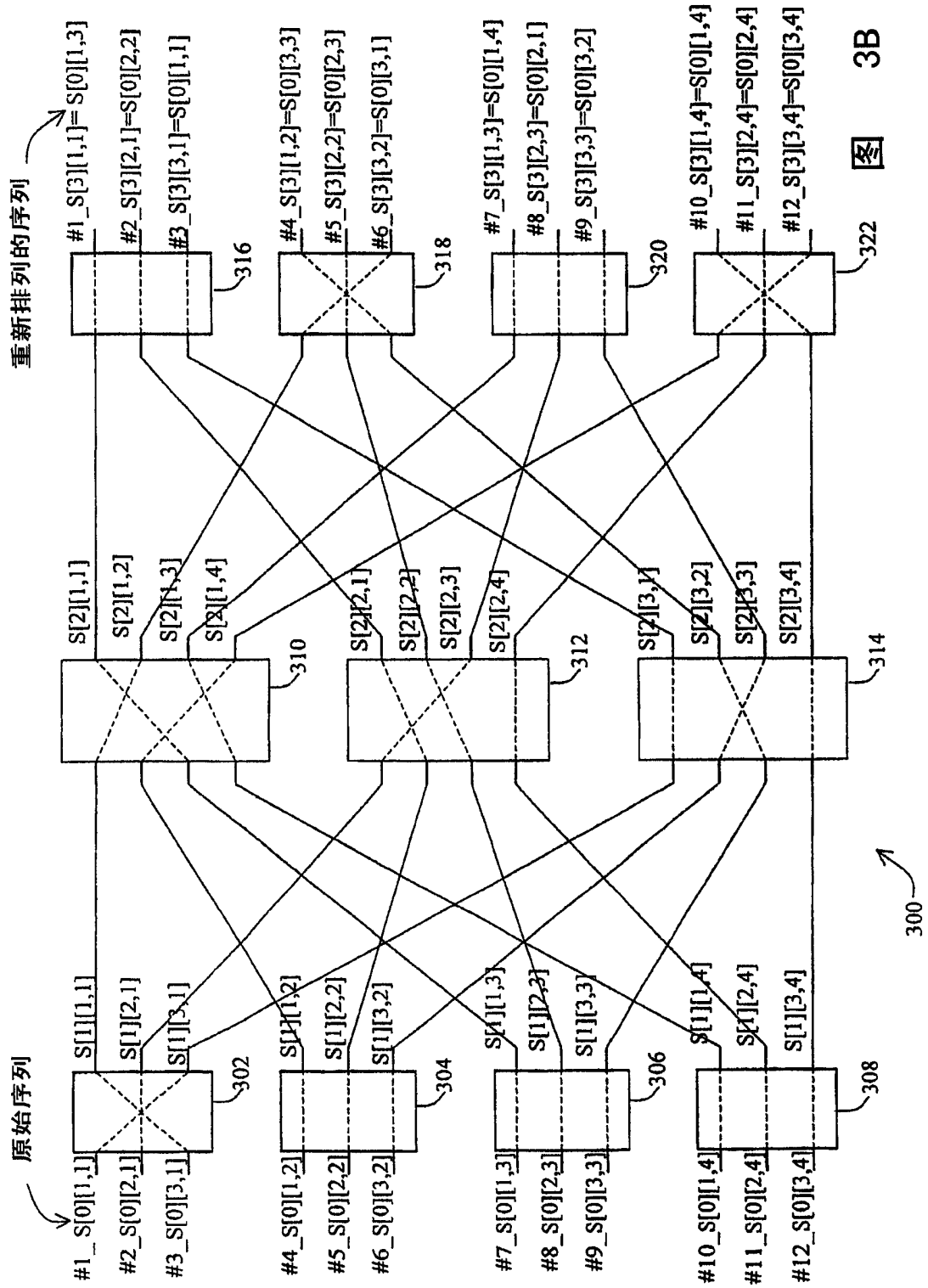


图 3B

## 映射目标

输入#1 : [1,1] → 输出#1 : [3,1]  
输入#2 : [2,1] → 输出#2 : [2,3]  
输入#3 : [3,1] → 输出#3 : [3,2]  
  
输入#4 : [1,2] → 输出#4 : [3,4]  
输入#5 : [2,2] → 输出#5 : [2,1]  
输入#6 : [3,2] → 输出#6 : [3,3]  
  
输入#7 : [1,3] → 输出#7 : [1,1]  
输入#8 : [2,3] → 输出#8 : [2,2]  
输入#9 : [3,3] → 输出#9 : [1,2]  
  
输入#10 : [1,4] → 输出#10 : [1,3]  
输入#11 : [2,4] → 输出#11 : [2,4]  
输入#12 : [3,4] → 输出#12 : [1,4]

图 4A

## 切换序列

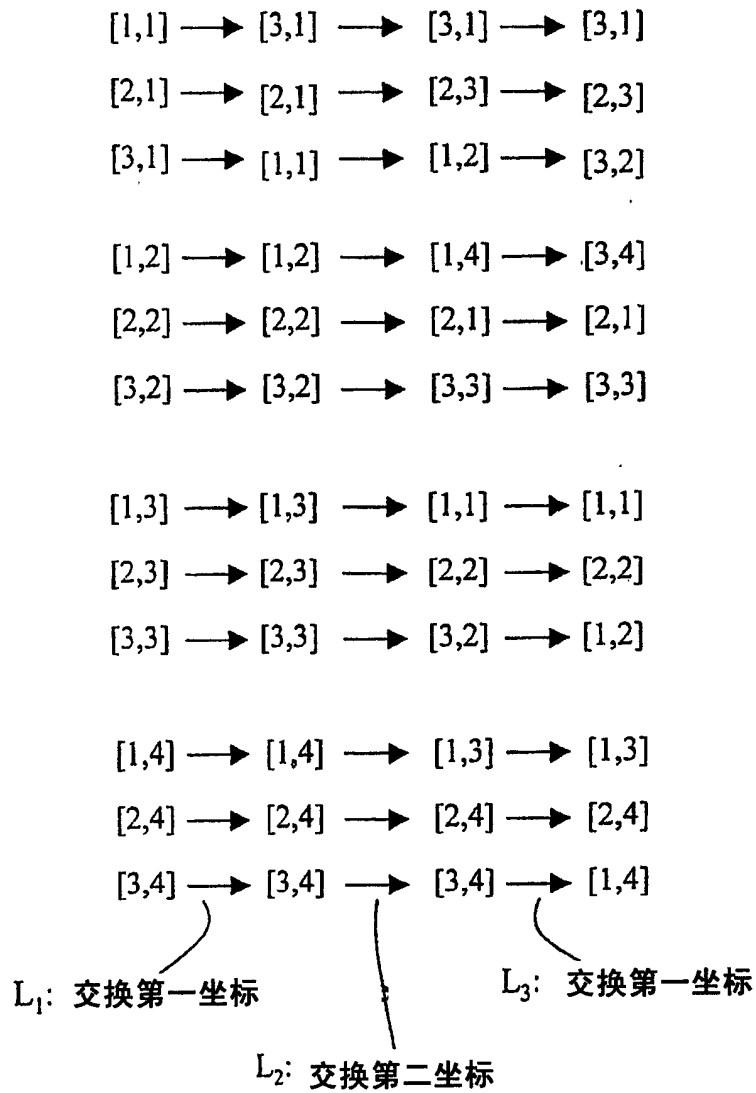


图 4B

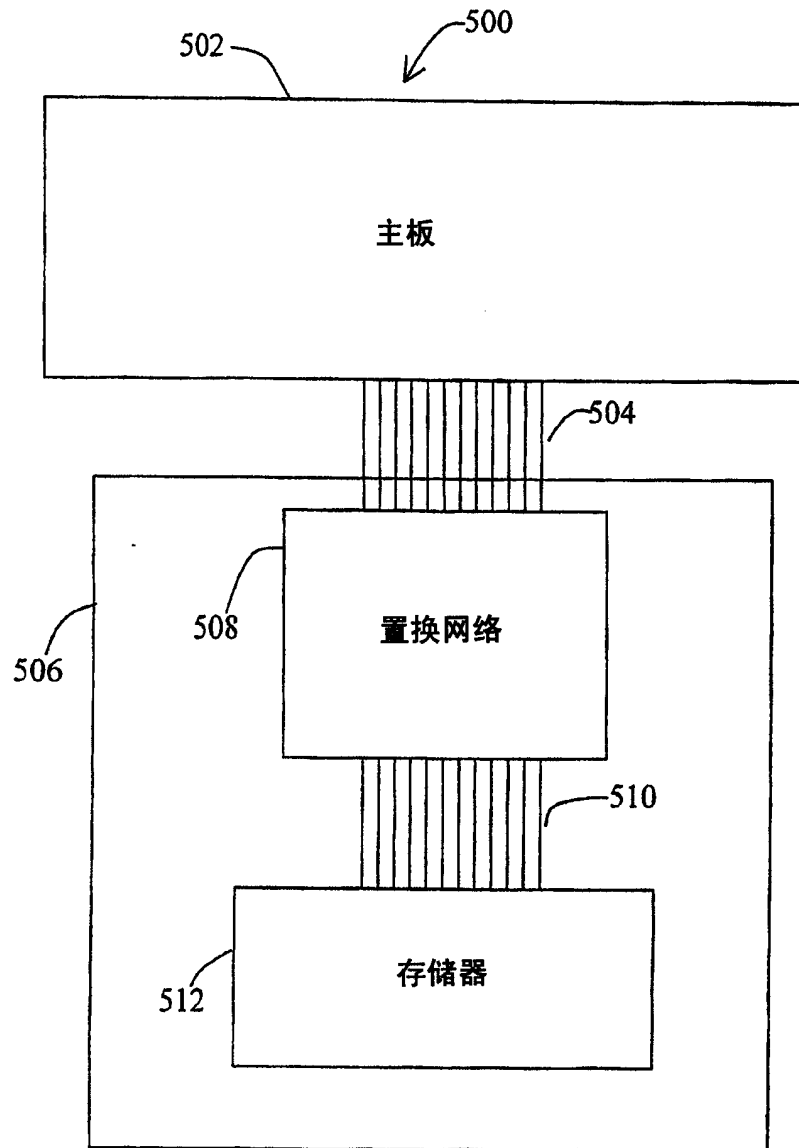


图 5A



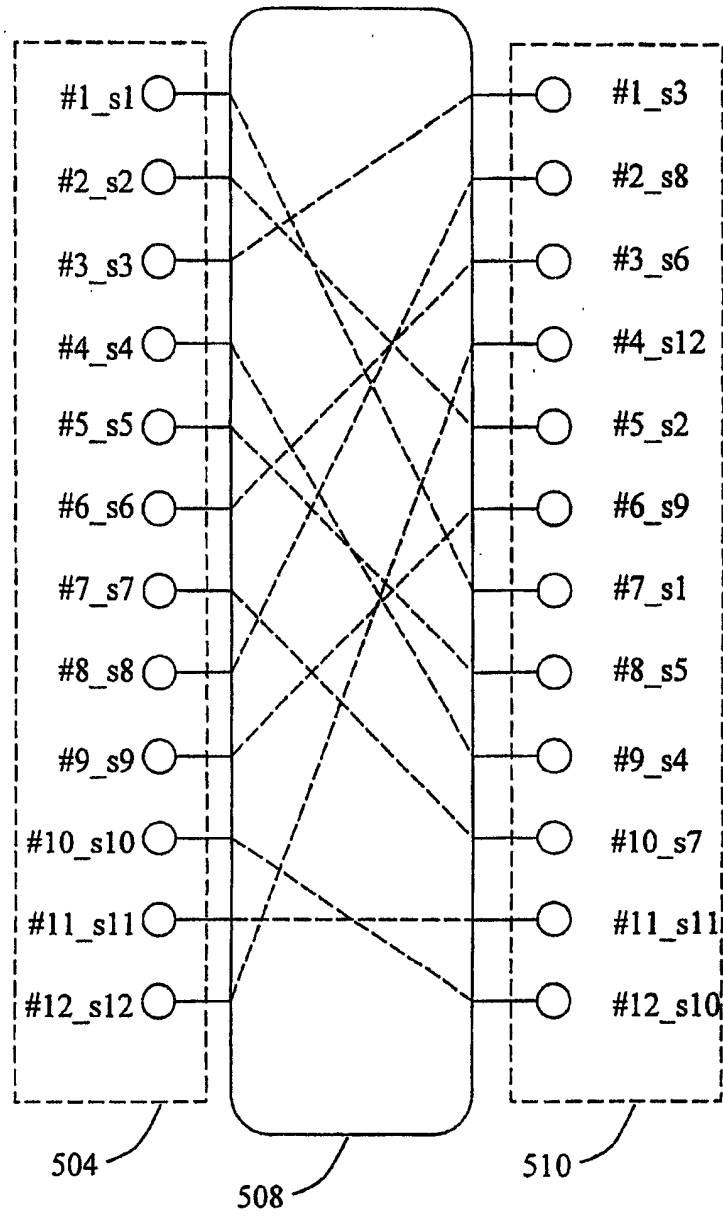


图 5B

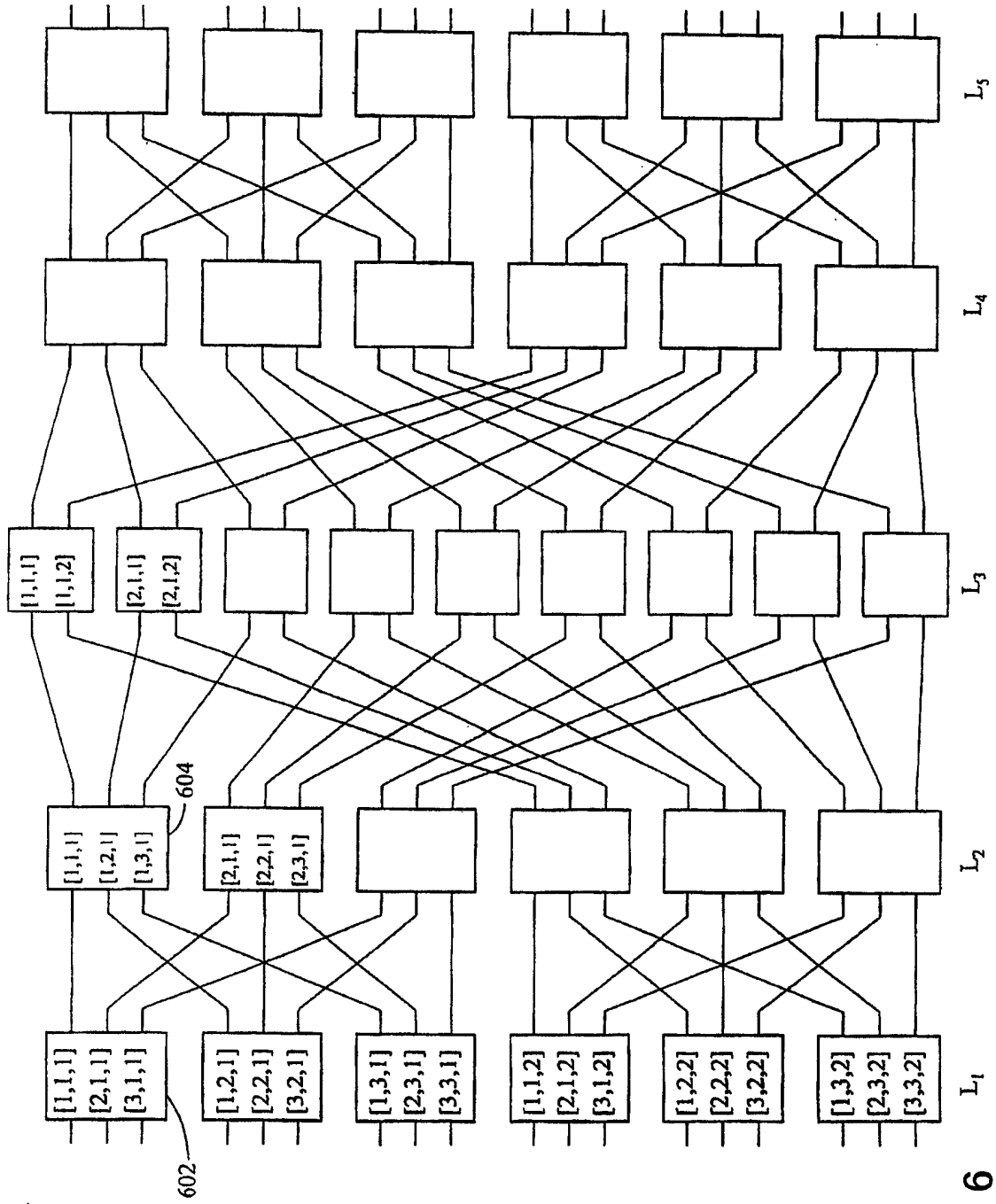


图 6