



(12) 发明专利申请

(10) 申请公布号 CN 101872333 A

(43) 申请公布日 2010. 10. 27

(21) 申请号 200910136216. X

(51) Int. Cl.

(22) 申请日 2006. 04. 17

G06F 13/40 (2006. 01)

(30) 优先权数据

60/674, 189 2005. 04. 21 US

60/698, 626 2005. 07. 11 US

(62) 分案原申请数据

200680013065. 3 2006. 04. 17

(71) 申请人 提琴存储器公司

地址 美国新泽西伊泽林伍德大道南 33 号 8 楼

(72) 发明人 乔恩·C. R.·班尼特

(74) 专利代理机构 广州三环专利代理有限公司

44202

代理人 郝传鑫

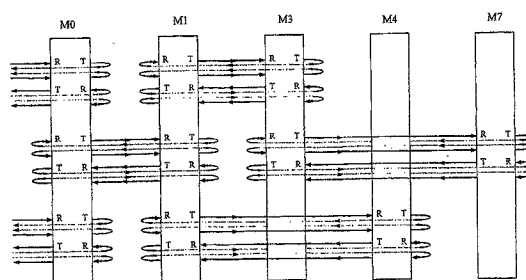
权利要求书 2 页 说明书 45 页 附图 136 页

(54) 发明名称

一种互连系统

(57) 摘要

本发明描述了一种在网络中安排元件的互连系统、设备以及方法,所述互连系统可以是数据存储系统、计算机系统或通信系统,其数据通路被配置成可以控制系统的功率消耗和数据歪斜特性;可配置的开关单元用于形成节点间的互连,其控制信号和其他信息被用于控制所述可配置的开关单元其他方面的功率状态;可以通过调整网络间一个或多个节点的逻辑和物理线路排布来改变网络中节点间的传输的数据的时延歪斜。本发明还公开了一种互连主板的布局方法,可以减少线迹(trace)路由的复杂性。



1. 一种互连系统,其特征在于,包括:多条信号线路,每一个线路可以进行传送或接收数据信号中的至少一个,该信号包括一字中的一个比特,其中,所述数据字有第一比特和第二比特,第一比特和第二比特中的每个均可在两条信号线路之间进行逻辑交换,以使当所述字在信号线路的远端被接收到时,所述交换可以改变第一比特和第二比特之间的差分时间延迟。

2. 如权利要求1所述的互连系统,其特征在于,所述逻辑交换是在信号线路的近端完成的。

3. 如权利要求1所述的互连系统,其特征在于,所述逻辑交换的排布是在信号线路的远端完成的。

4. 如权利要求1所述的互连系统,其特征在于,在远端的比特的逻辑排布可被改变,以使和近端的各比特具有相同的逻辑顺序。

5. 如权利要求1所述的互连系统,其特征在于,所述数据包括具有至少两个比特的字,其第一比特被指派给第一线路,第二比特被指派给第二线路。

6. 如权利要求1所述的互连系统,其特征在于,在数据被读取的第二节点区域,所述数据的到达时间的差可以改变。

7. 如权利要求1所述的互连系统,其特征在于,在数据被去歪斜的第二节点区域,所述数据的到达时间的差可以改变。

8. 如权利要求1所述的互连系统,其特征在于,至少在一个模块控制器中,可以配置第一节点或是第二节点,使得在第二节点接收的数据与一个模块控制器或第一节点上的输入数据的逻辑顺序一样。

9. 如权利要求1所述的互连系统,其特征在于,所述逻辑排布在链路的发送端完成。

10. 如权利要求1所述的互连系统,其特征在于,所述逻辑排布在链路的接收端完成。

11. 如权利要求1所述的互连系统,其特征在于,所述逻辑排布在介于第一节点和第二节点之间的第三节点完成。

12. 一种互连系统,其特征在于,包括:

多条信号线路,

主板,具有多个用于与存储器模块接口的连接器,其中,一个字包括第一比特和第二比特,所述两个比特与至少两条信号线路相关,并被逻辑上的指派在每一条信号线路上,以改变信号线路的远端接收到的比特的差分时间延迟。

13. 一种互连系统,其特征在于,包括:

多个节点,所述多个节点通过具有线路的链路相互连接形成网络;

主板,该主板包含多个连接器,用于与多个存储器模块进行接口连接,其中,第一节点是数据源,第二节点是数据目的地,至少有一个第一节点或第二节点被配置成可以交换到线路上的数据的逻辑排布,以改变在第二节点线路上收到数据字的时间。

14. 如权利要求13所述的互连系统,其特征在于,多个节点中的一个节点是安装在多个存储器中的一个上。

15. 如权利要求13所述的互连系统,其特征在于,所述系统还包括模块控制器,其中,可进行在输入到第一节点的线路上的一个字中的各比特的逻辑排布,以使在第二节点以一预定逻辑顺序接收该些比特。

16. 一种节点,其特征在于,包括:

输入端口及输出端口,每一个输入端口和输出端口具有多条线路;

切换开关,用于使多条输入线路中的每一条可连接到多条输出线路中的每一条,以调整信号在多条输出线路上的时间歪斜,以使在具有与来自输出端口的线路的远端通讯的输入端口进行测量时,使该多条输出线路上信号之间的时间歪斜被改变。

17. 一种互连系统,其特征在于,包括:

多个节点,所述多个节点通过具有线路的链路相互连接形成网络;

第一节点为信号源,第二节点为信号目的地;及

用于减少在第一节点和第二节点之间的两条或多条线路上信号的差分时间延迟的装置。

18. 一种网络控制器,其特征在于,包括:

与计算机通信的第一接口;

具有用于与节点通信的链路的第二接口;

用于确定布置在第一和第二节点之间的链路的线路上的信号之间的差分时间延迟的装置;以及

用于指派逻辑信号给在第一和第二节点之间通信的线路,以改变在第二节点上接收到的逻辑信号的延迟时间的装置。

19. 一种网络控制器,其特征在于,包括:

计算机接口;

第一数据接口,可以传送和接收信号;以及

处理器,用于产生用于通过第一数据接口传送的地址值,所述处理器还可以逻辑上指派数据值给信号线路,使得至少两条信号线路上的时间延迟差在接收设备接收到信号时可以被改变。

20. 一种节点,其特征在于,包括:

输入节点和输出节点,每一个所述输入节点和输出节点具有多条线路;以及

开关,可以将多条输入线路中的每条选择性地连接到多条输出线路中的一条,以在具有与来自输出端口的线路的远端通讯的输入端口进行测量时,使该多条输出线路上信号之间的时间歪斜被改变。

21. 如权利要求 20 所述的节点,其特征在于,还包括:第一和第二输出端口,该第一输出端口被配置成与存储器连接。

一种互连系统

[0001] 本申请文件要求在 2005 年 4 月 21 日申请的,临时申请号为 No. :60/674, 189 的美国申请,以及在 2005 年 7 月 11 日申请的,申请号为 No. :60/698, 626 的美国申请的权利,每一项申请在此处结合并被整体参考。

背景技术

[0002] 一般来说,计算机和通信技术的发展遵循着摩尔定律原则,其密度和组件都有所增长,而且,随着时间的推移,其组件的成本有所下降,但同样也伴随着传输和计算速度的增加。这里有一种趋势是增强功率消耗密度和增加新的解决方案,以可在高温的环境下散热或运作。在高速电路中,信号传播时延和设计以及制造公差都会限制实践中加速的实现。

[0003] 现有技术的存储系统结构需要在价格、性能以及升级能力之间做出权衡,例如,系统总的存储能力。增加存储能力通常是由具有连接器/插座接口的存储器模块或者卡来实现的。通常,这些存储器模块被连接至总线或者通过置于底板上的配线相互连接,以更有效地利用系统资源。除了升级能力以外,许多现有技术的存储系统也需要为带宽要求较大的应用提供较大的吞吐量,比如制图。

[0004] 在设计范围内应该考虑系统资源,例如,集成电路压膜的区域,封装插脚,信号线迹,连接器,底板区域以及电源和响应速度。

[0005] 一种普通型存储器模块是双插存储器模块 (Dual In-line Memory Module, DIMM), DIMM 是一种矩形的低轮廓电路板,其两侧沿长边排布有电触点。当所述 DIMM 被插入 DIMM 存储插座上时,所述触点形成了到主板存储总线的电气连接。

[0006] 存储系统通过模块的使用提供了一种进行升级的路径。一种插座和连接器接口会被使用,以允许每个模块被移除及被一种更快速或更大存储能力的存储器模块所代替,或者仅仅用于替换已损坏的存储器模块。存储系统可能被配置为无插座形式或者少于存储器模块的最大存储能力,(例如,悬空插座/连接器),而在以后的时间内采用存储扩展模块来增加存储能力。

[0007] 存储器被用在包括但不限于计算机系统或者通信系统中,各种随机存取存储器 (RAM) (例如,静态随机存取存储器 S-RAM,动态随机存取存储器 D-RAM);可编程只读存储器 (PROM),电可擦除只读存储器 (EPROM);快闪存储器,包含磁电机随机存取存储器 (MRAM),磁随机存取存储器 (FRAM) 以及非易失性随机存取存储器 (NRAM),变相存储器 (PRAM) 和媒体磁盘存储器在内的各种类型的磁存储器。除此之外,还有适用于未来的存储器,例如量子设备或类似的。

[0008] 在这,需要为存储系统和计算机系统结构或者拓扑结构在提供高速的带宽以维持微型处理器运行频率的速度时,提供灵活而成本效率的升级能力。可选地,对于相同或相似操作特性的低功率消耗和低存储反应时间以及类似特性也是一种需求。

[0009] 目前,存储系统的大小和性能由功耗、与功耗和装置密度相关的冷却能力、响应时间(如,由数据歪斜产生的)、以及需要与存储元件互连的电路板的复杂度,以及这些因素的经济成本所限制。需要对存储器、通信系统结构及装置进行改善,以减轻这些问题的一些

或全部。

发明内容

[0010] 本发明公开了一种与计算机和通信系统组件一起使用的互连系统,包括:多个节点,每一个节点具有一个端口,每一个端口都至少具有一个输入链路(link)或一个输出链路,每一个链路都包含有多条线路,第一节点的第一线路用于传输控制数据,第二节点的第一线路上接收的信号用来控制第二节点的第二线路的将来时间的功率状态。第一端点的输出线路的状态可在将来时间由控制信号决定而改变。将来时间至少包含一个固定时间值,这取决于互连系统中节点的位置,或者控制信号中包含的一个时间值。

[0011] 另一方面,提供了一种互连系统,多个节点,每个节点具有一个端口,该端口具有至少一个输出链路或输入链路,每一个链路包括一条第一线路和一个第二线路;所述连接器进一步包括主板,所述主板上有多条连接器,每一个连接器都与多个存储器模块中的一个接口连接,其中,一存储器模块与至少所述多个节点中的一个通讯;其中,每一个信号线路包括至少一个信号接收器或信号发送器;其中,第一节点的第一线路传输第一信号,第二节点的第一线路上接收的信号用来控制第二节点的第二线路的将来时间的功率状态。

[0012] 每个节点都是可配置的,以使控制数据可以控制所述节点接收的数据的目的地。数据的目的地从本地节点或者端口中选择。

[0013] 另一方面,还提供了一种互连系统,包括:多个节点,每一个节点具有一个端口,每一个端口具有输出链路或是输入链路中的一个,每一个链路进一步包括多条线路,且所述每个节点都是可配置的,以使在所述多条线中的一条选择的线路上接收到控制信号。可以在相同或者不同的线路上接收控制数据和信号数据。所述信号数据可以是模拟数据或者数字数据。所述数字数据被组织成字。在这里,字只是一种泛指,除了特殊情况外,不限制字的比特数,其传输的形式是串行或是并行的也不受限制。因此,一些字可以是例如,“半位元组(nibble)”,“字节”,全字、双字,具有或不具奇偶的字,及可以是具有非标准数目的比特的字。

[0014] 另一方面,还提供了一种互连系统,包括:多个节点,每一个节点具有一个端口,每一个端口具有输出链路或是输入链路中的一个,每一个链路进一步包括第一线路和第二线路,第二节点的第一线路上接收的信号用来控制第二节点的第二线路的将来时间的功率状态。所述状态在开电或关电状态中变化。在开电状态下至少可进行控制或者数据信号的传送或接收,待机状态比开电状态的功耗小,关电状态比待机状态的功耗小。

[0015] 在再一方面,提供了一种互连系统,包括:多条信号线路,每条线路可以进行传送或接收数据信号中的至少一个,该信号包括一字中的一个比特,其中,所述数据字有第一比特和第二比特,第一比特和第二比特中的每个均可在两条信号线路之间进行逻辑交换,以使当所述字在信号线路的远端被接收到时,所述交换可以改变第一比特和第二比特之间的差分时间延迟(differential timedelay)。比特的交换,可以在信号线路的最近端或者最远端完成。随后,可在远端逻辑上重新安排字节中的各比特的位置,使得接收节点的字中的各数据比特具有正确的逻辑排布。可选地,比特在发送节点被安排,以使在接收节点上该些比特处于布置中。

[0016] 另一方面,提供一种互连系统,包括:多个节点,所述多个节点通过具有线路的链

路相互连接形成网络；其中，第一节点是数据源，第二节点是数据目的地，至少有一个第一节点或第二节点被配置成交换到线路的数据的逻辑排布，以使在第二节点线路上数据字节的到达时间的差可被改变。

[0017] 所述数据可以是一个包含至少两个比特的字，其第一个比特被指派给第一线路，第二比特指派给第二线路。一控制模块，对所述输入第一节点的线路的字的比特进行逻辑排布，以使得这些比特第二节点被以预期的顺序接收。可选择地，还可以在第一节点和第二节点之间的第三节点上实现所述逻辑配置。

[0018] 另一方面，提供了一种互连系统，包括多个节点，所述多个节点通过具有线路的链路相互连接形成网络；第一节点为信号源，第二节点为信号目的地；及用于减少在第一节点和第二节点之间的两条或多条线路上信号的差分时间延迟的装置。

[0019] 提供了一种节点，所述节点具有输入端口及输出端口，每一个输入端口和输出端口具有多条线路；切换开关，用于使多条输入线路中的每一条可连接到多条输出线路中的每一条，以在具有与来自输出端口的线路的远端通讯的输入端口进行测量时，使该多条输出线路上信号之间的时间歪斜被改变。

[0020] 提供一种存储器模块，包括：读写存储单元；可配置的切换开关元件（CSE，configurable switching element），所述元件进一步包括：用于接收或传输信号中至少一种的第一端口，所述第一端口具有多条信号线路；其中，CSE 被配置成解析接收到的信号以实现下述中的至少一种：改变信号线路中一条线路的将来时间的状态，及路由接收到的信号到读写存储设备或第二端口中的一个。每个信号线路状态可以是开电、关电或者待机。

[0021] 另一方面，所述将来时间的值为由控制信号决定的或为一个可配置的值，也可能是为每条线路所分别确定的。

[0022] 另一方面，存储器模块的 CSE 是可配置的，以使由接收信号线路接收的数据的逻辑排布和由传输信号线路传输的数据的逻辑排布是可以改变的。

[0023] 所述存储器模块 CSE 是可配置的，以使第一、第二或是第三端口上的信号线路接收到的信号可以在上述三个端口上的至少一路线路上传输。传输和接收信号线路可以是单向线路、双向线路或者可逆的线路。

[0024] 在进一步的一方面，在一个固定时间间隔内，数据信号，控制信号或者同步信号中的至少一个在连接到 CSE 上或存储器模块的链路中的每条线路上传输。所述固定时间间隔可以被确定，以使数据沿跟踪，时钟，时钟相位，或者或帧同步中至少一个被维持。

[0025] 提供了一种存储器模块，具有用于存储数据的装置；用于控制一个或多个输入或输出接口的将来时间状态的装置；以及用于路由数据到输出接口或用于所述存储数据的装置中至少一个的装置。

[0026] 在另一方面，该存储器模块具有一个用于为信号交换逻辑和物理接口的排布的装置，以使在第二存储器模块的输入接口上测量时，该第一存储器模块的输出接口上逻辑信号之间的差分时间延迟被改变。在另外一个方面，该用于为信号交换逻辑和物理接口的排布，以使第一存储器模块的输出接口上逻辑信号之间的差分时间延迟被改变的装置包括但不限于，在模块上的电路迹的固定排布，所述模块的安装板，芯片封装底层，连接，作为他们物理上的或者邻近的，所述装置独自处在多芯片模块的芯片之间或者与其他固定或动态的装置相结合。

[0027] 另一方面,用于交换逻辑和物理接口上的信号的数据分配,以使在逻辑信号之间的差分时间延迟被改变的装置包括:通过 CES 和其他置于两个或者更多 CES 之间切换元件完成动态切换功能。所述切换元件包括数字开关,模拟 /RF 开关元件,如微型机电系统装置一类的机械开关,可以配置的邻近通信设备 (proximity communications devices) 或者其他类似的设备等等,所述切换元件可以是独立的,也可以是和其他动态或者固定装置相结合。

[0028] 提供了一种可配置开关元件 (CSE),具有至少两条信号线路,所述第一信号线路配置为接收控制信号,第二信号线路处于可配置的状态。所述第二信号线路的状态可以是开电,关电或者待机状态。所述第一信号线路接收的控制信号用于控制第二信号线路的状态。其中,至少一条信号线路用于接收地址数据,其他信号线路上接收到的数据被发送到其他的信号线路上或者存储接口中的至少一个。所述存储接口连接到一个用于存储数据的装置。

[0029] 另一方面,可配置的开关元件包括第一接口,用于接收或者传输信号中至少一个,所述第一接口具有多条信号线路,且所述 CSE 配置用来解释接收到的控制信号以完成如下性能:基于接收到的控制信号改变多条信号线路中至少一条的状态,或路由接收到的信号到其他接口,所述其他接口可以是存储接口,第二接口用于接收或者传输信号,或者传输第一接口的信号。所述多条信号线路的状态是开电,关电,或者待机状态。

[0030] 另一方面,CSE 是可配置的,以实现在接收信号线路上的数据的逻辑排布和到传输信号线路上的数据的逻辑排布之间的逻辑数据交换。这里还有可配置的第三接口,使在任何第一、第二或第三端口的任何信号线路上接收的信号都可以在任何第一、第二或第三端口的任何信号线路上传输。。

[0031] 提供了一种可配置的开关元件 (CSE),具有用于控制一个或多个输入或输出接口的将来时间状态的装置;及用于路由数据至所述输出接口或至一用于存储数据的装置的装置。

[0032] 另一方面,所述可配置开关元件具有,用于为信号交换逻辑和物理接口数据排布的装置,以在一第二 CSE 上的第二接口上进行测量时,该在第一输出接口和第二输出接口上的信号之间的差分时间延迟被改变。

[0033] 具有顶端和底端的连接接口,所述接口包括多条信号线路形成第一、第二和第三链路关系;

[0034] 由一个或多个第一、第二或第三链路与所述连接接口相连形成的网络,其中,至少采用下述中两种配置方式通过线迹连接所述连接接口:

[0035] (a) 布置第一至第三链路,使得所有的链路远离第一侧的连接接口;

[0036] (b) 布置第一至第三链路,使得一链路具有一条线路离顶端最近,而远离连接接口的第一侧,另外两条链路远离连接接口的第二侧,且所述远离连接接口的第二侧的链路邻近布置;

[0037] (c) 布置第一至第三链路,使得一链路具有一条线路离底端最近,而远离连接接口的第二侧,另外两条链路远离连接接口的第一侧,且所述远离连接接口第一侧的链路邻近布置;

[0038] d) 布置第一至第三链路,使得一链路具有一条线路离顶端最近,及一链路具有一

条线路离底端最近而远离连接接口的第一侧,以及一链路具有一条线路位于之前所述两个链路之间而远离连接接口的第二侧;以及

[0039] (e), (f), (g), 和 (h), 在此, (e), (f), (g) 和 (h) 为 (a), (b), (c), 和 (d), 不同之处在于, 其中, 通过绕与所述连接器垂直的轴将 (a), (b), (c) 和 (d) 的图案旋转 180 度而使顶端和底端相互交换。

[0040] 另一方面, 链路布置在平面印刷集成电路板上, 或者所述链路布置为底层的各层上的导电金属。其他组件, 如存储器装置, CSE 和其他主动或被动的电路, 可被置于相同的底层或者配线上。

[0041] 另一方面, 至少有三个连接器接口彼此并行布置。

[0042] 提供一种布局网络间互相连接的方法, 包括: 布置多个连接接口, 使得连接接口的顶端彼此相对, 连接接口的底端彼此相对; 在多个连接接口之间形成一个连接的网络, 并至少使用下述连接接口配置方法中的两种:

[0043] (a) 布置第一至第三链路, 使得所有的链路远离第一侧的连接接口;

[0044] (b) 布置第一至第三链路, 使得一链路具有一条线路离顶端最近, 而远离连接接口的第一侧面, 另外两路链路远离连接接口的第二侧, 且所述远离连接接口的第二侧的链路邻近布置;

[0045] (c) 布置第一至第三链路, 使得一链路具有一条线路离底端最近, 而远离连接接口的第二侧面, 另外两条链路远离连接接口的第一侧, 且所述远离连接接口第一侧的链路邻近布置;

[0046] (d) 布置第一至第三链路, 使得一链路具有有一条线路离顶端最近, 及一链路具有一条线路离底端最近而远离连接接口的第一侧, 以及一链路具有一条线路位于上述两条链路之间而远离连接接口的第二侧; 以及

[0047] (e), (f), (g), 和 (h), 此处, (e), (f), (g) 和 (h) 为 (a), (b), (c), 和 (d), 不同之处在于, 其中, 通过绕与所述连接器垂直的轴将 (a), (b), (c) 和 (d) 的图案旋转 180 度而使顶端和底端相互交换。

[0048] 提供一种网络控制器, 具有: 与计算机通信的第一接口; 具有用于与节点通信的链路的第二接口; 用于确定布置在第一和第二节点之间的链路的线路上的信号之间的差分时间延迟的装置。

[0049] 在另一方面, 该网络控制器可具有一个用于用于指派逻辑信号给在第一和第二节点之间通信的线路, 以改变在第二节点上接收到的逻辑信号的延迟时间的装置。

[0050] 本发明一种网络控制器, 包括: 计算机接口; 第一数据接口, 可以传送和接收信号; 处理器, 用于产生由第一数据接口传送的地址信号, 所述处理器还可以为信号线路的逻辑上指定数据值, 使得至少两个信号线路上的时间延迟差在接收器端接收到信号时可以被改变。

附图说明

[0051] 从以下几个比较详细实施例的描述, 以及附图, 可以更好的理解本发明。但是, 所述实施例和附图, 不能将本发明限制在所述具体的描述中, 仅用于帮助理解本发明。

[0052] 图 1 是一个网络中互连元件的示意图, 目的在于介绍本发明所使用的术语和符

号；

- [0053] 图 2 是可配置开关元件 (CSE) 接口的示意图；
- [0054] 图 3 是具有一个以上次级端口的 CSE 的配置示意图；
- [0055] 图 4 是在网络中与一个以上模块控制器相连的 CSE 的配置示意图；
- [0056] 图 5 是由 8 个包含 CSE 的存储器模块组成的树结构示意图；
- [0057] 图 6 是由 8 个存储器模块组成的二叉树结构的示意图；
- [0058] 图 7 是由 8 个存储器模块组成的二叉树结构的另一实例示意图；
- [0059] 图 8 是由 9 个存储器模块组成的三叉树结构的示意图；
- [0060] 图 9 是由 8 个存储器模块组成二叉树结构的另一种实例示意图；(其中 M0 的子模块同时与 M0 和 MC 相连)
- [0061] 图 10 是由 8 个存储器模块组成二叉树结构及附加设备 (如图形处理器) 的示意图；
- [0062] 图 11a-f 是与 INTEL 公司的 FB-DIMM 相似的存储系统的工作方式的示意图；
- [0063] 图 12a-f 是具有 CSE 的存储器模块的一种排布图；
- [0064] 图 13a-f 是在模块间具有独立的命令和数据线路的存储器模块的一种排布图；
- [0065] 图 14a-f 是在模块间具有独立的命令和数据线路的存储器模块的另一种排布图；
- [0066] 图 15a-f 是从模块控制器 (MC) 到模块 Mi 的读命令的传输过程的示意图；
- [0067] 图 16 是具有点对点链路的树结构中模块控制器和模块的排布图；
- [0068] 图 17a-d 是与图 16 类似的具有独立的命令信号链路的排布图；
- [0069] 图 18 示出了用于使用地址指示器来实现其路由的命令的实例；
- [0070] 图 19 是切换开关 S 的过程图,其从一个或多个输出端口传送 NOP 命令；
- [0071] 图 20 是一种链路为信道化的模块树结构图；
- [0072] 图 21 是另一种链路为信道化的模块树结构图；
- [0073] 图 22 是数据传输的一种程式化表示的示意图；
- [0074] 图 23a-d 是多数链路处于低功率状态的配置图；
- [0075] 图 24a-d 是与图 23 类似同时链路可处于更低功率的配置图；
- [0076] 图 25 是具有本地命令扩展表的配置图；
- [0077] 图 26a-b 是从 S1 的输入到树的叶子节点的控制流向的示意图；
- [0078] 图 27a-b 是可选用的自路由消息的配置图；
- [0079] 图 28a-b 是适用于可去除头比特的消息格式的示意图；
- [0080] 图 29a-b 是从诸如模块控制器向其他设备传输数据的一个简单实例的示意图；
- [0081] 图 30 是用于克服歪斜的一种方法和设备的示意图；
- [0082] 图 31 是在去歪斜后插入的多路输出选择器的示意图；
- [0083] 图 32 是在 INTEL 的先进存储缓存器 (AMB) 中引起数据歪斜的原因的示意图；
- [0084] 图 33a-b 是可减小信号歪斜的信号路由的一种配置图；
- [0085] 图 34 是在 5 个模块中传输形成数据歪斜的示意图；
- [0086] 图 35 是用于减少线路之间的延迟时间差的设备和方法的示意图；
- [0087] 图 36a-b 是模块对之间的延迟差别较大时的例子的示意图
- [0088] 图 37 是将线路 3 和 4 作为信号信道时的例子的示意图；

- [0089] 图 38a-c 是一种在线路间交换数据元件的模式示意图；
- [0090] 图 39 是一种 Batcher 网络实例图；
- [0091] 图 40 是另一种 Batcher 网络 (a-c) 和 Clo 网络 (d) 的实例图；
- [0092] 图 41a-b 是一种 Benes 网络和具有一个 8 条线路配置的实例图；
- [0093] 图 42 是一种部分线路中使用全交换的线路交换的实例图；
- [0094] 图 43 是模块中两条数据通路的合并的示意图；
- [0095] 图 44 是具有相位校准的时钟数据恢复 (CDR) 电路的示意图；
- [0096] 图 45 是在图 44 的基础在两条数据通路上分别加有去歪斜功能的实例图；
- [0097] 图 46a-b 是在更详细的网络中的图 45 的排布图；
- [0098] 图 47 是具有非临界数据通路的网络的方式示意图；
- [0099] 图 48 是在 8 个模块组成的树结构的网络中携带逻辑比特 0 的线路的示意图；
- [0100] 图 49 是在 8 个模块组成的同图 48 一样的树结构的网络中携带逻辑比特 N 的线路的示意图；
- [0101] 图 50 是标识了在图 49 显示的排布中临界和非临界通路的示意图；
- [0102] 图 51 是图 49 中的延迟的示意图；
- [0103] 图 52 是额外的可变延迟的示意图；
- [0104] 图 53 是在一个链路上的额外延迟的示意图；
- [0105] 图 54a-b 是网络中加入延迟的等同的方法的示意图；
- [0106] 图 55 是图 46 和 49 中所示的树结构的延迟的时间线的示意图；
- [0107] 图 56 是表示在网络中的延迟的另一种方式的示意图；
- [0108] 图 57 是在树中发送命令和接收响应的总的时间的示意图；
- [0109] 图 58a-c 是相同模块可具有用于传输到及接收自存储控制器的不同路由情况的示意图；
- [0110] 图 59 是图 48 中的树结构包括外向和返回通路的示意图；
- [0111] 图 60a-b 是具有与数据通路不相连的多信号通路的排布的示意图；
- [0112] 图 61a-c 是从模块到模块控制器的数据传输的时序图；
- [0113] 图 62a-c 是使用固定时隙从模块到模块控制器的数据传输的时序图；
- [0114] 图 63 是另一个执行多命令时的时序实例图；
- [0115] 图 64 是相对图 62 具有不同的延迟排程的一个实例图；
- [0116] 图 65a-c 是相对图 63 具有不同的延迟安排的一个实例图；
- [0117] 图 66a-c 是一种排程实例示意图, 其中, 移除了数据到间隙发送次数的限制；
- [0118] 图 67a-c 是另一种排程实例示意图, 其中, 移除了数据到间隙发送次数的限制；
- [0119] 图 68 是用于一种非时隙实例的时间选择示意图, 其中, 显示有信号延迟；
- [0120] 图 69a-c 是用于另一种非时隙实例的时间选择示意图, 其中, 显示有信号延迟；
- [0121] 图 70 是图 68 中具有不同的命令时间的示意图；
- [0122] 图 71 是具有多命令的另一个时间矢量图；
- [0123] 图 72 是以线性方式布局的与图 5 类似的树结构的示意图；
- [0124] 图 73 是图 72 中在其模块上标注有“前侧”和“后侧”的排布图；
- [0125] 图 74a-d 是一种可选择的模块的排布图；

- [0126] 图 75a-b 是模块之间除主板之外的连接示意图；
- [0127] 图 76 是图 72 中具有可选择的模块连接的排布图局部示意图；
- [0128] 图 77 是图 72 中具有可选择的模块连接的排布图局部示意图；
- [0129] 图 78 是用于图 77 的排布中的模块连接的另一例子的示意图；
- [0130] 图 79 是各链路之间具有不同宽度的模块的排布图；
- [0131] 图 80 是模块组的网格结构图；
- [0132] 图 81a-b 是模块组的六边形网格结构图；
- [0133] 图 82 是由六边形网格组成的六边形网络结构图；
- [0134] 图 83 是网格配置中的模块与处理器相连的示意图；
- [0135] 图 84 是处理器与具有两个根节点的树相连的配置图；
- [0136] 图 85a-b 是模块与多处理器相连的另外实例的示意图；
- [0137] 图 86a-b 是模块与多处理器相连的另一实施示意图；
- [0138] 图 87 是将控制和数据总线扩展到共享的存储区域的示意图；
- [0139] 图 88 是具有 4 个处理器和输入 / 输出设备的模块组成的网格图；
- [0140] 图 89 是图 88 的配置中 I/O 模块被其他的计算系统元件所代替的示意图；
- [0141] 图 90 是可提供线性和 / 或树拓扑结构的连接和模块的排布图；
- [0142] 图 91 是图 90 的排布用于线性结构的示意图；
- [0143] 图 92 是图 90 的排布用于树结构的示意图；
- [0144] 图 93a-b 是布置在模块上的 CSE 和 RAM 的排布图；
- [0145] 图 94 是具有独立的数据和控制线路的 CSE 和 RAM 芯片的排布图；
- [0146] 图 95 是叶子模块相对于剩余的树结构的物理配置图；
- [0147] 图 96 是叶子模块的另一种物理配置图；
- [0148] 图 97 是具有双向链路的模块的排布图；
- [0149] 图 98 是具有高速缓存的模块的排布图；
- [0150] 图 99 是 CSE 与互连系统相桥接的示意图；
- [0151] 图 100 是按物理上线性排布的模块二叉树示意图；
- [0152] 图 101 是用于图 100 的树结构的可选的连接排布示意图；
- [0153] 图 102 是一个多根节点树结构的排布图；
- [0154] 图 103 是模块间连接的俯视图 (a) 和正视图 (b)；
- [0155] 图 104 是具有冗余结构的多根节点树的一种排布图；
- [0156] 图 105 是具有冗余结构的多根节点树的另一种排布图；
- [0157] 图 106 是当图 105 所示的树中出现模块故障时对树进行重配置后的排布图；
- [0158] 图 107 是具有 4 组每组 16 个模块的模块组的排布图；
- [0159] 图 108 是用于在模块之间具有隔离栅的模块组的排布的空气流排布图；
- [0160] 图 109 是用于模块组的排布的空气流向排布图,以可以控制温度的变化；
- [0161] 图 110 是散热元件的另外排布图；
- [0162] 图 111 是模块间具有由线路组成的链路的双根节点树的配置图；
- [0163] 图 112 是现有设计中在连接的主板上的线迹的布局图；
- [0164] 图 113 是在连接器旋转了 30° 后的主板上的线迹的布局图；

- [0165] 图 114 是对于图 113 的连接器排布中一组以上具有最小弯曲的线路的示意图；
- [0166] 图 115 是图 113 的具有连接器定位的主板的大比例的布局图；
- [0167] 图 116 是将 82 个模块配置为 64 个具有 RAID 错误修正的存储器模块及 2 个空闲模块的逻辑排布图；
- [0168] 图 117 是对于图 116 排布中的物理板布局图；
- [0169] 图 118 是具有两个根节点的 82 个模块组成的树的结构图；
- [0170] 图 119 是图 118 中所示树的局部的物理板布局图；
- [0171] 图 120a-d 是图 119 中所示的线路和多层电路板的各层之间的关系示意图；
- [0172] 图 121 是对于 82 个模块排布的另一个树的逻辑图；
- [0173] 图 122 是图 121 中所示的逻辑树的物理布局图；
- [0174] 图 123 是对于 82 个模块排布的再一个树的逻辑图；
- [0175] 图 124 是图 123 中所示的逻辑树的物理布局图；
- [0176] 图 125 是可用于布局物理板排布的单个模块的连接器的物理特性的示意图。

具体实施方式

[0177] 参考附图可以更好的理解本具体实施例,但是这些具体实施例不应作为对本发明的限制。在相同或不同附图中相同或相似的标注代表同一部件。部件可用数字或 / 和字母缩写标注,选择不同的表示方法仅仅是为了使表示清楚,因此,用数字标注的部件与相同的使用字母标注或字母与数字的组合标注的部件不应理解为不同的部件。

[0178] 需要说明的是,所描述的方法和附图中的装置可以由可执行的指令进行配置或具体实现(其具体方式可为软件、硬件、或软件与硬件的结合)。所述指令可以用普通用途计算机、特殊用途处理器,如数字信号处理器(digital signal processor, DSP)或阵列处理器、或其他类似可使用指令编程以完成各种操作的装置实现。可选择的,上述各种操作可以由特殊的硬件组件来实现,所述硬件组件包含有用于执行所描述的操作硬件逻辑或固件指令,也可以由任何可编程组件和客户端硬件组件的组合来实现,上述组合中可包括模拟电路。

[0179] 本发明具体实施例中提供的方法可以(至少部分可以)由计算机编程产品实现,上述产品包括机器可读媒介,在该媒介上存储了在计算机(或其他电子设备)上运行即可实现上述方法的指令。从这方面考虑,“机器可读媒介”应包括可保存或编码一系列指令或数据的媒介,该指令或数据可在计算机或特殊用途的硬件上执行以实现本发明实施例中所描述的所有方法或功能。“机器可读媒介”还应包括固态存储器、光盘、磁盘、磁介质存储器、光存储器、和载波信号,但并不局限于上述媒介。同时,上述可由计算机实现,但也不排除由人工来实现。

[0180] 举例来说,机器可读媒介可包括只读存储器(read-only memory, ROM);所有类型的随机存储器(random access memory, RAM),如 S-RAM、D-RAM;可编程只读存储器(programmable read only memory, PROM);电可擦除只读存储器(electronically alterable read only memory, EPROM);磁随机存储器;磁盘存储介质;闪存;电、光、声或其他形式的可传播信号(如,载波、红外信号、或数字信号)。但上述机器可读媒介不局限于上述的各种具体形式。

[0181] 此外,在本领域中,术语“软件”也可以有其他的称呼,如程序、进程、过程、应用、模块、算法或逻辑等,其描述了执行某一动作或产生某一结果的行为。本领域普通技术人员可知,上述表达方法仅仅是为了方便描述软件在计算机或类似器件上的执行,这种执行使得计算机或类似设备完成一个动作或产生一个产品或是结果。

[0182] 当描述某一具体例子时,在该例子中可能包括特定的特征或结构,但不是每个例子中都包括特定的特征或结构。但这并不意味着两个或多个例子中的特征或结构必然或可能不能相互结合,除非在文中已明确说明排除了这种结合。对于在文中将某特定的特征或结构与某一例子相互联系时,本领域普通技术人员可以容易的将所述特征或结构与其他有或没有明确描述这种联系的例子相互联系起来,并获得某些效果。

[0183] 本具体实施例中所所述的连接器或连接器接口(如存储器模块连接器接口),并不局限于物理上分开的接口(如公连接器或接口啮合与母连接器或接口啮合)。连接器接口还包括任何物理形式的接口或连接,例如来自存储器模块的接口其导线、焊球或连接均可焊接到电路板上。例如,在叠式冲模上,多个集成电路冲模逐层堆叠(如存储设备及缓存设备),其底层形成基底,则该冲模与存储控制器或处理器通过一个球栅阵列类型的连接器接口相接。又例如,存储设备和缓存可能经由柔性带状互连和接口与存储控制器通过一个球栅阵列类型的连接器接口或是物理上可分开的接插件类型的连接器接口相互接。连接类型还包括集成电路片和基底上互连的导体、基底、印刷电路板、或其他类似物之间的接口。

[0184] 存储模块可包括存储器,还可具有控制存储器的控制器、可配置的开关元件(configurable switching element, CSE)和其他用于处理、传送或接收信号的电路。当然,存储模块也可能只包括存储器,其他的单个或多个部件可能分开安装。除非是特别指明,特定模块的功能的布置是为了讨论的方便,本领域的普通技术人员可知可以有多种等价的方式对实际的物理和计算机的各方面进行安排。

[0185] “总线”或“链路”表示单条信号线路或多个信号线路,其中每一线路有一个或多个用于“传输”(发送、接收或两者)的连接点。每一个连接点可连接或耦合到一个传输接收器(如,传送器-接收器)或一个单独的传送器或接收器电路上。。其连接或耦合形式可以是电的、光的、磁的、量子缠结的、或其他类似的形式。使用相同或类似的方法形成的其他的电连接可用来满足如能源、接地、辅助信号和控制、或类似的其他附加系统的需求。有时也会描述这些附加的连接以使整个例子的描述更加清楚。这些附加的连接对于本领域普通技术人员来说是显而易见的,即使在具体实施例中沒有这些连接的描述也不应认为在该实施例中沒有这样的连接。

[0186] 一个链路在信号线路上传输信号。信号可分成如下类型:时钟和控制信号、地址信号、命令信号和数据信号。数据信号携带从存储设备或接口中存储或取得的数据。地址信号指明数据要读取或写入的在存储设备或系统中的位置或是位置范围,同时,地址信号需要选择要访问哪个或哪组存储设备或接口。命令信号指明存储设备或接口需要完成的操作类型,如,读出、写入、刷新、也可能是要使用哪几种访问模式(如脉冲或广播模式)传输数据。时钟和控制信号用于对在控制器和存储器之间传送的其他信号进行同步。尽管链路可以使用不同的信号线路传送不同的信号(如,在一个时钟周期内用 32 路地址线路传送 32 位地址,在一个时钟周期内用 32 路数据线路传送 32 位数据),亦存在各种设计来复用一个或多个信号线路用于不同的信号,也存在各种设计来复用一个或多个信号线路用于在一个

存储事务中的不同时钟周期内传输命令信号、控制信号、地址信号和数据信号。

[0187] 需要注意的是,当讨论连接至接口或端口的线路时,为了方便起见通常会描述该线路是活动、被上电的或关电的、处于待机模式的、或类似的。上述描述是指在接口处与节点的端口的连接状态。线路本身可能只是用于提供节点间的连接的电路板上的金属导线或类似物。

[0188] 树结构中的节点的数字标注经常使用“幅度第一”方法,该方法是指:自根节点开始,首先命名同一层的所有节点再命名下一层的节点。这个标注原则仅仅是为了方便。在实际中也可以使用不同的数字标注方法,或者同时使用两种或多种标注方法。在信号领域中,“端口 0,端口 1,等等”这样的标注通常意味着这些端口之间是有关的而不是相互独立的端口。通常,“南向”、“下行”、“次级”是指远离模块控制器或根节点的方向,而“北向”、“上行”、“初级”是指靠近模块控制器或根节点的方向。当有多个模块控制器或根节点时,可对各模块控制器和根节点同时进行操作。

[0189] 如图 1 所示,为本发明的具体实施例中的一些符号和术语的意思的示意图。这是为了描述和理解起来更方便、清楚和简明,包括其他在本发明具体实施例中将要用到的类似的术语都是本领域普通技术人员所公知的。在本具体实施例的以下描述中除非特别的规定还将用到此处提出的术语。但是,上述说明不代表要排除或区别类似的已被认可的术语或是可能将要使用的表示相同概念的术语的使用。

[0190] 模块控制器 MC 10(其可能是独立的模块控制器或存储控制器,也可能是其他器件中的接口或电路,其中上述其他器件可能是中央处理单元 (central processing unit, CPU)、数字信号处理器 (digital signal processor, DSP)、微处理器、图形协处理器,桥接芯片、或其他可用来与互连系统相接的类似器件,上述具体器件都没有在附图中显示)的功能可能由一个器件完成,也可能由一个以上的器件完成。上述控制器与一组模块 M#20(或其他计算机系统器件,如接口、其他处理器、存储器,或类似器件)通过点对点链路相连。上述术语“模块”或“存储器模块”也可以用来代表其他包括接口和处理器在内的计算机系统组件。

[0191] 如图 1 中所示,存在一或多个多支路总线用于控制和信号线路;同时还存在一个或多个全局时钟信号,该信号线路和数据线路有可能是单向的,也可能双向的。在 MC 10 和 M#(M0, M1, M2, M3) 之间以及从一个模块到另外模块的的链路可同时包括数据和控制/信号线路。这些链路可以有多种不同的配置方法,图 1 中为一些可能使用的配置方法。

[0192] 以下介绍图中的各个线路组中的具体线路:

[0193] L0 包括:分开的数据线路 30 和控制线路 40,其中,每一路分别是具有相同宽度的北向或南向的单向线路;

[0194] L1 包括:分开的具有相同宽度的北向或南向的单向数据线路 30a、30b,以及从 M1 到 M0 的北向控制线路 40a,从 MC 到 M1 的南向控制线路 40b,从 M1 到 MC 的北向数据线路 30c;

[0195] L2 包括:在 M1 和 M2 之间的北向和南向的双向数据线路 30d 和控制线路 40c,以及单独的在 M1 和 M2 之间的双向控制线路 40d,从 M1 到 M3 的北向的控制线路 40e;

[0196] L3 包括:从 M3 到 M2 的组的北向数据线路 30e 和控制线路 40f,在 MC 和 M3 之间的双向控制线路 40g,从 M2 到 M3 的南向数据线路 30f 和北向数据线路 30g。

[0197] 跳过邻接模块的控制线路传输控制信号时会比依次顺序经过每个模块传输控制信号具有更短的延迟时间,如从 MC 到 M1 (40b)、M1 到 M3 (40e),数据线路的情况也类似,如从 M1 到 MC (30c)。图中标有“数据”、“控制信号”、“时钟信号”的线路“主要”仅为数据或控制线路,这些线路也可能用来传输其他类型的信号。比如,标有数据的线路可能传输控制信号,数据信号可能在控制信号线路上传输。如上所述,数据和控制线路这种说法是逻辑上或物理上依赖于其具体传输内容的。

[0198] 由时钟产生电路产生的时钟信号可能直接分别发送至不同的组件和模块,该时钟信号的传输使用单向的时钟线路或是连接在两个或多个模块之间的不同的时钟。高速时钟信号也可以与数据或控制信号组合在一起。

[0199] 如图 2 所示,为可配置的开关元件 (Configurable Switching Element, CSE) 15 的一个具体实施例,所述 CSE 15 可以有一个以上次级或下行端口。CSE 可用于与存储器或其他器件通讯;上述存储器或其他器件可以与 CSE 位于主板上相同的物理模块内,也可与 CSE 分别处于不同模块上,也可以在其他地方。两端带有箭头的链路表示双向的数据通路,该通路可能是分开的单向链路,或是双向链路,或是逻辑上的双向链路(如由两个单向连接通过环状模式形成的逻辑上的双向链路)。链路可以为串行或并行配置,或是串并行的组合,其方向可以是单向的或是其他的形式。

[0200] CSE 可具有不受限制的将任何输入端口和任何输出端口相连接的能力。为了方便逻辑上的描述,在以下的描述中,可将端口看成北向或南向的,当然,这样的描述不应作为对 CSE 的端口之间通讯能力的限制。比如,在一个 CSE 中北向端口可以和南向端口通讯,而在有多根节点树结构连接中南向端口也可以作为北向端口工作。当然,对于某一特定配置,不需要同时实现端口的上述所有功能。

[0201] 如图 3 所示,为本发明中 CSE 15 的另一具体实施例。所述 CSE 可以有不止一个次级或下行端口,也可以有不止一个上行或初级端口。此外,所述 CSE 可以有一个或多个内部存储器以补充或替代外部存储器,而 CSE 可以与除了存储器模块或存储系统以外的器件进行外部连接,上述器件包括输入输出设备、协处理器以及类似器件。如图所示,CSE 可以有多个次级端口,这样 CSE 可以实现将信号分离成一路以上的通路的功能。此处 CSE 也可以作为具有南向和北向端口组的转发器。此处使用术语“内部”和“外部”表示不同的存储器时,仅仅是为了描述上的方便,不应作为一种对上述器件的限制。在微处理器或特殊功能硬件中越来越多的使用了存储电路,内部和外部这样的术语的使用是为了帮助读者更好的理解本发明中的具体实施例。本领域的普通技术人员能够理解,电路的组合或分离、可拆成不同的物理组件的能力(如可以被集成为相同的封装或设置在相同的底层上,及类似的)均是由具体的应用和相关的技术状态所决定的。

[0202] 如图 4 所示,为 CSE 的另一具体实施例,其中,在一个系统中该 CSE 可以与不止一个模块控制器连接,并且没有数据或控制流的“上行”或“下行”方向的区别。CSE 可有多种不同的内部处理功能(或者内部结合有或与具有不同处理功能的器件进行协作操作、),所述器件如,微处理器、直接存取存储器 (direct-memory access, DMA)、作为模块控制器用于控制其他 CSE 的 CSE,同时,CSE 还与除存储系统以外的外部设备相连,如输入输出 (I/O) 设备、微处理器 (microprocessors, uP),图形处理器 (graphics processors, GPU),协处理器、其他 CSE 等。术语“初级”和“次级”的使用是为了描述的方便。当 CSE 中包含微处理器

(或处于微处理器之间,或与微处理器耦接)时,这种 CSE 可以作为处理单元也可以作为切换开关使用。

[0203] 如图 5 所示,为由 6 个存储器模块组成的树形结构,上述存储器模块中包含有与存储器关联的 CSE,在根节点具有一个存储控制器 MC。为了方便起见,图中没有明确标示出 CSE,包含在存储器模块 M# 中的 CSE 标识为 CSE#。存储器模块可能同时包含存储器和 CSE;当然,也有仅包含 CSE 的存储器模块。在本配置中,存储器模块 M0 与 M1 和 M2 相连,而 M1 与 M3 相连,M3 的下一级连接到 M7。在这种连接方式中,最远端的模块的跳点数为 4(M0- > M1- > M3- > M7),如果 8 个存储器模块以线性连接或菊花链方式排列,则最大跳点数为 8。一个跳点是指网络中连接两个连续节点的链路或线路。当讨论网络的节点间的信号传输的时间延迟或时差时,可使用跳点作为描述符。

[0204] 如图 6 所示,图中 8 个存储器模块连接成二叉树形结构,可将模块排布在 PC 主板上,并使得模块在物理上成线性排列。需要注意的是,其中一些链路,如 M0 和 M1 之间、M2 和 M6 之间的链路穿过中间模块 M4 和 M5,但并没有与 M4 和 M5 连接。当链路穿过一个模块但未与其连接时,在图中将链路穿过该模块的部分表示成阴影来指明这种链路的情况。如图 6 中所示的排布方法表明,诸如存储器模块或其他模块的大量的树形结构的各元件可以有组织的形式(如物理上成线性排列的结构)排布在电路板上或以插入式的方法将模块排布在主板上。上述排布方法有利于主板的布局,或有利于模块的冷却。当主板的尺寸由于其他原因(如主板所在的设备的尺寸)受到限制时,线性和非线性排布方法的组合也可以有效的用于在主板上排布模块。

[0205] 主板可用于通过替换出现故障的存储器模块来帮助存储系统的修复。是否需要主板取决于存储器模块的故障率、平均修复时间(mean-time-to-repair, MTTR)的分析、不可复原的数据丢失的概率、和其他各种因素。

[0206] 在本发明实施例中,有许多描述元件如何构成二叉树结构例子,但这些例子不应作为对连接的各种拓扑结构的限制。在例子中还有超过一个的入口节点或“根节点”的树结构,而且其中命令和数据信号会沿着非“树”形通路传输,如沿着环形结构传输。在树结构的例子中对各种情况的描述可以更加清晰,并且可以使如对信号路的反转之类的描述更简洁。但是这并不意味着排除其他的解决方案,如使用冲突解决、队列、存储和转送技术。对于其他网络结构来说,上述排布方法也是可以适用的。此外,可以采用并行方式对多个独立的存储控制器进行使用和控制。

[0207] 需要理解的是,当在图中出现一定数目的存储器模块,讨论一定数目的存储器模块时,并不意味着用图示方法连接的存储器模块或其他器件必须限定为上述数目,在图中使用某一特定的数目的元件仅仅是为了方便描述和作图。

[0208] 如图 7 所示,为一个由 6 个存储器模块连接而成的二叉树排布示意图,这种排布方式可以以非线性的形式在 PC 板上实现。这种排布方法在基于某一特定设计的物理要求时可能是方便的。

[0209] 如图 8 所示,为一个由 7 个存储器模块连接而成的三叉树排布的示意图,该排布方式可以以非线性的形式在 PC 板上实现。如图所示,使用具有两个以上次级端口的 CSE 可以使系统实现三叉树结构,当然,并不需要将树的所有可能存在的节点都与设备连接。

[0210] 为了简单起见,下述的例子都只讨论二叉树结构,但是还可以有三叉树(如图 8 所

示), 四叉树, 甚至更高分叉的树的结构。术语“树”可表示包括有任意分叉的树。当使用术语“树”时, 如无特殊说明, 也不排除树结构中的部分或全部模块使用线形或菊花链的排布, 同时, 术语“二叉树”表示的是有两个或两个以上分叉的树的结构。如前面所提及的, 在例子中的树的用途并不意味着要将所涉及的器件或方法仅仅限制于这些例子中的用途, 所述的器件或方法可以普遍的适用于各种网络中。

[0211] 如图 9 所示, 为一个由 8 个存储器模块连接而成的二叉树的排布示意图, 该排布可以以非线性的方法布置在 PC 板上, 其中存储控制器 MC 还同时与树的第二层的模块 (即 M1 和 M2) 相连。这种排布方法可以减少 MC 和存储器模块之间的通讯反应时间, 同时还可以将原来需要通过 M0 传输的信号通过其他模块传输来降低 M0 的功耗。这样的结构由于在透过由 MC 到树的第二层 (M1 和 M2) 以及由 MC 到第一层 (M0) 的连接链路或线路上发布命令、接收响应、发出和接收数据, 从而便利地增加了网络的吞吐量。

[0212] 如图 10 所示, 为由 8 个存储器模块连接组成的二叉树结构的排布示意图, 该排布可以以非线性的方法配置在 PC 板上, 其中该二叉树中的模块还与其他器件或模块相连, 如 M4 与图形处理器 (GPU) 相连、M5 与网络 I/O 设备相连。上述其他器件或模块还可分别地与内部或外部存储器 (GM、NM) 相连, 上述存储器可各自被直接访问, 也可被 MC 通过这些模块上的 CSE 进行访问。通过提供到与 GPU 相连的存储器模块 GM 的直接访问通路, 并允许对 GM 进行互不冲突的多路访问, 也可进一步提高网络的吞吐量。

[0213] 作为对比, 如图 11a 到 11f 所示, 为类似 INTEL 公司开发的一种存储系统, 该系统为全缓冲的双列直插内存模块 (fully-buffered Dual Inline Memory Module, FB-DIMM) 系统。以下是基于 INTEL 公司公开发表的该系统的说明文件对该系统做出的描述; 但是该系统的细节部分 INTEL 公司并未公布, 因此, 不保证图 11 中的 INTEL 公司生产的器件在 INTEL 公司生产的存储器系统中是确实如本文所述般工作。为了便于作为一个比较的基准, 图 11 中描述的分布是预期的。在每一个存储器模块 m 200 中都包括一个设备, 其接口特征与 INTEL 公司已公布的用于其高级存储缓冲器 (advanced memory buffer, AMB) 中的接口特征相同。

[0214] 为了简化对其功能的描述, 此处所用的读命令和其响应的表示方法与 INTEL 公司所使用的表示方法不同。例如, 在 INTEL 的 AMB 中, “读”命令包括几条消息, 而此处则表示为一条消息。另外, AMB 可以将一个单独的包含命令的“帧”发送给一个以上的 AMB, 举例来说, 对于一些命令, 包含这些命令的“帧”当其在存储控制器 mc 300 和存储器模块 mk 之间传输时并不只被某一特定的模块接收, 其中每一个存储器模块都有包含 AMB。

[0215] 在图 11a 中, 存储控制器 300 将读命令作为一个地址 A_i 和一个读命令 R 进行发布, 并作为一个“信息包”发送至下行通路。在所给的例子中, 数据的地址区在存储器模块 m_i 中。如图 11b 所示, 信息包到达存储器模块 m_i 200i。在 FB-DIMM 系统中, 信息包通过线性链路重复地从一个模块快速传至下一个模块。因此, 如图 11c 所示, 即使读命令只要传到存储器模块 m_i , 但其信息包还是要进一步的传到存储器模块 m_k 200k。因此, 包含读命令的每一个信息包都要传到存储器模块 200 所链路的所有模块。存储器模块 m_i 200i 的响应表明, 已从地址 A_i 处读取了数据并将数据比特或数据包通过上行通道回传 (如图 11d 中所示, 本例中的数据比特为 D0-D3)。为了方便, 仅显示图中所示的数据比特的数目, 实际传输的比特可能比图中的多也可能比图中的少。如图 11e 和图 11f 所示, 数据通过存储器模块 m_0 后被

存储控制器 mc 接收。被所有模块传送的读取数据包都经过了除了去歪斜,去连续化,或解码操作之外的相同的处理。

[0216] 如图 12a-12f 所示,为对照图 11 中基线组成的线性方式的一系列存储器模块 M# 的示意图;但是,图 12 中包括 CSE 的存储器模块可以“侦听”或读取包含在命令包中的地址,即使该包并不计划发送给该特定的 CSE 或其相关的存储器。每一 CSE 可以处于一其上行线路为“降低的”功率调整的状态,用虚线表示。可降低的能量的对象和方法包括但不限于,低功耗的 I/O 设备、I/O 逻辑时钟的门控(或任何其他相关的可被停止的逻辑)、降低 I/O 逻辑的时钟速率、降低 I/O 逻辑的电压、将当前输入装载到可降低漏电流的 I/O 逻辑的 I/O 逻辑、或任何其他可用于降低芯片的任何部分的功耗的同时还可以撤销这种低功耗状态以处理返回的数据的方法。在图 12d-12f 中,上行链路在数据返回之前恢复正常状态(以下称开电状态),当数据传输完成后又变成“降低的”功率状态。

[0217] 在本例中,控制信号在数据线路上以信息包的形式传输,当数据在线路上传输时该线路处于开电状态。一方面,信息包在“帧”或“时隙”定时系统中传输,当每一个“帧”或“时隙”计时开始时下行通路则处于开电状态,当有信息包需要发送时,发送方保持开电状态并将包发送出去;否则,发送方保持关电状态直到下一个时隙时间开始,接收方将保持开电直到确定信息包已经接收完成;如果没有信息包要接收,接收方保持关电状态直到下一个时隙时间开始。

[0218] 如图 12a 所示,为 MC 10 发布对地址 Ak 的读命令 R 的情形示意图。在图 12b 中,读信号 R 所在的数据包到达非预期的存储器模块 Mi 20i,在图 12c 中,该数据包传到了存储器模块 Mk。Mk 执行读命令 R 的结果是将数据 D0-D3 向上行通路传输(如图 12d 所示),数据经过 Mi(如图 12e 所示)后被存储控制器 MC 接收(如图 12f 所示)。按图 12c 到 12f 的顺序,每一个连续的上行链路在该链路上要传输的数据到达之前转换到开电状态,而当读取的数据 D0-D3 通过该链路后转换到关电状态。在现有的排列中,当控制信号和数据在下行通路的数据线路上传输时,所述下行链路始终处于开电状态,

[0219] 当数据读取命令 R 信息包沿下行通路从 M0 到 Mi 再到 Mk 时,每一个存储器模块 M 观察或“侦听”到该信息包,并获知目的地和该命令的种类;在本例中,上述命令为从 Mk 中的一个地址读取数据。这个读数据的业务被认为是沿上行通路从 Mk 到 MC。因此,在模块(该模块即为读命令的目的模块)和 MC 之间的每一个上行通路的链路将在一个合适的未来时间处于开电状态(如图中从虚线到实线的转变)以传送该读取的数据,当该读取的数据从该链路上传出时,该链路可被转换成一个低功耗状态或关掉。此外,当读数据命令 R 的传送目的地不是链路中的最后一个模块时,仅有目的模块(如 Mi)和 MC 之间的上行连接需要在合适的未来时间转换成开电状态以降低功耗。

[0220] 如图 13a-13f 所示,为包括 CSE 的存储器模块 M 的示意图,其中各模块间有分开的命令和数据线路。图中所示的信号和命令线路可能仅仅是数据线路的一部分,而不是专用的信号线路。其中,下行线路和与之关联的逻辑电路的一部分处于降低的功率状态。当命令控制信号通过每一个模块时,信号被解码,同时,如果需要的话,其他的下行线路转成开电状态以传送跟随在数据包后的数据或命令。如图所示的情况为,读命令 R 发往存储器模块 Mi 中的一个地址 Ai,并且,所述读命令 R 和地址数据 Ai 在不同的线路上传输。地址数据 Ai 表明目的地地址或地址范围是处于存储器模块 Mi 中的。因为,地址数据包 Ai 早于读命

令 R 发送,因此,如图 13a 和 13b 中所示,地址数据包 A_i 比读命令 R 早到达每一个存储器模块 $M_{\#}$,同时,地址数据可用于使发送模块 M_i 和接收模块 M_{i+1} 之间链路转成开电状态为传送和处理将要到达的命令信号做准备。与图 12 中的描述相似,当命令信号被接收后,下行命令通路将重新恢复关电状态,作为命令的执行结果,上行链路将为传输读取的数据及时地转成活动状态。

[0221] 因而,可以说线路的功耗状态在将来时间内是可改变的,这种改变是基于被 CSE 接收的控制或命令信号、地址或数据信号而发生的。该将来时间的状态的改变的时间值可能是由接收到的命令的特征、目的地的地址、网络中接收信号的 CSE 的相对位置、相似或衍生出的信息决定的,或是由 CSE 的部分或所有线路产生的。

[0222] 另一方面,如图 13b 和 13c 所示,命令信号可能被该命令的目的地模块(本例中为 M_i) 改变,此时,无操作(可简称为“no op”)命令可通过下行线路发送至下行所有模块,并表示在下一帧、时钟周期、或是可作为命令间间隔的时间片段内没有命令将会被发送至这些模块。如本例中,在图 13c 中,没有出现读命令,而地址数据 A_i 则被目的地模块的下一级模块 M_k 接收到。这样, M_i 和 M_k 之间的下行命令线路不会被激活。图 13d-f 中的数据返回情况如图 12d-f 中的类似,此处不再一一描述。

[0223] 存储器模块(实际上是存储器模块 20 中的 CSE 15)可对通过它们的每一个地址数据包进行解码,而不对命令包进行解码,因此当地址数据包和命令包作为信号向下一级传送时,上述由于解码造成的额外的时间延迟,会使地址数据包和命令包之间的时间间隔减少(考虑到,地址数据包先于命令包发送)。在地址包和命令包在同一数据线路上传输的情况下,第一个包结束的后沿和第二个数据包结束的前沿的之间有一定时间间隔以解决对第一个包的额外的处理造成的时间延迟对下一个包传输的影响。上述处理在某些情况下是需要的,但并不表示该例子中的带宽会因此受到损失。

[0224] 在另一情况下,存储器模块在对地址数据进行下行传输的同时对其进行解码,这样将不存在解码造成的时间延迟。在一些情况下,这将使得命令信号早于实际需要的时间在线路上传输,但取决于实际的配置与该线路的另一端接收该命令有关的 CSE 可能并没有被激活。

[0225] 如图 14a-f 中所示,为与图 12 和 13 类似的配置情况,但是在之前的例子中,相对于当前模块目的地模块可以是上行的模块也可以是下行的模块。这使得每个模块需要将接收到的地址与许多模块的地址进行比较,取决于寻址逻辑的复杂度,上述寻址过程需要几个时钟周期。这个处理时间将增加 MC 传送地址和命令信号之间的时间间隔。通过修改目的模块 M_i 接收的地址 A_i ,使 A_i 先于地址数据传送到下一级模块 M_k ,可以减少上述增加的时间间隔,这样一个新的地址 AX 可作为地址和命令已经被适当的模块 M 解释的标记。AX 可以作为保留地址或其他相似的意思。

[0226] 这样,每一个模块只需要比较接收到的地址 A_i 和模块 M_i 本身的地址或地址范围或保留地址即可,而不需要将地址 A_i 与系统中所有的模块的地址进行比较。在其他方面,图 14 和图 12 及图 13 类似,细节此处不再一一描述。

[0227] 高速数据链路可使用时钟恢复机制来保持同步。如果在每单位时间内没有收到足够的比特数,链路的时钟同步将丢失,将耗费一定的时间进行同步的恢复,这会降低吞吐量,或是导致数据丢失而造成数据重传。为了保持同步,诸如 FB-DIMM 这样的系统将会周期

性地传输包括这样的变换的同步帧（图中表示为“sync”）至所有的模块。

[0228] 然而，在拥有许多存储器模块的配置中，这样的同步帧将会传送至每一个模块，因此会经过每一条线路，或是将多重同步帧传至树的不同叶子节点。如果由模块组成的树足够大，上述保持时间同步的方法将占用系统的相当一部分的带宽，而由于需要进行时钟同步往往会限制系统的规模和复杂度，除非对时钟进行刷新或采用其他办法保持时钟同步。

[0229] 解决上述问题的另一个方法是为每一个模块提供定时器，当有足够数目的传输发生时该定时器可以记录已用的时间间隔，并在局部发起时钟重新同步化。计数器或看门狗为上述定时器的一种例子。

[0230] 如图 15，表示了从 MC 到模块 M0 传输读信号的过程。读信号 R 发送至所有模块，而命令线路没有处于低功耗状态。在此情形下，当预定的时间间隔已经过去而线路上没有被用于传输数据时，该模块开始数据的传输，模块发送“noop”数据包以继续保持该线路的同步。如图 15d 所示，在一个帧时间内，M0 确定不需要使用 M0 到 Mi 方向的链路发送信息包，则使用该时机，可以为了保持模块 Mi 的接收同步在该链路的线路上传输变换信号。类似的，当 Mk 确定在某该帧时间内不需要使用 Mk 到 Mi 方向的线路发送信息包时，可在该帧时间内传送变换信号。

[0231] 进一步的，如果链路是半双工或全双工的，则该链路或其中线路可以转向，这样数据可以在模块间通路上传输而不需高业务密度。这种方法取决于接收到的命令信息，以便模块可以确定没有数据将来自其他的方向并在半双工链路上部分或全部的执行转向。如图 15e-f 所示，模块 Mi 和 Mk 转变了 Mk 到 Mi 方向（如图 15e 中所示）的链路，然后在图 15f 中，Mi 发送变换信号以保持 Mk 中的接收机同步。带箭头的线段表示各链路，而箭头方向表示了当前数据流的方向，一个链路可能具有一个或多个传输数据、命令和地址数据的线路，上述链路的方向可以反向，如在图 15f 和图 15d 中，Mi 和 Mk 之间的链路就改变了方向。在本描述中，传输方向可改变这种功能只是一种可能而不是必须的，这取决于在实际应用中的需要。

[0232] 如图 16 所示，为一个模块控制器 MC 和模块 M0，……，M6 组成的树形结构示意图，该树型结构具有点到点的链路。在上述树结构中，包含地址的数据包可向下一级发送，每一个切换开关（如 CSE）或模块（如存储器模块或其他与 CSE 相结合的模块）对地址和命令信号解码及确定相应的数据或命令被定向到何处。然而，上述处理在下述步骤上需要消耗时间：接收信号，与目的地地址比较，确定将要进行的操作。当地址在先于命令或数据信号被发送前的某一时间发送并使用了一系列的链路时，由于地址计算时需要进行数据处理，这将减少地址和其后的数据之间在相邻两级传输时的时间间隔。上述过程使用的时间比仅仅传输数据而不对其进行处理所用的时间长。并行执行数据处理功能（如将数据往不需要的链路上发送）会额外增加复杂性。是否需要进行并行处理取决于整个网络所需要的带宽。

[0233] 另一个需要考虑的问题是每一个数据包里的数据比特的时间分散。在有许多链路的情况下，数据的各比特可以在并行的信号线路上传输。上述情况可能只是实际情况的简单描述，比如说，数据可以在并行线路上传输，每一条线路上可以以串行的方式传输一个以上比特。由于线路长度和阻抗等其他因素的影响造成数据在线路上有差分延迟时间，因此，对于需要实时处理的数据比特，该比特可能不能足够同步到达，而需要等待一个或多个时钟周期直到模块接收到所有的比特，再进行解码或其他数据处理。该延迟还增加了每一个

节点在处理地址或其他信息之前的必要时间段。在一些情况下,如仅当需要进行数据传输时才将线路的状态转成开电状态的情况下,地址信息可能要比命令或数据明显早一段时间发送。这将会降低系统的传输能力或带宽。

[0234] 图 16 与图 12 类似,但是图 16 中模块以树结构相连,图中只显示了相关的通路。如图 16a 中所示,MC 向处于模块 M5 中的地址发送读命令 R。如图 16b 所示,M0 向 M2 发送地址命令和读数据命令,而不需要向 M1 及其他与 M1 相关的模块发送上述命令(但是如下所述,将在随后发送“no op”命令)。如图 16c 所示,由位于 M2 内本地 CSE 做出决定并将地址 A5 和读命令 R 向 M5 传送。如图 16a-c 中所示,比较了地址包 A5 和读命令 R 之间的空隙,图中定性的表示了两个包之间的时间间隔不断缩短的情况,在图 16c 中,当 M5 接收到地址和命令数据包时,上述两个包之间的时间间隔可能是最小的。这相当于下述情况,即当 MC 发送两个包时设定它们之间的时间间隔,使得经过参与传输的模块的解码延迟后整个时间间隔被抵消掉。M5 对读命令 R 的响应的过程(如图 16d 所示)如前所述,本处不再对其细节一一描述。

[0235] 如图 17a-d 所示,为如图 16 类似的排布,但是在如图 16 所述的地址或数据线路的基础上,在图 17 中命令信号使用单独的信号线路。由于命令和地址或数据信号使用不同的线路传输,将可能更有效的利用地址和数据线路的带宽。为了简单起见,假定在每两个模块之间的地址信号的传输延迟相同,则可以调整读命令的传输,使得地址和数据以预定的时间间隔到达目的模块,在本例中目的模块为 M5。当读命令序列连续发送时,上述设置将是有优势的。

[0236] 如前例,地址链路和命令链路在即将被使用前的一段合适的将来时间可转换成开电状态,而当链路上的信息包传输完成后转换成关电状态。本领域的普通技术人员可以了解到,允许每一条线路在使用完成后保持一段时间处于开电状态以调节附近距离的信息包将会得到一些有益的效果。这样的一段时间可是时钟周期的倍数。

[0237] 如图 18 所示,为使用地址指针实现路由的命令的一个例子。本例中的模块排布的拓扑结构与图 17 一样,但是,为了清楚起见,本图中仅显示命令线路,其线路的开电和关电状态的转换如前所述。在本例中,地址命令的结构为自选路由包而不是目的地地址,该命令包括 3 个 2 比特字以及跟在该字后可被目的模块执行的本地命令。图中,地址字在细六边形框中,本地命令在粗六边形框中。另外,为了方便描述,每一个字可以视为有一个最低有效位(least significant bit,LSB)和最一个高有效位(most significant bit,MSB)。上述 4 个 2 比特字如同在两个并行线路上传输一样向 S0 传输。可选择的,该信息也可以在一条线路的通路上以串行比特的形式传输。

[0238] 每一个字中的比特的解释如图 18 中的表所示。注意到,一般的,每一个模块顺序标识为 S0-Sn,这意味着每一个节点可能仅仅是一个切换开关,或是对于布置在切换位置的其他电子元件的补充。上述切换开关可以是 CSE 或其他可以解释或响应命令的设备。

[0239] 在本例中,每一个切换模块 Sn 都有一个输入端口和两个输出端口,输出端口分别标识为“0”和“1”。地址命令按照目的地选择表进行解释,其中,命令中 2 比特字的值决定了要使用哪个输出端口。虽然图中只显示了 2 个输出端口,但并不意味着切换开关元件的输出端口的数目或是每一个地址命令字中的比特数目受此限制。同时,所用术语输出和输入端口以及上行和下行方向是为了描述的方便,并不排除在任何逻辑或物理输入和输出端

口之间的连接。

[0240] 地址包中的最后一个字包括一个本地命令。该命令为由目的切换开关执行的操作。图中,本地命令表给出了一些命令的具体例子。如,值为 1(二进制值为 01)表示选择 CSE,值为 2(二进制值为 10)表示选择本地存储器,值为 3(二进制值为 11)表示无操作命令(NOP 或“no op”)。如图 18a 所示,为输入到 S0 的地址命令包,其执行命令的目的地为 S5。第一位比特对(二进制值为 10;MSB,LSB)表示选择 S0 的输出端口 0。如图 18b 所示,数据从 S0 的输出端口输出,注意到,由于第一位字 1(二进制值为 10)已经在 S0 使用过,所以从地址命令中去除了。如此这样,每一个切换开关 S 可根据接收到的包中的第一位字确定需要执行的行为。每一个切换开关 S 解释的是字在包中的位置而不是第一位字的内容。为了简化处理过程,假定在上述格式下每一个切换开关 S_n 解释地址命令的时间是一样的,那么处理上述操作增加的延迟则与该地址命令的目的模块在树结构中的深度无关。

[0241] 当该地址包通过切换开关 S2 后,由于其第一位 2 比特字的值为 1(二进制值为 01),则应路由至 S2 的端口 0,然后将第一位字从包中再次去除。如图 18c 所示,为地址命令将被 S5 接收的情形。此时,第一位字值为 3(二进制值为 11),指示目的地为本地,即表明其后跟随的数据的目的地为本地。第一位字后跟随的字为本地将要执行的命令,在本例中值为 3(二进制值为 11),即为 NOP,表示目的地模块不需要向上行回传任何响应。

[0242] 如图 19 所示的另一个例子中,切换开关 S 将 NOP 发往一个或多个没有被地址命令选择的输出端口。这有助于时钟脉冲在整个树中以有效的方式传输以保持同步。输入到 S0 的地址命令的目的地仍是 S5,该命令包括本地存储器命令 2(二进制值为 10),如图 19b 所示,与图 19a 中所示的原始命令相比,S0 的端口 1 处的地址命令中去掉了原来的第一位字。同样的,没有被选择的端口 0 向与本端口相连的切换开关发送包括本地地址命令和 NOP 命令的信息包。如图 19c 所示,为 S1 接收到 NOP 命令后执行的操作。由于 S1 的两个输出端口都没有被选中来输出数据,则 S1 可向端口 0 和端口 1 输出包括本地地址命令和 NOP 命令的信息包,该包将发往 S3 和 S4。S2 执行的操作与之不同,其输入地址命令如图 19c 所示,该地址命令将发往端口 0(二进制值为 01),然后将其第一位字去除,并将剩下的命令包发往 S5。但是,在本例中,数据没有发往 S2 的端口 1,因此该端口向 S6 发出一个包括本地地址命令和 NOP 命令的包。树的末端上的每一个模块(称为“叶子”,在本例中为 S3, S4, S5, S6)可在其每一个输出端口发送一个本地命令和一个 NOP 命令,但是,如果没有与之连接的下行通路,则上述处理过程结束。模块亦可被配置成树的叶子模块,其输出端口可以设置成非活动状态,或省略输出端口。

[0243] 如图 20 所示,模块树可被配置以使链路通道化,使用如图 3 所示类型的 CSE,并允许两个或多个命令同期或同步执行,其中,同步执行的命令数目取决于分支度。在本例中描述了两个独立的数据包的传输情况。第一个数据包是作为读命令 R 的响应而从 M4 经 M1 和 M0 返回至 MC 的信息包。同时,第二个对读命令 R 响应的数据包从模块 M6,经 M2 和 M0,发往 MC。返回的数据包将使用连接中一些的线路,上述线路的数目少于模块间的链路的线路的总数。在一个连接中的各“通道”之间的线路的划分并不需要相同。

[0244] 如图 21 所示,为更进一步的树中通道化的示意图。此处,在模块 M3 和 M6 分别处理读信号 R 的同时,M4 和 M5 之间经由 M1、M0 和 M2 构成了一个存储器直接存取(direct memory access, DMA)传输。从模块 M4 到 M5 的数据传输和 M3、M6 发出的数据传输都以不

间断的相同的速率进行。在本例的配置下, DMA 传输可以传向或来自没有正在被访问的模块,也可以在可同时处理一个以上数据通路的模块上进行。这样,可以进行存储器直接存取操作而不影响其他同时进行的存取操作。当模块内有仲裁单元,或模块支持一个以上操作/访问同时进行(比如,模块内包括多端口存储器,或 CSE 可以对部分或所有的冲突操作进行缓冲)时,在从模块控制器向模块发送信号的同时,可在模块构成的树结构内进行其他操作,如 DMA。本例所述的 DMA 操作中还示意了下行端口和上行端口之间的连接,该连接是可在一个 CSE 内连接其逻辑或物理端口以进行数据传输的例子。

[0245] 如图 22 所示,为在树结构中同时或同期传输数据的一种方式(例如树结构)的示意图。在本例中,模块 M8 通过 M4、M1 和 M0 向 MC 发送读操作信号(这条路线在本图中表示为 M8 到 MC 的虚线),MC 通过 M0、M2 和 M5 向 M9 发送写操作信号(该路线在本图中表示为 MC 到 M9 的实线)。与读信号 R 有关的数据从模块 M5 经由 M2、M0 和 M1 传向 M4(如图中点虚线所示)。取决于特定的源和目的地,当树结构中的存储器间传输信号时自 MC 的存取也可能不是同时发生的。

[0246] 如图 23 所示,为具有多个处于低功率状态的连接(图中表示为虚线)的配置的示意图。当接收的地址命令表明下行方向将会有数据通过时,在本例中的读命令 R 经过的下行通路将转换至开电状态以传输命令。在图中表示为虚线到实线的转换,如图 23b 中 MC 到 M0 之间的连接变成了实线连接。如图 23c 中所示,MC 到 M0 的连接状态转换成了关电状态,而 M0 到 M2 和 M2 到 M5 之间的连接状态转换成了开电状态。如图 23d 所示, M5 到 M2 和 M2 到 M0 之间的线路在合适的时间转换成了开电状态以传输数据,该数据是响应读命令 R 而需传回的数据。

[0247] 如图 24 所示,其中配置与图 23 类似,但在图 24 中其上行通路可设置为低功率状态。一般来说,通路状态从很低的功耗状态转换成普通状态的时间要更长一些。当上行数据通路处于低功耗状态时图中用点虚线表示(上述状态转换可以是如开电状态或待机状态到关电状态或类似状态之间的转换。在上述相互转换的状态之间应该有总功耗的区别和变成完全工作状态的时间的区别,但是对于一特定的设计其总的时间或总的功率都是明确的)。如前所述,当某一特殊的命令通过各模块(本例中为 M0 和 M2)时,在 MC 和目的模块(本例中为 M5)之间的每一个通路都会知道(或侦听到)该命令的传输和其目的模块在树中的深度。可以将每一个上行连接按照一个预定的时间重新转成开电状态,使得当有数据在该连接上传输时,其处于可操作(开电)状态。图 24a-c 还示意了将 NOP 命令 N 传输至树结构中没有参与目的地址传输的其他分支的情况。

[0248] 对于命令传输时间受到限制的系统,比如时隙传输系统,或是在命令之间有最小时间间隔的系统,或是限制命令的具体发送时间的系统,还可以进一步降低功率。如图 24a-d 所示的例子,在一个时隙传输系统中,传输地址命令的连接可以在时隙开始时转换成开电状态,在图中以半虚线表示。如果不需要传输命令,则连接在下一个时隙开始前转换成关电状态。如果需要传输命令,则链路保持开电状态以传输命令。如图 24b-d 所示,在一个没有固定的命令起始时间的系统中,将 NOP 命令 N 发送至实际命令没有使用的链路上,以使该未使用连接处于关电状态直到另一个命令发送至该未使用链路时该链路可以再次转换成开电状态。

[0249] 对于具体的物理线路传输技术,线路的开电和关电需要传送器和线路末端的接收

器的共同作用才能实现。

[0250] 在关于地址命令的另一个例子中,可对命令进行扩展以实现命令的广播或组播传送。如图 25 所示,其中还包括了一个本地命令扩展表,表中包括的值为 0 广播命令(二进制值为 00),和扩展命令表。与扩展命令有关的字表示在图中虚线的六边形框中。虽然本图中该字处于地址包的最后一个字,但是实际上扩展命令的位置也可以与本图中的位置不同。该扩展命令包括值为 1 的读命令(二进制值为 01),值为 2 的写命令(二进制值为 10),值为 3 的广播开始命令(二进制值为 11)和值为 0 的广播转发命令(二进制值为 00)。在传送序列中,扩展命令可跟在本地命令字的后面。在本例中,根据地址命令中的第一个字指向 S0 的 0 端口,由于该端口与 S1 连接,所以会将一个地址命令发送至 S1。如图 25b、c 中所示,NOP 命令被发送至与 S0 的端口 1 相连的 S2,并一直传送到 S5 和 S6,上述端口 1 是未被选择的端口。对于发送至 S1 的包(如图 25b 所示),其第一个字表明 S1 就是命令的目的地,而本地命令值为 0(二进制值为 00)表明其为一个广播命令。切换开关 S1 将会根据广播命令产生一个新的信息。广播命令将从切换开关 S1 的端口 0 和端口 1 发往在树结构中处于 S1 的下一级并分别与 S1 的两个端口相连的 S3 和 S4。在广播命令中还有扩展命令,在本例中该扩展命令为广播转发命令(值为 0,二进制值为 00)。根据上述广播转发命令,S3 和 S4 将产生相似的命令并将其向树结构的下级传送。

[0251] 如图 26a、b 所示,为从 S1 的输入到树的叶子的控制的流向的细节的示意图。如图 26a 所示的配置可以用来向 / 自其他模块同步发送 / 接收少量的单独的状态或控制信号。

[0252] 如图 27a 所示,为自路由消息的一些可选用的配置的示意图。在需要非常高速传输的场合,线路到线路造成的歪斜可能是,仅当信息在单条的线路上传输时其解码没有显著的延迟,否则当信号在多条线路上传输时,由于需要等待其他一个或多条线路上的同一信息的其他部分都传到则解码可能会增加显著的延迟。此处描述了多种比特序列的形式,其形式具有多种不同的有用的特性。

[0253] 值得注意的是,让第一个切换开关配置消息的路由可以使消息的传输具有最小的延迟。在向处于树的根节点的第一个模块发送消息的特殊情况的例子中,其消息的头两个比特(从右往左读)为二进制值 01,其后为命令 CMD(可能是读命令 R,写命令 W,或其他命令)。在将消息发往最底层 N(如一个叶子节点)的情况的例子中,头两个比特的二进制值为 11,该消息后为通路指示 [PATH],上述指示可以是 N 比特长的通路信息,其中,该比特的形式可为 0 或 1,具体取决于要使用到目的地的通路中下一个分支或节点的输出端口的哪一个。通路指示后跟要发往某一模块的命令 CMD。如需将信息发往末级的上一级则头比特为 011,其通路指示长度为 N-1。同理,若要发往下一个最高级则头比特为 001。其通路指示的长度为 N-2。对于 N-3 级,其头比特为 010,去往其他各级的信息的设置类似。二进制序列 11 则指示通路指示比特序列开始。按照上述模式,要到达的级数每增加一级则头比特的长度增加大约一位,通路指示 [PATH] 则减少一位,因此,头比特和通路指示的总长度变化范围可能为 1 个比特。可见,消息的长度不会随其要到达的模块在树中的深度有大的变化。如图 27b 中所示,为发往 S9、S4、S1 和 S0 的消息的头比特的组成的示意图,其中模块的连接关系与图 26 相同。

[0254] 图 27a 中还示意了附加命令 CMD-TIME 和 CMD-EX 在信息中的位置。上述附加命令包含了命令应该在什么时间执行和 / 或数据在什么时间被返送回来的信息。上述时间可以

是绝对时间,或是相对于接收命令时的时间,或可以基于命令本身的特性进行的解释。

[0255] 如图 28a 所示,为适合在每一节点去除头比特的一种消息格式的示意图。在该格式中,到模块 0 的消息的头比特为 0,其后一个比特指示后面跟随的是一个命令还是 NOP 操作。到第一级的消息的头比特为 1,其后为端口选择比特 P,其表明选择的是端口 0 还是端口 1。格式 0N 表示其后跟随的是命令 (1),或需要发送 NOP (0) (0)。对于通路的下一级的模块的地址,每增加一级则增加一对比特,该比特指明了增加的深度和通路上的路由。在每一级,消息的头两个比特确定了消息是发往本地切换开关还是向前发送。当消息往树的下一层(级)发送时,则要去掉消息的头两个比特,消息则通过合适的端口向前发送。图中左边矩形框中的数字表示树中的深度或等级。

[0256] 如图 28b 所示,为发往 S9 的信息在树的不同级的组成的示意图,其中,当消息中的头比特使用过后将从该消息中删除。

[0257] 当总线(链路)的速度加快时,存在于同步总线上的数据和时钟或命令之间与同步相关的问题也随之增加了。其中的一个问题是,同一链路、线路的不同的末端之间可能会具有不同的时钟。如果使用并行数据传输技术,则链路中不同线路的有效物理连接的长度造成的影响也会不同,这将可能导致不同线路上传输的各比特的到达时间不同。这就是所谓的数据“歪斜”。一种解决数据歪斜的方法是在每两个相邻链路之间进行去歪斜,以对信息包头进行解释并在下一个相邻链路上传输信息包。但是,由于去歪斜过程至少需要与各节点的线路间的最长的差别延迟时间一样长的时间,所以这将显著的增加重新传输的延迟时间。其他的解决方案是不在每一个节点上进行去歪斜以避免增加额外的延迟,而是允许通路上存在歪斜,并在目的模块或节点的接收器上进行去歪斜。但是,由于需要处理从源节点(如存储控制器)一直累积到目的节点的不同长度的延迟,则即使数据包的一部分到的很早,累积的延迟也会显著的增加总的延迟。

[0258] 在由 INTEL 开发的 FB-DIMM 技术中,要求器件可以修正每跳高至 6 个单位时间间隔(Unit Interval, UI)的歪斜。术语 UI 是指传输一个信号的平均时间间隔,当时钟周期不是十分稳定或没有颤动时,可以在高速时钟速率的场合使用术语 UI 代替时钟周期。如果一个存储系统中有 8 个相邻的链路,在每个相邻链路上的歪斜为 6UI,则总的歪斜为 48UI。

[0259] 如果链路运行在名义上的时钟频率为 2GHz,数据传输率为 4Gbps 的情形,1UI = 250ps,则 48UI 的延迟歪斜时间等于 12 纳秒(ns)。这表明,通过某一条线路第一个到达的比特的到达时间和通过其他的某一条线路最后到达的比特的到达时间差为 12ns。第一个到达的比特需要存储在存储器内,然后对寄存器进行闭锁或移位直到后续比特到达,最后将所有比特送至去歪斜和读取数据的设备。

[0260] 一方面,此处描述的互连系统可提供在存在歪斜的环境中进行数据处理的方法和设备。在上述存在歪斜的环境中,甚至设备到设备之间的线路上的延迟不是固定的。上述数据处理的方式可减少延迟均衡、延迟排程、缓冲或类似的操作的需要。如图 29a 所示,为简化的从一设备(如模块控制器 MC)到另一设备(如切换开关 S、存储器模块 M、接口、其他处理器、或类似装置)的数据传输的例子的示意图。在本例中,使用串行器 SER 300 将一个宽字(如 16 位字)转成 4 个 4 位字。上述 16 位字 310 通过 4 条分开的线路各自经 4 个连续的高速时钟周期传输。数据在链路的另一端被接收,由去串行器 DES320 进行去串行,并重新排列成原始输入字的顺序。

[0261] 如图 29b 所示,为与上述类似的处理过程,但是在各链路上传输比特的速度不一样,并且有比特的到达时间之间差一个以上时钟周期。造成上述差分延迟的原因可能是,比如,模块之间线路的长度不同。如图所示,比特 1、3、5、7 早于其他比特到达,除非时间延迟造成的歪斜得到解决,否则上述四个比特可能会被当成在前传输的字。而剩下的比特则被当作其后的字,这将会使数据出现错误。

[0262] 一方面,如图 30 所示,为处理歪斜的一种方法和设备的示意图,具有时钟数据恢复模块 (clock data recovery, CDR) 的切换开关元件接收歪斜的数据。可选地,切换开关元件中也可以没有 CDR。去歪斜模块中具有对每路线路进行时间延迟装置,以使所有线路的延迟是相同的。然后,使用去串行器 (DES) 对去歪斜后的数据进行处理以恢复原始的传输字节,并使用 CSE 和其他的本地功能模块对数据进行处理。虽然各功能模块在图中是分别单独表示的,但是上述全部或部分模块可以在同一个电子器件内工作,或是集成在一个模块内。去歪斜模块和 CDR 也不一定要在同一个电路上。另外,如图 31 所示,可以将多路输出选择器 330 设置在 CDR 和去歪斜模块 340 的输出与 DES 320 和发送器 (TX) 350 之间,这样,输入的数据可以发送至 DES 320 或 / 和 TX 350。上述配置,使得数据可以根据需要发送至 DES 或 TX,而不需一定要被所有电路处理。

[0263] 如图 32 所示,为在 INTEL 的先进存储缓冲器 (AMB) 或其他类似装置中引起歪斜的原因的示意图,其中 AMB 在可插入连接器的扩展板上的模块内。上述布置被用于 FB-DIMM、注册的 DIMM、和类似元件上。如图所示,上行通路上的消息来自下行模块,该上行通路具有许多线路,其中线路 0 在左边,线路 4 在中间,其他线路布置类似。上行通路与连接器的前侧相接,在穿过 AMB 后,连接到连接器的后侧上与前侧类似排布的线路。注意到,插入板的前侧和后侧都有 AMB,但并不表示需要分开的 AMB 元件来完成其功能。另外,线路仅仅显示到 AMB 的边缘,但是如果 AMB 是在一个球形栅格阵列 (ball grid array, BGA) 封装或其他不仅为允许边缘连接的封装中时,上述线路实际上连接到了 AMB 的下面。线路从 AMB 连结至板子背面的连接器。线路在板子背面的排布顺序也是线路 0 在左边,线路 4 在中间。图中所示的线路可以代表板子上 AMB 和板子接口的连接器 (本图中没有显示出来) 之间的连接线迹。如图中所示线路的排布方式,使得有连接混合模块接口的连接线迹的主板可以不需要使用交叉连接线迹来连接一个以上的模块。从这个方面来说,主板上连续的连接器的插脚之间传播信号的时间对每一条线路来说是大约相等的。但在模块板上的情况与之不同。如图所示,线路 4 从连接器到 AMB 为短距离,线路 0 则要长些。

[0264] 为了对具体实施例进行更好的说明,使用标准尺寸的 DIMM 模块作为参考。该模块宽 140mm,从切换开关所在的中央往线路 0 方向的距离为 70mm。线路 4 的长度为 20mm,该线路为 AMB 和在连接器上与之相关的插脚之间的线路,在本例中,最短和最长线路迹的长度的最大差为 50mm,该长度差将会导致大约 250 ~ 350ps 的传输时间差。上述情况是一个实际存在的 DIMM 的情况的估计,其中,数据传输速度为 4GB/S,根据这个传输率仅由于上述长度差将会导致 1 或 2 个时钟周期的歪斜。

[0265] 对于实际存在的标准 DIMM 或存储器模块,此处使用的术语 DIMM 或存储器模块应当理解为表示任何可以安在电路板上的存储器类型,该电路板具有与其他模块电路或类似设备进行连接的接口。插脚或连接线迹的数目,电路板的尺寸,各类在 DIMM 上的存储器的容量或类型,均不局限于上述插脚数目,互连数目,尺寸,容量和类型。

[0266] 如图 33a, b 所示, 为可用于减少时间延迟的不同信号路由的排布方法的示意图。如图 33a 所示, 为连接线迹从 CSE 直接散开延伸到连接器的情况。如图 33b 所示, 较长的连接线迹布置在板的顶部, 而其他的信号线路则从中间开始布置。在这种布置下, 一部分的连接线迹要比在图 33a 中的连接线迹长, 但是不同信号组之间连接线迹长度的差别变小了, 比如线路 0, 1, 2, 3 之间或 4, 5, 6, 7 之间的差别可以变小。不同线路迹组长度之间的差别还是仍然存在的, 但是可以通过降低延迟的差别, 如图 33b 中对 N 个信号, 从 N 个差分延迟减少到 2 个差分延迟, 来简化延迟管理和信号歪斜的处理。

[0267] 如图 34 所示, 为从模块控制器 MC 传输信号的一个例子的示意图, 如图所示, 从左侧开始的线路 0-4 通过了五个模块。从 MC 发出的数据按照线路 0, 1, 2, 3, 4 的顺序标记为 A, B, C, D, E。对应每一个下行模块有一个表, 表中输入的数据的逻辑名字为“*I*”, 相邻模块之间的线路上的延迟为“*D*”, 从 MC 到当前模块的累积的线路延迟为“*T*”, 输出数据的名字为“*O*”。表中每一行对应的一条线路; 其中, 最上一行为线路 0 的各项参数值, 最下一行为线路 4 的各项参数值。在本例中, 一个延迟与一条线路有关, 不同线路的延迟可能不一样。延迟可以根据概念上的线路解释为差分延迟, 上述概念上的线路可以没有额外的延迟或由于前一模块或 MC 的发送器到接收模块的接收器所造成的延迟。

[0268] 在本例中, 假定模块之间的延迟的范围为 1 到 6 (任意单位), 对每个模块的输入和输出线路, 在从给定的线路号上输出到相同线路号的输入时, 赋以相同的延迟值, 与每个模块相关的输出数据的格式和输入数据的格式相同; A, B, C, D, E 分别在线路 0-4 上传输。从按照下行方向的模块的表的变化可以看出, 每一条线路的总的延迟是在 MC 和与表对应的模块之间的线路上的延迟的总和。比如, MC 和第五个模块 (M4) 之间的线路 0 上的总的延迟为 30 个单位, 但线路 4 上的总的延迟为 4 个单位。在本例中, 其他线路也存在时间延迟的中间累积。若延迟单位对应为时钟周期, 则线路 0 上的数据将落后线路 4 上的数据 25 个时钟周期到达 M4。以整数值给出延迟的时间, 以及令模块间的给定线路上的延迟相同仅仅是为了说明的方便。延迟时间并不一定是整数, 在相邻模块间的给定线路上的延迟时间也不一定相同。举例来说, 如果将要发送的命令或包或帧的长度为 10 个时钟周期, 则在任意时间中, 模块 4 都可能正在处理收到的 2.5 个命令。这样, 由于最后一个模块需要使用上述时间将一个命令的所有比特作为一个完整的组来接收, 所以最后一个模块在其可以处理第一个命令之前可能需要在任何时间都能容纳 2 个完整命令的缓冲器来保存数据。在本例中, 第二个命令和第三个命令的一部分会在第一个命令的所有比特都被 M4 接收完成后才会被 M4 接收。

[0269] 如图 35 所示, 为减少线路间的差别延迟的设备和方法, 以减少系统中歪斜的累积, 以及每个节点或相邻连接上需要同时处理的命令的数目。在本例提供的设备中, 每一个模块的输入数据可以输出到与输入线路不同的数据线路上, 上述布置与插入模块或电路板或底层有关。图 35 中, 线路的实际物理分布与图 34 中的一样。但是, 如果使用 AMB 或 CSE 则可能导致连接和信号线路的类型和数目的不同。所列的例子是为了说明, 如前所述实际存在的 AMB 和本发明实施例中提供的方法和设备在处理数据歪斜的不同的效果。在某些场合下, 输入和输出数据线路与之前的线路有相同的逻辑排布, 则可能仅有部分数据在不同的线路上传输。

[0270] 举例来说, 对第一个模块 M0, 数据 A 从线路 0 输入从线路 4 输出; 数据 B 从线路 1

输出到线路 3 ;数据 D 从线路 3 输入从线路 1 输出 ;数据 E 从线路 4 输入从线路 0 输出 ;数据 C 从线路 2 输入从线路 2 输出。在 M1 的输出端,延迟的差别的情况与图 34 中所示的 M1 的情况一样。但是,考察模块 M2 的表的情况,则两图之间的情况是有差别的。在介于 M1 和 M2 之间的线路 4 上传输的数据 A 只加上线路 4 的延迟,为 1 个单位 ;而在线路 0 上传输的数据 E 则加上了线路 0 的延迟。在第二个模块 M2,可以看到数据 E 的总的延迟为 7 个单位,数据 A 的总的延迟也为 7 个单位。其他数据 B, C, D 的延迟时间并不完全一样但是非常接近,可见重新分配逻辑和物理数据线路可以减少累积的延迟的差。

[0271] 在之前的例子中,在每个模块上数据 A-E 重新指定或相互交换其连接的线路 0-4,可以看到标记为 A-E 的各逻辑数据的延迟的差别是单调增加的。数据可能在每个模块上都被交换至不同的线路。在最后一个模块 (本例中是 M4),如图中与其相关的表中所示,差分逻辑数据的延迟的范围为 15 单位到 18 单位,不同延迟之间最大差 3。而与图 34 中所示的相比,按照图 34 的连接方式其延迟之间最大差 25。交换逻辑数据到物理线路之间的通信连接可以将需要进行缓冲的数据降低至原来的八分之一,上述数据是指在发送端的一个时钟周期内的所有被发送的数据完全被某一模块接收之前需要进行缓冲的数据。在本例中,所有的比特在模块 M4 内重新被排列成发送时的原始顺序。

[0272] 在本例中,比特安排策略为 :将经历过最长延迟的数据输出到具有最短延迟的线路上。类似的,具有最短延迟的数据将输出到具有最长延迟的线路上。

[0273] 在上述描述中,为了简便起见延迟都是整数值 ;但是也可能有非整数的值,非整数值可以是代表时钟频率的若干倍的以任意单位定义的延迟。可选择的,延迟和延迟间的差别也可以是时钟周期的整数和分数倍。

[0274] 如图 36 中所示,为模块对之间的延迟完全不同时的情况的示意图。图中只显示了两条线路,但是图中所示的排布方式也可以泛指到任意数目的线路。如图所示的两条线路为传输命令的线路。图 36a 中所示的情况与图 35 类似,其中,对线路进行了交换使得在通路上每两个相邻节点上各线路总的延迟的差都最小。在第一和第三个模块上进行了线路交换,在第二和第四个模块上没有进行线路交换。如果在有必要的情况下,可在第五个模块对线路进行重新排序。在第四和第五模块之间的线路 0 的延迟为 10 单位,而此时的线路 1 的延迟则为 1 单元。由于通路上总的延迟之间的差别已经最小化了,则在最后两个相邻模块之间的较大的延迟的差别使得总的延迟的差别在第五个模块变得比较大,在本例中,线路 0 的总的延迟为 18,线路 1 的总的延迟为 9,两条线路之间的差别为 9。如图 36b 所示,为一种可替换的情况的示意图,其中,允许延迟的差别进行累积直到倒数第二个模块,并在最后一段传输线路上进行线路交换,然后再在最后一个模块内将线路交换回来。则在最后一个模块上延迟的差为 0 ;但是,要注意的是,第四和第五模块之间线路上的延迟的差别应是显著的,当模块之前的单段线路上的延迟存在显著差别时,对该模块之前的通路上的单段线路进行交换才能使延迟的差别减少。

[0275] 如图中所示,各段线路上的延迟是在设定线路交换切换开关之前预先决定了的。各段线路上的延迟是在设计系统或系统原型时,制造时,模块装配到系统上时,系统启动时,每一个系统启动时,或系统操作时进行测量并确定的。选择确定延迟时间的时间与系统的复杂性和其所需的工作状态有关,并可能会考虑制造的不稳定性,温度的影响,以及各主动或被动元件的故障或老化情况。

[0276] 线路的相互交换或“交换”功能可以由切换开关实现,但是此处使用的术语“切换开关”并不用于限制上述特定功能分别在不同的设备上执行;任何可以实现线路的互换、再分配或交换功能的器件都可以称为切换开关。为了便利起见,应用在系统或方法的上述方面的术语“切换开关”包括所有可以执行上述功能的技术。该技术可使用存储区域、状态机或保存了程序的计算机的操作,以及专门的电子逻辑电路和类似器件,可以是上述单独的元件或操作也可以是它们的组合。对切换开关的操作可以是动态的也可以是静态的,这取决于产品的设计的需要。

[0277] 如图 37 所示,线路 3 和线路 4 为信号通道,这两个通道具有最低的延迟,而线路 0 到线路 2 为数据通路。使用互换技术对信号通路进行去歪斜,使其传播延迟小于数据通路,这样命令可以同时或早于数据到达。数据通路使用前述方法中的一种进行去歪斜。本例主要介绍信号线路 3 和 4,信号线路可以与任何其他的线路进行交换,这样命令可以与数据同时到达目的模块。甚至在某一相邻模块之间,部分数据传输使用的线路的延迟时间比信号使用的还短,只要控制数据和命令信号的传输时间的总的差别使得它们在合适的时间被接收即可。上述做法可以最小化数据信号的总的延迟时间。

[0278] 如图 38a 所示,为一种在线路之间交换数据元素以使所有模块之间延迟时间的差别最小的方式的示意图。图中没有显示延迟表,但是假定按照其延迟可如图布置线路间的交换。数据进行交换后在 M4 上的顺序为 C, D, E, A, B,而实际期望的顺序为 A, B, C, D, E,因此通过在模块 M4 内适当的交换数据以获得后续系统所需的数据的顺序。图 38b 总结了从 MC 的输出到 M4 的输入的交换后的总的效果,以及 M4 内进行交换后的效果。中间各连接和各中间过程中发生的交换可以逻辑上使用 MC 到 M4 的总的一次交换和 M4 内的一次交换替代。这种表示方法表示了端到端的传输过程。图 38a 示意了每一个模块上切换开关的配置状态,MC 上输出数据的顺序与 M4 的输出的数据的顺序一致。另一方面,如图 38c 所示,各切换开关的配置状态不变,但是数据顺序在 MC 传输之前进行了预重排以使数据被 M4 接收时为期望的顺序,这样就不需要在 M4 内对数据顺序进行交换。上述处理方法可以使每一个模块上进行处理时的延迟最小化,当处理过程在 MC 上进行时,预重排可以在命令和数据仍然在输出序列中时完成,而该操作所需的时间可以包含在其他系统处理时间中。由于允许利用本地 CSE 对输入数据在输出到下一个模块前预重排或不进行预重排,则进行预重排也可以简化 CSE 中的数据通路。如果 CSE 配置成树结构中的分支,则系统中的一部分 CSE 可能是叶子节点,叶子节点的下行方向没有其他的 CSE。上述预重排可以是静态形式也可以是动态的形式。

[0279] 以上描述的情况为从 MC 直接到模块的传输,但是也可以有效的应用于其他的通路,如模块之间的 DMA 传输,或应用于树结构中不同的通道或分支之间的传输。虽然本发明实施例中的描述为从 MC 到某一模块的例子,但是上述的操作可以同等的应用于反方向的传输。

[0280] 如图 39 到图 42 所示,为将逻辑数据重新分配到各线路的切换开关和切换方法的各种不同例子的示意图。如图 39a 所示,为具有 4 个切换开关元件的 Batcher 排序网络的配置的示意图,其 4 个切换开关元件连接输入线路 1-4 至输出线路 1-4。如图 39b 所示,为上述网络中 2×2 切换开关的配置示意图,其中切换开关的配置方法可以有两种:其一,元件内的输入线路分别和相同号码的输出线路相连;其二,元件内输入和输出线路互换。如图

39c 中所示的切换开关,除具有如图 39b 所示的功能外还具有复制功能:线路 1 输入的数据复制到输出线路 1 和 2,同样线路 2 输入的数据也可以复制到输出线路 1 和 2。如图 39d 和 e 示意的切换开关的电子元件,所示电路元件为 2 到 1 多路复用器。如图 39d 中所示的电子元件实现了图 39b 所示的功能,如图 39c 中所示的电子元件实现了图 39c 的功能。上述两个电路的区别在于,在其中一个电路中,2×2 切换开关没有复制功能,多路复用器选择共有的活动线路,其中一个多路复用器具有 NOT 输入 420。在具有复制功能的 2×2 切换开关中,对单个切换开关是可以分别控制的。

[0281] 如图 40c 所示,为一种 Batcher 排序网络的示意图,在该网络中输入线路顺序为 1, 2, 3, 4, 输出线路顺序为 3, 2, 1, 4(图中,输出线路的号码在前,其后的括弧内为对应的被传输至相应输出线路的输入线路的号码)。但是,如果试图得到输出的顺序为 1, 3, 4, 2, 则如图 40b 所示,可能会造成一个通路闭锁。

[0282] 如图 40c 所示,为一个无闭锁的十字切换开关矩阵的示意图。这种类型的切换开关可以用于实现任何需要的交换。如图 40d 所示,为另一种切换开关类型为 Clos 网络的示意图,该网络类型为严格无闭锁网络,可以以仅交换那些分配顺序确实被改变的线路的方式改变输入和输出线路的顺序,

[0283] 如图 41a 所示,可以通过增加其他的步骤扩充排序网络的切换开关的能力,这样的网络有时称为 Benes 网络。这样就可以实现如图 40b 所示的交换了。切换开关增加的步骤可能引起额外的延迟。不过上述开关是一个可配置的无闭锁切换开关。如图 41b 所示,为上述切换开关扩展成 8 条线路的配置示意图,对切换开关的宽度没有限制。

[0284] 在某些排布方法中,存在一个或多条线路互换的方式。如图 42 所示,为系统中的一些线路经常会需要进行全交换的情况的示意图,并说明了使用器件和电路元件连接如输入线路 1 到多路复用器的输出再到线路 8 的输出的情况。其他的输入和输出也是类似进行连接。如图所示,还有两个 Benes 网络连接输入和输入至多路复用器的输出,并对其他排布进行配置。所需要的排列可以通过输出多路复用器进行选择,该多路复用器与固定的网络 and 不同配置的切换开关相结合。此处仅仅示意了一种固定的网络的形式,可以使用不止一种的固定网络与具有更大输入电容的多路复用器结合使用,如 3 到 1 的多路复用器。对于之前所述的多种电路,当电路不使用时可以使其处于低功耗状态或关电状态,如果电路有时钟的话对其进行门控,或执行任何其他可以降低功耗的操作。

[0285] 在通路或电路中,系统可能会有许多差分延迟。这种差分可能是热膨胀或参数变化,或其他确定性或不确定性的因素的影响引起的。当系统中存在差分延迟,信号从树结构的根节点(比如 MC)向各模块传输时,其延迟时间不固定在某一确定的值上。即使在不需修正相位差异,而且数据可以与一时钟速度足够慢以致数据不会产生歪斜的全局时钟锁定的情况下,仍然会产生小范围的相位改变;同时,仍然存在从器件到器件的传输距离的差别,或从任何一个器件到首端(根)传输距离的差别。当时钟变得不是完全同步,或数据存在歪斜,全局时钟里有抖动,或相对全局时钟存在相位差异且线路是自同步的时,上述考虑仍然是适用的。

[0286] 如图 43 所示,为两路数据通路在其到达的模块中合并(如在树结构中两个分支(线路)在其交叉的节点中的合并)的示意图。在上述情况下,造成每一个树结构中的延迟的差异的原因可能是,在节点上的分支之间存在的延迟的差异、在每一个分支内的延迟的

差异、在不同线路之间的延迟的差异。可以由 CSE 或相关的电路对分支进行合并,对上述功能的任何描述并不意味着对其物理位置的限制,而是表明该功能在每一个元件内完成。

[0287] 如图中所示,为了描述方便,将通路 I 上 4 条数据线路上传输的数据用数字编号表明,通路 II 上 4 条线路上传输的数据则用字母编号表明。通路 I 上各通道输入的数据按时间顺序排列整齐。数据输入进时钟数据恢复单元 (clock data recovery unit, CDR)。每一个线路或线路组都有一个单独的 CDR 相连。通路 II 上输入的数据在不同的线路之间存在时间歪斜。每一个 CDR 的输出都与多路复用器 500 的输入相连,该多路复用器用于选择两个通路中的一组数据并将数据输出到发送器 (transmitter, TX) 510 或其他电路。虽然多路复用器仅选择了一个通路,但是连接多路复用器和 TX 之间的 4 条线路上还是同时显示了通路 I 和通路 II 上发送的数据。图中的上述显示仅仅是为了更方便的观察到数据传输时的时间问题;而实际上仅仅选择了一条通路上的数据,被将其在多路复用器和发送器 TX 之间传送。从图中所示的数据的比较可以看出即使每一条线路都有自己的时钟,在通路 I 和通路 II 之间的数据传输相对于时钟仍然存在时间差。虽然每一个通路中的每一条线路都可能有一个单独的时钟,但是为了便于描述,假定两路数据通路分别有单独的时钟。

[0288] 当选择通路 I 或通路 II 上的数据并发送时,需要考虑其在一串 CSE 中的下一个 CSE 或任何相关的数据接收器上的影响。为了描述方便,称一串 CSE 中的下一个 CSE 为上行 CSE。上行 CSE 的配置与图 43 中所示的 CSE 一样,接收到的数据在上行 CSE 的输入端将会在一个单独的数据通路上,在该通路上数字或字母标注的数据由多路复用器布置其顺序并通过发送器 TX 发送。通路 I 上的数据输入上行 CSE,而树结构中的另一个分支则输入通路 II 上的数据。以下描述适用于从通路 I 或通路 II 的输入,但都以从通路 I 的输入的数据为例进行描述,该数据来自如图 43 中所示的 CSE。

[0289] CDR 可以使重新获得的时钟与通路 I (数字编号) 的数据联系。当多路复用器的开关将输入切换到通路 II (字母编号) 时,与通路 II 有关的时钟频率则可能与通路 I 的时钟频率不同,而在上行 CSE 上根据时钟频率进行的数据采样将会出错。

[0290] 如图 44 所示,为一个至少部分缓减上述情况的电路的示意图。在 CDR 中有相位校准电路,该电路调整数据的时间延迟使得来自通路 I 和通路 II 的数据对齐于相同的时钟沿,比较通路 I 和 2 的输入数据和发送器 TX 输出的多路数据即可看出数据在 TX 的输出端是对齐的。需要提醒的是,两路通路中实际上只有一路数据会在某一时间出现在发送器 TX 的输出端,但是图中在 TX 的输出端同时显示了两路数据是为了描述两路数据与同一个时钟的关系。上述数据的对齐可以减少通路合并中的一部分的延迟差异,在本例中,线路 2 上的数据情况则显示两个通路之间存在一个完整的时钟周期的偏移。实际上,偏移可能不止一个时钟周期;而时间歪斜仍然没有完全消除。

[0291] 如图 45 所示,为在图 44 的基础上为通路 I 和通路 II 各增加了去歪斜处理的示意图。通过对通路 II 的线路 2 的数据增加一个相等的时钟周期的延迟来达到去歪斜的目的,这样通路 I 和通路 II 的线路 2 上的数据被多路复用器选择后再被上行 CSE 接收时,不存在明显的数据歪斜。

[0292] 如图 46a 所示,为在一个更高级的结构中图 45 所描述的概念的示意图。图中存储控制器 MC 和模块 M0-M7 连接成一个树结构。每一个数据通路的时间延迟用连接模块 Mi 的线路的长度来表示,每一个模块内的时间延迟用每一个模块的长度来表示。从图 46a 中

可以看出,通路 MC-M7 为延迟时间最长的通路。图中只显示了在每一个电路元件之间的 K 条可能的线路中的一条线路,但是本具体实施例中的描述还适用于其他的线路。在图 46b 中,重画了上述树结构,用水平轴代表连接电路元件的线路和每一个电路元件内的时间延迟。对每一个数据比特,电路元件之间使用的线路可以是不同的物理线路,但对数据或命令进行解释的电路元件处的逻辑数据比特将是正确的。如图中所示,即使其物理线路为树结构中其它部分的不同的数字编号的线路,例如图 39-42 中所示的对线路进行互换之后的线路,但最终连接至 MC 的线路 0 的线路组即是携带会到达线路 0 的数据的线路,

[0293] 如图 47a 显示了网络中的两个子树的情况:M0, M2, M6 和 M0, M1, M3, M7。如图所示,通路 M7-MC 为最长的通路。在每一个电路元件上都有两个输入通路和一个输出通路。两个输入通路中的一个通路为非临界通路,非临界通路在源和目的地之间的总的延迟比其他通路的延迟小。比如,在模块 M1,其到 M4 的通路相对通过 M7-M3-M1 的通路为非临界通路。在上述情况下,相对于通路 M7-M3-M1-M0,通路 M4-M1-M0 会增加一个额外的延迟,但通路 M4-M1-M0 的总的延迟并没有增加。在模块的数据通路的连接端处延伸出的一个突出部 550 来表示额外的延迟。

[0294] 用图 47 中介绍的表示方法,图 48 中表示了一个与模块控制器 MC 的逻辑比特 0 通路相连的 8 个模块 M_i 形成的树结构网络。从模块 M_i 到 MC 的比特 0 的比特传输时间延迟表示为 dM_i_N ,其中 i 表示模块的编号, N 表示逻辑比特。所述延迟是指从模块 M_i 的输出端到 MC 的输入端的这段线路产生的延迟。为了简便起见,没有显示源模块 M_i 和接收电路 MC 内的延迟。需要注意的是,模块内的额外的延迟可能发生在模块内的任何地方,所述延迟用模块上的突出部表示。对于网络中的各时刻,其中时刻 tM_i_N 表示从 MC 的逻辑比特 0 发送的命令或数据的被模块 M_i 接收到的时刻。如图 48 所示,为比特 0 和模块 5 的情况,则接收时刻 $tM_5_0 = t + dM_5_0$,其中 $t = 0$ 表示数据从模块控制器 MC 发出的时刻。

[0295] 对于任意的比特 N 也可以得到类似的分析结果,在图 49a, b 中,其模块的排布情况和图 46b 中的情况一样,但是图中的延迟情况是针对另一比特,比特 N 的。如图 50a, b 所示,对于比特 N ,可能有不同的具有最长通路的子树组,在每一个模块组合成的通路中可能有不同的具有非临界通路的子树集合。在本例中,具有最长延迟时间的子树为 M5, M2, M0 和 M7, M3, M1, M0, MC。与上例中比特 0 的情况相比,第一个子树与图 46 中的情况不同,但是第二个子树则没有改变。对于比特 N ,延迟可能加在互连 M2 和 M6 的端口上,而在比特 0 的例子中,额外的延迟则可能是加在连接 M2 和 M5 的端口上的(如图 47 中所示)。

[0296] 如图 51 所示,为使用之前介绍的符号表示的延迟的示意图。对于热效应和其他缓慢变化的影响需要调整其造成的延迟以保持数据的相位和比特的同步,上述数据由模块中的多路复用器进行选择,并按照公共时钟进行采样。如图 51 所示的模块 M0, M1 和如图 52 中所示的模块 M1 中用延伸出来的梯形 560 标识的延迟表明了上述的变化的额外的延迟,其中梯形的总的长度 565 代表了最大总的额外延迟时间,而横向的虚线 570 表示最小的额外延迟时间。另一方面,由于存在某些因素的影响,对每条线路模块内的延迟可能不同,上述因素可能包括模块的内部结构、处理延迟和其他类似因素。当选择不同的线路在模块间传输逻辑比特时,模块的延迟可能取决于所选择的线路,这一点可用如图 53 中所示的宽突出部 575 来表示。需要再次强调的是,上述仅仅是一个表示符号,并不意味着延迟必须在模块的输入或输出端产生。

[0297] 取决于控制时间延迟方法的选择,可能可以在模块中使用去歪斜之前利用延迟来进行歪斜的纠正。如图 54a,b 为两种引入延迟的方法的例子示意图。如前所示,在图 54 中,延迟加在与 M6 相连的 M2 上,其中 M5 到 M2 的通路为非临界通路。可选择的,如图 54b 所示,延迟可以加在 M5 上,也可以从 M2 和 M0 上去掉。上述情况可被看作,减少了从 M6 到 M2 到 MC 之间的总的延迟,而没有减少 M5 到 MC 之间的延迟。

[0298] 如图 55a 所示,为图 46 和 49 中描述的树结构的延迟的示意图,其中分别表示了标识为 0 和 N 的比特的延迟。如图 55b 中所示为与图 55a 相同的树结构,但是图 55b 中没有模块 M7。删除了长延迟,如图 55b 中所示,即为没有“最长延迟模块”的情况。比特 0 的最长延迟发生在到模块 M6 的通路,而比特 N 的则发生在到模块 M4 的通路上。

[0299] 图 46-55 为时间延迟的一些例子,如,模块 M 和模块控制器 MC 的返回通路上的延迟。MC 和模块 M 之间的命令和数据的传输,信息的确认和数据的返回也同样存在时间延迟;但是,在数据返回的情况中返回通路需要从一路以上的通路中选择数据,与之不同的是从 MC 发出的命令或数据则只需要将数据下行发向树结构中的各分支即可。同步化时需要对相位进行校正,而不需要对额外时间延迟造成的歪斜进行校正。因此,即使是对相同的比特,从 MC 到模块 M 的时间延迟和自模块 M 的返回通路的时间延迟可能不同。在图 56 中,当模块上存在延迟时,为了简便起见,模块之间所有的延迟,包括相位同步延迟的歪斜都作为连接到模块的线路的长度中的一部分。如图 56a 所示,在树结构中,命令从 MC 发出,模块内传输产生的全部延迟用点线表示。为了与通路管理中的讨论一致,对于不是广播命令的其他命令实际上不会经过在模块之间的每一个连接,而是仅仅经过一部分取决于目的地址的连接。但是对于经过特定的通路传输的情况,图中则显示了该特定通路上的延迟。

[0300] 如图 56b 所示,其中树结构与图 48 中所示的一样。模块之间的线路是图 48 中所示的线路和图 56a 中所示的线路之和。点线 580 表示从 MC 的比特 0 传向其他每一个模块时遇到的时间延迟,实线和突出部则代表信号从各模块返回至 MC 的比特 0 时延迟。为了方便表示,从 MC 发出信号的准备时间,目的模块解码、对信号进行操作和传输响应信号的时间,以及 MC 上解码和对信号进行操作的时间都忽略了。

[0301] 图 56 中表示了命令从 MC 的比特 0 发送向任何一个或多个模块 M_i 以及 M_i 对命令的响应被 MC 接收的总的传输时间。

[0302] 图 57 中显示了图 56b 中的更多细节,包括表示在树结构中命令和响应的总的传输时间的符号。 sdM_i_N 表示命令(或数据)从 MC 到 M_i 以及通过线路 N 第一个比特从 M_i 到达 MC 的输入端的时间延迟。 $stM_i_N(t)$ 表示从比特 N 发送到响应被发送电路接收到的所用的时间。如图所示,对于模块 M5,命令从 MC 的比特 0 向模块 M5 发送,并且其响应被 MC 接收的情况有 $stM_5_0(t) = t + sdM_5_0$ 。

[0303] 在上述例子中描述了输出和返回的通路结构一样的树结构中的情况。但是上述结构并不是必须的。图 58a 和 b 表示了模块相同但是信号从 MC 发出和接收的路由不同的情况。图 58a 中表示的是一返回路由,而图 58b 中表示的是一发出路由。图 58c 中表示的是信号的整个来回传输的时间,其中虚线和实线分别与之前的例子中一样代表不同的用法。在本图中,一些通路在不止一个地方出现,这些延迟用圈在线路上的椭圆圈以及表明存在延迟的线路进行提示。图 59 与图 49 意义一样,但图 49 中只显示了返回通路,而图 59 中还显示了发出通路的情况。

[0304] 如图 60a, b 所示, 为具有多条信号通路的示意图, 其中上述信号通路 with 数据通路是分离的。在图 60a 中, 有经过 M0 到 M2 到 M5 再到达 M4 的通路。上述通路用以箭头结束的线路与前面讨论的通路 M0 到 M1 再到 M4 相区别。类似的, 从模块 M3 也可以经两条不同的通路到达 M7。举例来说, 上述两条到达 M7 的通路可能代表链路的通道化, 其中每一通道都有相关的信号成分。用这种方式表示信号通路可以说明不同的通道具有不同的信号延迟的情况, 比如, 当因为使用具有不同延迟的不同线路时。到 M4 的附加的通道可能代表用于通信的附加的连接。上述的情况可能可以看作是当模块以栅格网孔的方式连接时的一个例子。在一个方面, 存在一个以上的模块控制器与同一模块通讯时, 将会存在不同的通路长度, 由于存在一个以上可用于传输控制信号的路由, 则可选择不同的通路以避免信号冲突。

[0305] 如图 60b 所示, 为与模块控制器 MC 和每个模块 M_i 之间的通路有关的时序的示意图, 其中, 信号通路用点线来表示, 返回数据用实线表示。来自模块 M4 的额外延迟出现了两次, 用椭圆圈指示了与上述部分有关的地方。其中一条通路为从 M0 到 M1 再到 M4 的通路, 另一条为从 M1 到 M2 到 M5 再到 M4 的通路。

[0306] 在以下对从模块返回的数据或命令的细节的描述中 (其中的数据或命令是对其他命令的响应), 为了便利起见, 假定命令从 MC 传至所有模块时没有时间延迟, 目的模块对命令的处理和响应也没有时间延迟。这将有用于使用符号进行描述。进一步的, 将会首先讨论时隙模型, 而在后面的说明中也可以看到这种限制也是可以去除的。图 61-71 中所示的关于时序的例子说明了数据是如何到达 MC 的。

[0307] 时隙显示在一个时间轴上, 时间轴的右边表示到达模块控制器 MC 的时刻, 而向左的时间表示命令或数据将在未来的时间到达 MC。符号 t_{Ci} 表示包 C_i 的第一个比特到达 MC 的时刻。上述讨论与线路 0 有关, 而相同类型的时序安排也可用于每个单独的线路。进一步的, 以线性的方式显示有延迟 d_{Mi} 的时间轴, 而延迟是在树结构中产生的, 并且上述延迟也可与差分延迟抖动缓存混合。这即是表明, 模块向链路的一条线路上插入比特的实际时间为传输的延迟时间。图中没有显示由于抖动 / 热效应缓存导致的时间的变化影响, 因为这将使图示变得非常的复杂。上述简化的结果是, 如果在网络结构中的较低层考虑比特的时序, 则看上去各比特是相对于彼此在移动, 这使得这些比特处于时间轴的错误的位置上; 但是, 对抖动 / 热效应缓存进行纠正后在 MC 收到的时序则是正确的。

[0308] 一个时隙的总的长度用六边形框 900 表示, 一个比特的时间周期则用小的六边形框 910 表示。在本例中使用了时隙系统, 其中对模块进行限制, 只有当一个合适的时隙到达时数据才会被传输否则模块需要进行等待。例如, 在 t_1 时刻读命令被发往 M7, 由于 $t_{C_{i+4}}$ 已经在模块 M7 开始了, 则对上述读命令的响应时间不能早于时隙 $t_{C_{i+5}}$ 。图 61b 中描述了当一个完整的时隙已经过去的情况。此时, 所有的时隙都移动一个时隙, 时隙 $i_{C_{i+1}}$ 将要抵达 MC。模块 M7 已经开始插入数据, 在时隙 $i_{C_{i+5}}$ 中已经插入了 5 个比特的数据。图 61c 显示了 $i_{C_{i+2}}$ 将要到达 MC 的情况, 可以看到此时, 来自 M7 的所有比特已经插入了时隙 920。在以后的某个时刻, 上述时隙将会被 MC 接收。上述情况中, 比特在每一个时钟脉冲中被顺序插入, 这样数据就被填入一个时隙中了。

[0309] 一方面, 如图 62a-c 中所示, 给每个模块都安排一个固定的时隙, 这样从模块到树结构的根节点 (比如模块控制器 MC) 的数据或命令以一个预定的顺序抵达。在本例中, MC 可传送多个命令。为了简便起见, 假定命令和数据在具有固定延迟的系统传送。在本例中,

对命令进行处理时假定具有最远模块的延迟。这样,即使模块具有较短的延迟,其返回的命令或数据将会被延迟以使其可填入预定的时隙中。

[0310] 如图 62a 所示,时隙 $tC(i+6)$ 提前了 5 个时钟周期到达 M7,并且已经在该时隙内插入了 5 比特数据。另外,已经向 M0 和 M2 发出了读命令,对这些命令的响应的时隙分别分配在 $tC(i+6)$ 和 $tC_{-}(i+7)$ 中。如图 62b 所示,为 8 个时钟周期过后的情况,其中 M7 已经将数据填满分配的时隙,另外,已经向 M3 和 M6 发出了读命令,对这些命令的响应时隙分别分配为 $tC(i+9)$ 和 $tC_{-}(i+10)$ 。经过 8 个时钟周期后时隙移动了一个完整的时隙周期。需要注意的是,通过比较时隙的位置和延迟时间 dMi 可知,尚没有随后的时隙到达用于为这些时隙中安排携带响应的设备,也没有其他模块插入了信号比特。

[0311] 在图 63a 中,又过了 8 个时钟周期的时间,而被分配给模块 M0, M2 和 M6 的时隙向前移动了,但是还没有在时间轴上达到相应模块可向其插入数据的时间。在图 63b 中,又过了 8 个时钟周期,与模块延迟 dMi 比较可知,此时分配给模块 M0, M2 和 M6 的时隙都到达了合适的模块;并且模块 M0 已经插入了 1 个比特,模块 M2 已经插入了 5 个比特,模块 M3 已经插入了 2 个比特。可以注意到,到达 MC 的时隙上标识了从序列开始以来已经过去的总的时钟周期的数目。

[0312] 在时刻 $t_{MC_0} = t_1 + 45$,模块 M6 已经插入了 7 个比特,而其之前的时隙都已经插满了数据。在 4 个时隙时间后,在时隙 $tC_{-}(i+10)$ 结束时,所有的数据比特将会抵达 MC。

[0313] 完成上述序列传输的全部时间即是最远模块的时间延迟,在本例的线路中最远模块为 M7。对于别的线路,具有最大延迟的模块可能为别的模块。在这种情况下,延迟可能会增加成最大延迟。另一方面,如图 64 中所举的例子,其具有可变的延迟时序。仍然使用时隙系统,但是向模块发出的命令可能被分配给第一个返回的时隙,该时隙没有被分配给任何模块,也没有到达与上述接收命令的模块相关的延迟时间。为了进行对比,令图 64 和 65 与图 62 和 63 的时间序列一致。但在本例中,当向 M0 和 M2 发出命令时,其命令响应时隙分配为 $tC_{-}(i+4)$ 和 $tC_{-}(i+5)$ 而不是图 62 和 63 中的 $tC_{-}(i+7)$ 和 $tC_{-}(i+8)$ 。如图 64b 中所示,经过 8 个时钟周期后,向模块 M3 和 M6 发出了命令,其响应时隙分配为 $tC_{-}(i+7)$ 和 $tC_{-}(i+8)$,而不是原来的 $tC_{-}(i+9)$ 和 $tC_{-}(i+10)$ 。图 64a 到图 65c,显示了返回时隙向前行进,数据已经在适当的时钟间隔中插入的过程。对发往模块 M6 的命令的响应的最后一个比特在 $t_{MC_0} = t + 55$ 时刻被 MC 接收,而在如图 62 和 63 所示的使用固定延迟时序的例子中,其响应的最后一个比特直到 $t_{MC_0} = t + 71$ 时刻才被 MC 接收。可以看出,由于使用了可变的延迟时序,来自任何模块的响应的延迟都可以减少。

[0314] 如图 66 和 67 所示,在本例中取消了对数据与传输时隙的限制。在图 66a 中,在 $t = 1$ 时刻向 M7 发送了读命令 R(在本例中忽略了传输和处理命令的延迟),图中所示为 4 个时钟周期后的情况,此时已经在线路上插入了 4 比特数据。如果也向 M0 和 M2 发送了命令,则这些模块也可以插入数据。如图 66b,为图 66a 所示的时刻又过去 4 个时钟周期的情况,其中 M7 的所有响应比特都已经插入时隙中,M0 和 M2 的响应中的四个比特也插入了时隙中,并且此时向 M3 和 M6 发出了命令。如图 66c 所示,又过了 4 个时钟周期,模块 M2 和 M0 已经完成了响应比特的插入,但是 M3 和 M6 还没有开始插入比特。因为有可能与 M7 的响应时隙重叠,所以没有立刻为 M3 和 M6 分配响应时隙(对 M0 和 M2 则是立刻分配的),而是将 M3 和 M6 的响应时隙放在了下一个可能的时钟周期。图 67a-c 中显示了后续时间序列的情况。本

例描述的情况,可能会使时隙之间间隔一定的空间。这使得可以利用上述空间来传输不同长度的数据或命令,或者用来交错传输来自不同模块的响应。

[0315] 如图 68 和 69 所示,为显示了信号延迟的非时隙响应的情况的示意图。信号的延迟决定了响应时间的位置,其中还考虑了命令到达模块的时间,并且对比特插入到线路上进行了延迟。这描述了没有指定模块向线路上插入比特的时间的情况,上述线路在模块接收命令的时刻已经通过了该模块。因此,系统并不需要具有时隙时间序列。

[0316] 模块控制器 MC 所在的系统或是总线可能以一个较低的时钟速度运行,其时钟速度可能低于该 MC。MC 也有可能在一些时间周期中接收高速数据、对数据进行去串行化、对数据进行闭锁以及将数据发送到其他接口、或在模块控制器 MC 内使用数据。如果到达 MC 的数据与较低频率的时钟进行锁定的时间不足时,则数据将在低速时钟的时钟频率下使用。

[0317] 如果比特没有在将被使用的下一个低速时钟周期及时到达,则该传输与比特在刚过去的一个可能的时间到达的传输相比不具有任何优势,上述可能的时间中比特可能在下一个低速时钟周期被使用,这就需要提高系统性能以及降低逻辑复杂度。

[0318] 如图 70a 所示,为与图 68a 中所示的系统具有相同信号延迟的系统,其中, M3 的信号延迟 (sdM3) 比 M6 的信号延迟 (sdM6) 小,上述两个模块对 t1 时刻发出的命令的响应发生在同一时刻 $t1_ (t1)+48$ 。因此,如图 70b 所示,如果 M3 的信号延迟调整到调整的信号延迟 asdM3,则 M3 的响应潜伏期将会保持不变。在本例中,模块控制器 MC 的命令返回时序控制器可能不会在高速时钟周期的基础上对时序进行控制,而是在低速时隙上进行。命令时序控制器也可以对信号延迟落入具有相同时隙的模块的时序任务进行处理,比如图 70b 中所示的 sdM6, asdM4, asdM4 和 asdM3。

[0319] 如果上述延迟的调整有利于模块控制器 MC 的操作或执行,则模块信号延迟的这种调整将会仅仅作为“虚拟”调整来进行。即是说,信号仍然以 sdMi 而不是 asdMi 的延迟到达模块 Mi,但是模块控制器则将信号看作以 asdMi 的延迟到达来进行操作。

[0320] 延迟也有可能由于进行如图 35-38 中所示的线路交换而增加,上述交换通过增加某一模块的信号延迟而减少了另一模块的信号延迟。总的信号延迟为前向和后向延迟的组合。如在图 70a 中,如果增加一小部分 sdM2 的时间,则可以减少 sdM3,使得 $sdM3 \leq dM6$,这样就可以在不增加 M2 的命令潜伏时间的情况下,减少 M3 的命令潜伏时间。

[0321] 通过调整一个以上的其他模块的延迟可能可以改善一条通路上的总的延迟的情况。对延迟的改善也可能使另一模块的命令延迟时间增加。如图 70a 所示的例子中,增加了 sdM1 使得 $sdM1 = sdM2$,同时降低了 sdM7 使得 $sdM7 = sdM6$ 。在本例中,M1 的命令潜伏时间可能增加了一个时隙时间,而 M7 的命令潜伏时间则减少了一个时隙时间,同时 sdM1 仍然比 sdM7 小。

[0322] 当模块的命令延迟减少时,对延迟的调整实际上可能使得其延迟比模块增加的延迟变少,这将会有利于总的延迟的减少。

[0323] 如图 70c 所示,其配置与图 70a 中的一样,其中,模块控制器 MC 可以在不固定的时隙传输命令,或 / 以及可以在低速时钟速率下传输命令。如图 70c 所示,为 $t1+4$ 时刻,或者说是图 70a 中所示时刻的 4 个高速时钟周期后的情况,其中时隙向模块控制器 MC 移近了 4 比特,这产生了一系列不同的调整信号延迟值,使用这些值可以不影响模块的命令延迟。

[0324] 如图 71a 所示,其中模块 M0 和 M2 的数据延迟和命令延迟与之前的例子中的情况

一样,图中还示意了每一个模块执行 3 个不同的命令:CMD#1, CMD#2 和 CMD#3 的时间,以及当前一个命令为 CMD#2 时执行命令 CMD#3 的时间(这个时间表示为 CMD#3_(2))。这种表示方法也可以表示之前有一个以上命令的情况,或者表示模块或系统的不同状态。在本例中,上述两个模块的命令延迟时间相同;但是对于每个模块其给定命令的延迟并不一定是相同的。

[0325] 如图 71b 中所示,为命令信号延迟的产生的示意图,其中符号“sd[3]M2”表示模块 M2 对命令 CMD#3 的信号延迟,而 sd[3_(2)]M2 表示模块 M2 对命令 CMD#3_(2) 的信号延迟。

[0326] 对于一个模块,可能存在不止一个调整信号延迟。例如,模块控制器可能将从一个模块返回的数据发送至一个以上的目的地,或者对上述返回数据进行不止一种类型的操作,这样就可能遇到一个以上的时钟周期的边界。上述时钟周期的“边界”也可以是指向设备、功能或处理过程等的连续过程中的时间段起始时间。并没有要求时钟具有固定的相位或频率关系。

[0327] 在前面的例子中使用了“高速时钟”来表示高速连接上的“一个比特时间”,虽然高速时钟与比特时间有些不同,比如,数据在两个时钟边缘传输时,使用多值的信号,或者不同的时钟,多相位时钟,或 / 和使用一个以上的时钟。需要说明的是,为了叙述的清楚,所给的例子中认为时钟时间和比特时间是一样的。

[0328] 同样是为了描述清楚,所说的数据在“下一个低速时钟周期”到达也应该理解为下一个数据可以被使用的时机,这个时机可能不一定是立即到达的下一个时钟时间。

[0329] 本具体实施例中还给出了物理和逻辑的布局结构。对于本领域普通技术人员来说其他的布局方式也是可能的并且是显而易见的。例如图 72 中所示的与图 5 类似的树结构可以布局在主板或底层上,并使得模块 M0-M7 以平行于彼此的方式线性排布。这种排布方式有利于冷空气的流通,或优化母板或主板上的电子布局。如图 72 中所示,模块中的线路以箭头表示终止,并在模块的边界处起始以表示模块之间的连接。这些线路中的一部分穿过了模块但是没有与穿过的模块有任何功能上的连接,这种情况用点线或虚线来表示;但是从一个模块起始,穿过另一个模块且没有用箭头作为终结的线路将被当作物理上通过了该模块。通过某一模块而没有功能的相互作用的线路的物理方面的特性可能取决于设计的细节,但是一方面,点线或绕过的线路可代表是穿过了印刷电路板的背面,或是印刷电路板或底层中的某一层的线路。

[0330] 如图 73a 所示,为与图 72 一样的排布方式,但是其中每一个模块具有“前”侧和“后”侧,图中用宽箭头示意了模块的前侧所处的位置。每一个模块在前侧有一个端口,后侧有两个端口,其中每个端口有两个单方向的线路组。另一方面,每一个端口有可由双向的线路组成。在上述排布下,两个模块(如 M0 和 M1)之间的一个接口将会导致如图 73b 中所示模块的前侧面向另一模块的前侧的情况。图 72 中所示的排布中的模块的方向都显示在了图 73a 中。

[0331] 如图 74a-d 所示,为模块的另一种排布方式。图 74a 为模块 M0 和 M1 的俯视图,图中的模块之间的线路为单向线路。可选的,其中的每一条线路也可以是双向的或可以是线路组的组合,符号 R 和 T 应当看作是方向区别标识而不是代表接收或发出连接。在本例中, R 连接在模块的左边, T 连接在模块的右边。当连接模块 M1 的 T 到 M0 的 R 时,上述连接会通过上述两个模块的下方,而模块 M0 的 T 连接到模块 M1 的 R 连接之间的线路则不需要通

过模块的下方。

[0332] 图 74b 为图 74a 中的排布的剖面图,其中一些线路布置在板中某一层上 970a,一些线路布置在板的另一层 970b 上。每一个模块都可以是一个电路板,在该电路板上可以布置不同的电路元件:上述元件可包括存储芯片、CSE、其他接口、计算元件和类似的设备。图 74c 为另一个排布的俯视图,图 74d 为相应的剖视图。在图 74c,d 中的模块用符号 S_i 表示,其可代表切换开关。切换开关 S 可以是 CSE 或其他布置在电路板上的元件。除非有其他特别的限制,符号 S , M , T , R 为一般性意义的符号,而不是局限为代表切换开关、模块、发送功能、或接收功能。其中的线路可以是单向的或是双向的,也可以是具有上面描述的特性的组合的线路组。

[0333] 如图 75 所示,其中模块之间的相互连接可能不在同一个主板或底层上。如俯视图 75a 中所示, M_0 和 M_1 之间的连接在同一个主板上,而 M_2 和另一模块(图中没有显示,但是该模块处于模块 M_2 的右边)的连接也在同一主板上,但是 M_0 和 M_2 之间的连接则处于板 980 上,板 980 位于模块 M_0 和 M_2 的顶部并连接了这两个模块。可选择的,也可以通过有线连接和连接器或其他方式完成上述连接。上述配置方式可参见如图 75b 所示的剖视图。

[0334] 图 76 表示了图 72 中的排布方式的一部分,其中到模块的连接为交替出现在模块的任一边的连接端口 T 和 R 。如图所示,对于一个单独的交替,接收连接 R 在模块的右边,发送连接 T 在左边,接下来则是发送连接 T 在左边,接收连接 R 在右边。对于每一个模块都有三个端口。上述连接的交替可以有任何多次,与每一个交替相互连接的线路为 0 条以上。每一组中的线路数目可以不相等。这种布置连接的方法可以不对模块的前侧和后侧进行区分,可以如在图 73 中所示布置连接,其中可以不需要旋转该模块两端位置颠倒来确定面向区方向。因此,在一些应用中可以生产对称的插入卡。

[0335] 另一方面,图 77 中模块的排布方式与上图一样,但是 T 和 R 连接没有相互交替。本例中, R 连接在板的左边, T 连接在板的右边。这表明对于特定的模块和连接的布置可以不进行连接的交替。

[0336] 如图 78 所示,其模块的连接布置方式与图 77 中一样,但是从 M_1 到 M_3 的两组传输线路不是都布置在顶部,而是在距离顶部的第二组的位置。如图 77 中所示,模块 M_0 到 M_3 之间的线路从 M_0 的顶部经过处于 M_0 的中部的切换开关(图中没有显示),再穿过 M_1 的顶部到达 M_3 。即最长的信号通路为 M_0 到切换开关并穿过 M_1 ,再到模块 M_1 的顶部的整个长度的一半。根据如图 78 中所示对线路进行排布,从 M_3 到 M_0 的输出再到 MC (图中没有显示)的距离被缩短了。在这方面仅仅考虑了模块上的通路的长度。而模块之间的距离也是整个路由的一部分。在一些情况下,信号在模块上经过的距离比模块之间的距离更长。可以看到,对模块上的通路进行互换可以减少到树结构的更下级的模块的延迟,这种互换类似于数据在逻辑上在传输线路上进行交换的情况。

[0337] 如图 79 所示,其中模块之间具有不同宽度的链路。模块 M_0 到模块控制器(图中没有显示)之间具有两个具有完全端口的完全的线路连接。而 M_0 到模块 M_1 , M_2 , M_3 和 M_4 的线路数目分别只有到模块控制器的线路数目的一半。由于每一个模块都可以将线路连接至 M_0 ,则两个模块可能同时以全速相互通讯。 M_5 , M_6 , M_7 和 M_8 都有到 M_1 的连接,在这样的例子中,所有的模块可能同时具有全速连接。上述这种配置方式可以称为“丰满(fat)”树。在这样的树中,通往根节点的数据通路在靠近根节点时逐渐变宽,这样可以在没有拥塞

的情况下访问多个模块。

[0338] 如图 80 所示,为以网格形式排布的模块组的示意图。在图 80a 中,4 个模块连接成基本的正方形,其中模块对之间相互平行,并且模块对处于同一直线方向上。当模块通过连接器垂直安装在板上时,上述的这种模块之间的平行排布关系将有利于冷空气的流通。不考虑逻辑连接关系,其物理上的这种排布可以变得比较规律化。在图示中,模块的发送和接收连接交替出现。当采用 U 形连接时,U 形线路组中的外圈的连接具有最长的路由,但同时该路由离模块中心最近并且可能离位于中心位置的切换开关或 CSE 最近,则使用 U 形线路连接 M0 和 M2 可能可以降低歪斜。

[0339] 如图 80b 所示,为图 80a 中所示的电路的逻辑表示的示意图,图中每一个方块代表一个模块或类似装置,每一条单一的线路代表互连在模块之间的所有连接(如模块之间的链路线路和线路)。基于上述基本结构,图 80c 中表示了 4 个模块的基本组合如何构造成模块网格网络的情况。上述网络并不需要在任何维度都是规则的,在网络中可以存在间隙。

[0340] 如图 81a 中所示的 6 个模块组的排布,模块之间可以形成各种各样的几何排布形状。(需要注意的是,如图中所示,并没有要求不同的模块为不同的物理尺寸;使用不同尺寸的块仅仅是为了表示的方便)。6 个模块的逻辑排布与 4 个时类似,如图 81b 所示为一六边形形状的排布。图 82 中表示了由六边形组成的网格。在图 80 中,从 M0 发出垂直向上的通路与 M2 的底部的通路相连接,而在图 81a 中,在板的顶部上的传输线与 M2 的板的顶部的接收线相连,且 M0 与 M3 的 T 连接相连。实际上,M0 可与 M5 相互通讯,M3 可以与 M2 相互通讯,而如图 82 中所示,由于 6 个模块的模块组相互之间形成分支使恰当的终端相关联。

[0341] 另一方面,如图 83 所示,为与处理单元(CPU)、或模块控制器、或其他可以与模块进行连接的设备相连的一种模块网格结构的示意图。在这种情况下,4 个中央处理单元(CPU1, ..., 4)可共有一个存储器。

[0342] 如图 84 所示,为 2 个 CPU 与 2 个树结构(即有两个根节点的树结构)相连的排布的示意图。一方面,CPU C0 可以作为一个树结构中的根节点与模块 M0, M2, M4 和 M6 连接,而 CPU C1 作为模块 M1, M3, M5 和 M7 的根节点。通过根节点 CPU 到达一个树结构可以不需要考虑对另一个树结构的操作。但是,当需要访问另一个树结构时,需要提供一种防止在树结构系统或 CPUs 之间的潜在冲突的协议。双端口存储器可以允许两个或两个以上的 CPUs 在同一个模块上存储信息,并提供公有模块的访问通道。特别的,可以对模块 M0 到模块 M1 之间的连接进行配置,例如,使得这些连接线路中的一半在一个 CPU 的控制下工作,而另一半在另一个 CPU 的控制下工作,这样 CPUs 之间就可以互相传输信息,或进行其他类型的通讯了。

[0343] 如图 85a,b 所示,为模块组与 CPU 连接的两种不同的排布方式,其中在逻辑树结构中的每一个模块都与 CPU 相连。在这种情况下,没有在逻辑通路上的链路没有处于开电状态;但是也不排除使启动这些链路。控制和返回通路也可以使用不同的逻辑树结构。在上述树结构中,可以根据单个模块的使用情况分配数据和控制流上的负载,也可以重新分配链路的逻辑关系使得可以对 I/O 设备进行访问。

[0344] 如图 86a 所示,为六边形排布模块组,其中每一个六边形都有一个单独的 CPU(CPU1, ..., 4),其控制通路用粗实线表示。如图 86b 所示,为四边形排布模块组,其中每一个四边形都与一个单独的 CPU 相连,为了表示清楚,每一个 CPU 的逻辑控制通路用不同的

符号表示（分别为不同的实线、虚线和点线）。在上述图示中，各 CPU 只可访问与之相关的存储元件组，但是 CPU 没有与其他计算设备共享访问通路。而在 87b 中，扩展了控制和返回总线使得上述总线可以访问共享的存储器区域。对到共享的存储器的访问的控制可以在树结构中进行，或通过不同通讯通路上的 CPU 之间的通讯进行，或通过其他方法进行。对于 CPUs C1 和 C2，其中一些线路的宽度只有之前所示的线路的一半，如前所述，这种配置可被两个处理器同时进行访问。也可以对处理器进行配置，使其可以选择性的访问共享的存储器。

[0345] 如图 88 所示，为具有 4 个 CPU 的模块网络结构的示意图，其中，上述模块可以是存储器模块，还有 4 个模块可为输入 / 输出 (I/O) 设备（分别为 I/O#0 到 I/O#4）。在本例中，每一个 CPU 都可以在不需仲裁的情况下同时访问任何一个 I/O 模块。

[0346] 如图 89 所示，其中 I/O 模块由其他的计算机系统元件代替，上述元件可为用于连接多支路总线的总线桥 (bus bridges, BB)，其中上述总线可以是 PCI 总线、点对点总线（如 PCI 专线）；也可以是与多支路的存储器总线连接的存储器网络集线器 MH#0 的接口，以及与点对点总线连接的存储器网络集线器 MH#1。图中线路的宽度是不同的，这表明不是所有的通路都必须具有相同宽度，也不需要通路之间是对称的。

[0347] 如图 90 所示，为同时支持线性和 / 或树结构布局的模块和其链路的排布示意图，其中具体为何种布局取决于使用哪一组连接线迹。

[0348] 如图 91 所示，为使用图 90 中的中央连接线迹形成的线性布局的排布图。

[0349] 如图 92 所示，为使用图 90 中的部分中央连接线迹形成的树结构布局的排布图。

[0350] 如图 93a 所示，为一组 RAM 芯片和一个 CSE 在模块 M 上的排布图，其中 CSE 位于模块的两侧，并提供从模块控制器到 RAM 芯片的连接。RAM 仅仅是各种存储器类型中的一个例子，此处也可以使用其他类型的存储设备或技术来代替 CSE。

[0351] 如图 93b 所示，其中 3 个 CSE 以“位片 (bit slice)”的方式提供了图 93a 中的功能。

[0352] 与图 93a 类似，图 94a 为在模块上布有一组 RAM 芯片和 CSE 的排布示意图，其中连接到 RAMs 的数据线用实线表示，地址 / 命令线用虚线表示。在各实际应用中，模块的大小以及在板上的位置都是可变的。

[0353] 如图 94b 中所示，模块上设有几个 CSE，其中 RAM 发出的数据通过一系列较小的 CSE 被分成位片。虽然位片 CSE 也可以控制地址 / 命令线，但控制到 RAM 的地址 / 命令线的 CSE 为在模块最右边的 CSE，并且该 CSE 也可以从一些 RAM 芯片获取数据，。

[0354] 如图 95 所示，其中树结构的底部的叶子模块 (M6 和 M7) 上的信号链路的相对物理位置是相同的，在这样的布置下可以进行多种优化，如使用仅有一组链路和 / 或连接器插脚并且只能与一组链路相接的 CSE 建立叶子模块，也可以从其他具有较少的功能的模块中获得其他的好处。

[0355] 图 96 中的左侧与图 95 中的左半部分的模块组的情况一样，其中还显示了模块 M6 相对于其他线性对齐的平行模块的偏移。在图 96 的右侧中，其模块还是一样，但是对连接线迹进行了调整以减少模块 M6 的偏移量。

[0356] 如图 97 所示，为使用双向链路的模块的排布的示意图。

[0357] 术语“模块”可代表 CSE 以及其相关的功能和 / 或元件，即使“模块”可能布置在

相同的板子上,而另一“模块”不一定要与模块控制器和 / 或其他模块通过可插拔的连接器相连。CSE 之间的连接用板子上的连接线迹来表示,但是这不代表排除其他通过电线、电缆、光连接、底层连接线迹以及其他类似方式进行的连接。

[0358] 如图 98 所示,为之前描述过的模块,其中模块中增加了高速缓冲存储器(该缓存可以采用 RAM 或其他类型的存储技术)。上述缓存可以只是 CSE 的一部分,也可以是整个模块,用于对来自或要发往模块控制器或任何其他模块的数据在通过本模块时进行缓冲。缓存的使用可以有利于增加数据流的时间灵活性,而且,如果存储了合适的数据则可以对数据流重新配置以解决如模块故障等问题。

[0359] 如图 99 所示,为一个 CSE 桥接在多个可能的不同互连系统之间的示意图。图 99a 中, CSE 连接了在模块的顶部边缘处的连接。图 99b 中, CSE 连接了两个处于模块底部边缘的连接。

[0360] 在另一方面,还需要考虑系统通讯、计算和存储能力的稳定性。数据或功能可能会由于一个或多个元件的故障而丢失或丧失,因此需要相应的考虑单个器件或元件的稳定性和系统结构的稳定性。应当避免单个点的故障,如计算机处理器、存储控制器、节点、存储器模块或类似元件的故障。系统的总体配置中提供了数据错误修正和对被破坏数据复原的冗余技术,比如检验和,循环冗余码检验 (cyclical redundancy check, CRC), 错误修正码和 RAID 等。术语“RAID”字面上为“经济磁盘的冗余阵列 (Redundant Array of Inexpensive Disks)”的缩写,但是现在 RAID 在本领域中还代表任何与从存储元件上的数据中去除数据和校验比特的技术有关的冗余技术,这也是在本例中 RAID 所代表的意思。在此处所描述的通讯、处理或存储结构中均具有数据修正功能和冗余连接,这样在整个系统完全不能工作之前至少可以承受一次以上的故障。上述故障可以在比如数据丢失之前被检测到,并在数据丢失前识别出故障的元件或设备并替换故障元件或设备。一方面,设备、元件或模块在被替换时可以仍然保证电力供应,这样上述设备在整个修理过程中仍然可以保证正常执行其功能,这种修理方式在本领域中通常称为“热切换”。

[0361] 如图 100 中所示,为一个二叉树结构的示意图,其中,模块的物理排布方式为线性的。模块之间的通讯线路或链路或总线用两端都以箭头结尾的线条表示。当总线通过模块但并没有与其连接时,总线的这部分用灰色的线条表示;即只有当总线或链路上的箭头与模块接触时才代表上述总线与该模块之间相互连接了。在本例中,每一个端口都分为两个通道,用模块的第一侧上的一条总线和模块第二侧上的两条总线表示上述通道。

[0362] 当描述模块的树结构时,用数字表示的“层”来表示树的结构以及描述模块所处的位置是比较方便的。在图 100 的下面部分,显示了 5 层二叉树中模块之间的逻辑连接关系。树中的根节点为一个单独处于第 0 层的模块,用模块 0 (M0) 表示。根模块可能通过一个端口与存储控制器 MC (本图中没有显示) 相连,通过另外两个端口与位于第 1 层的另外两个模块 (M1 和 M2) 相连。在二叉树中使用的模块都具有三个或更多的端口,如前所述,上述端口可以又分成一个或多个通道。在本例中,一个端口被分成了两个通道。还需要注意的是,在本结构中可对软件和硬件进行设置使得在一个模块的任何两个通道之间都可以进行通讯,其中端口的具体指派不影响上述通讯,其主板的信号线可连接至处于每一个模块上的不同位置的连接器上。在本例中,直接相接的模块之间可以用两个通道进行通讯。

[0363] 在图 100 的上部分中显示的为在线性布置的每个模块的下方,数字序列中的第一

行 M# 表示的是模块的数字标号,第二行数字 L 则表示模块在树结构中所处的层。在本例所示的排布方式中,处于第 0 层的根模块 0 布置在模块组的中间,总线位于图的顶部并连接至如存储控制器 MC 或 CPU 或类似的设备上。在模块 0 的两边是位于第三层的模块 10 和 12。而属于第 1 层的模块大约位于根模块和图中最左边或右边的模块(即物理上线性排布的分别处于最边上的模块)的中间。在本图所示的情况中,相同层的模块并没有排布在彼此的对面,这减少了热点,有利于存储阵列的冷却。

[0364] 在本例所示的树结构中,由于仅有模块 0 是直接和 MC 连接的,所以所有的数据最终都会通过模块 0。这意味着模块 0 的线路将会比其他模块的线路更经常处于活动状态,而且模块 0 将会具有较高的功耗。为了方便地讨论功耗问题,假定所有的模块都出现,且读和写命令平均分摊到每一个模块上。则第一层模块将会接收和发送根模块处理的所有数据的一半。这意味着第一层模块的线路处于活动状态的频率平均为根模块的一半,这使得第一层模块的功耗相对较低,简单考虑的话就是根模块功率的一半。类似的,第 2 层模块的功率消耗则为根模块的四分之一,第 3 层模块的功耗为根模块的八分之一。因此,除了功耗最大的根模块以外,其他各模块的消耗相对较少,在如图 100 中所示的排布情况下,根模块的两边为第 3 层模块,该模块的功耗仅为根模块的八分之一。这样的排布将会有利于根模块的冷却,而其他模块只有较少或没有热点其冷却将会不那么困难。在此之后将会讨论具有多根节点的树以分散系统吞吐量的情况,对本领域的普通技术人员来说很容易理解到每一个根节点的功耗将会减少为原来的一半。

[0365] 如图 101 中所示,还有许多可用于二叉树的连接方式,在二叉树中每个模块有 3 个端口,每个端口被分成两个通道。在图 101a 中,通道相对于线性排布的模块的中线对称分布。可选择的,如图 101b,模块之间仍然具有相同的通道和端口,但是连接两个模块的两条通道之间有一定距离,这使得另一模块的通道至少有一条是与其他通道间隔开来的。

[0366] 对于具有多根节点的树来说,其第 0 层有一个以上的模块。在这种情况下,每个树中的各个根模块有不同的数字标号。在图 102 中,对第一个根节点其树结构中各模块的数字标号 M# 在图的顶部,而第二个根节点其树结构中各模块的数字标号 M## 则位于图的底部。比如,一个模块可以标为 0 或 11,这取决于这个模块是作为树的根节点还是节点。

[0367] 模块之间的线路或总线常常受到针脚密度、连接线迹或线路密度、串扰或其他类似因素的限制。发送和接收功能的分离,以及线路的分离将会有利于解决上述问题。如图 103a 所示,为连接一组模块的线路的一个例子,在俯视图中可以看到在模块组中其他模块相对于中间模块对称排布。图 103b 为在印刷线路板 PCB 实现上述总线连接的正视图,其中上述 PCB 板有 3 层及 4 个表面。总线可以方便地进行排布并避免或减轻线路交叉、多路由、盲路由以及类似的问题。

[0368] 在连接的主板设计中,连接器上的针脚穿过 PCB 板并与每一层的连接线迹相连,这样每条总线的垂直部分实际上为连接器针脚。连接器针脚通常都具有相同的物理长度,即使在有些层中不需要进行连接上述针脚仍然会穿过电路板的所有层。当与时钟和信号数据率相关的频率较高时,这样的针脚可能会成为伪信号的辐射点。

[0369] 虽然可以使用多层 PCB,但是并不排除部分或全部的总线连接排布于单独一层上。同时此处描述的排布方式的各部分也可以有许多其他的排布方法。例如,包括 CSE 和存储器在内的部分或全部的元件和互连可以与半导体电路结合在一起,冲模可以处于互连底层

或底层或类似结构上。则产品的结构和封装可能与此处所举的例子完全不同。

[0370] 如图 104 中所示,为一多根节点的树结构的排布示意图,所示结构中有冗余保护,由于提供了额外连接线迹,且各模块的软硬件可对路由重新配置,使得当某一个根模块故障时系统仍然可以正常工作。图 104a 中为树结构的基本连接情况,其中实线表示基本连接中的总线,图 104b 中所示为其逻辑连接情况。在图 104a 中,每一个方块表示树中的一个存储器模块,并将其布置在相应的存储模块之下。其中 M0 为根模块并与存储控制器连接。

[0371] 如图 104a 中所示,本系统的额外互连用粗虚线表示,当系统出现故障时可能会使用到上述额外连接。当与 MC 连接的根模块 M0 故障时,该树结构可以重新配置成如图 104c 中所示的结构。此时,模块 M0 故障前的部分线路将不会被使用(如 M0 和 M2 之间的线路),而此前处于非活动状态的线路将会被启用(如 M4 和 M5 之间的线路,图中用粗虚线表示)。相应的,树结构中的各模块所处的层数可能有所改变。例如,原来为第 0 层的模块 M0 现在在第 3 层,而原来在第 1 层的 M4 现在处于第 0 层。虽然图 104 中所示的结构中提供了一些冗余,但本结构中并没有为任何节点或线路的故障提供冗余替换结构。

[0372] 如图 105 所示,为一个可以适应任何单个地方故障而不丢失故障元件的数据的树结构的连接的示意图。其中保留了到所有模块和所有可操作的链路的连通。如图 105b 所示,为树结构的逻辑连接关系图,其中实线链路线路表示当所有线路和节点均处于正常状态时的树基本结构示意图。而虚线表示原处于非活动状态的线路,当根模块 0 故障时其会变成活动状态。此时,原处于第 3 层的模块 10 重新指派为根模块。如图 105b 所示,上述变化用从模块 10 发出的垂直向下延伸的虚线表示。

[0373] 如图 105a 所示,为主板上的线路和模块的俯视图,而图 105b 中为相应的树结构的逻辑图。用实线表示的链路为活动状态,此时树结构中没有任何故障。而用虚线表示的链路为当系统中存在一个故障时使系统重新配置后仍可工作的替代链路。本图中所示为根模块故障的情况,这种情况可能是最严重的系统故障,由于根模块为功耗最大的模块因此这也可能是统计上最可能会发生故障。在其他条件相同时,某一类型的半导体电路的故障率随温度的升高而增加。当树结构重新配置后,之前处于活动状态的连接并不都会继续为系统所用。

[0374] 树结构的重新配置将会导致数据的改变、控制通路和一些模块所处的层的位置的改变。如图 106 中的表示有助于使上述改变形象化。图 106a 中所示为系统中无故障的基本配置情况,其中辅助链路处于非活动状态(用虚线表示)。当模块 0 发生故障时,如图 106b 中所示,将树结构图沿水平轴对称翻转,这样模块 0 处于底部而模块 15 则位于图的顶部。如图 106c 所示,可操作的额外链路用粗虚线表示,而不可操作的链路则用细虚线表示。

[0375] 如图 107 所示,为一个较大容量的主板的示意图,其中有 4 个模块组,每组有 16 个模块。图示为一种进一步增加主板容量并保持冗余结构的配置。每一组模块标识为“第 n 组”(其中 n 为 0,1,2,3),并分别具有图 105 和 106 中所示的配置结构。四个模块组之间的连接由电路元件 C 完成,这些元件可以是 CSE 或类似元件,这样任何设备 C、或 CSE0 和 CSE1、或任何组中的任何模块的单个故障都不会引起数据的丢失。这种特征还使得模块可进行热切换,这样,可以对故障模块进行替换,而无需要将该单元退出服务。

[0376] 如前所述,模块的功耗取决于该模块在树结构中所处的层数,而层数也与各种情况下的平均数据吞吐量有关。对每一个模块所处位置的功耗进行测量,并更有效的使用系

统的冷却资源、减少系统内的热点。如图 108 中所示的例子,为一组有 15 个模块的模块组的俯视图,与之前的例子类似,图中还显示了 CSE 在模块上的位置。图中用宽箭头指示了空气流的方向,主板上还附带有金属隔离栅 1100 用来帮助将冷空气导向模块。从主板延伸出来的隔离栅与主板上的模块高度差不多,也是插在主板的连接器上。处于第 0 层的模块 0 具有最高的功耗,其两边的隔离栅在空气的流入方向具有较大的开口,因此与处于第 3 层的模块 11 和 10 相比,模块 0 处的空气流量较大。其他隔离栅布置的位置与之类似,但是隔离栅 1100 布置在合适的位置并使处于第 1 层的模块 1 和 2 比位于第 3 层的模块 8,9,12,13 能获得更大的空气量。

[0377] 在需要对一组以上的模块进行冷却,并且冷空气会在通过主板之前通过第二组模块时,对不同的模块可以使用相似的方法来控制空气流向以减小不同模块之间的温度的差别。如图 109 所示,为两组每组有 16 个模块的模块组的分布,其中,在第二个树结构中较高层的模块处于第一个树结构中较低层的模块的正下方,这样同一垂直方向上的两个模块所处的层数的和相对于两组模块同样排布时减少了。如果模块的层数的和越低,则两个模块组的功耗就越高。例如,离空气入口较近的模块组的处于第 0 层的模块 0 在另一模块组的处于第 3 层的模块 11 的前面,则这两个模块的功耗仅仅只有当两个模块组的分布一致(例如,两个都是第 0 层的模块处于与空气流向相同的一条直线上)时的两个模块功耗和的 60%。

[0378] 另一方面,由于模块在主板上所处的物理位置与模块在树结构中的层数有关,则模块之间的空间的大小可能不同,这样树中被认为具有较高的功耗同时处于较低层的模块(如根模块)相比于处于较高层的模块(如叶子模块)离邻近的模块要远一些。对于有一个以上根节点的情况,不论是在正常工作或冗余工作状态的情况下,在排布模块的位置时都会考虑其功耗情况。

[0379] 如图 110 所示,为处于 CSE 和 / 或模块上的 CSE 和 / 或散热片的不同放置和从不同的方向上观察的示意图,上述的布置方式可在空气流和 / 或模块空间方面提供一定的改进。

[0380] 如图 111 所示,为一个具有两个根节点的树结构的示意图,其中节点具有由 12 条线路组成的链路。根节点到 MC(图中没有显示)有 4 条线路,到其两个子模块各有 3 条线路,到另一根模块有 2 条线路。根模块的子模块有 3 条线路到其根模块,上述子模块到其两个子模块也各有 3 条线路,同时其还有 3 条线路到相应的另一根模块的子模块。

[0381] 大型存储阵列的物理布局存在很多问题,如需要较多的 PCB 层,在各层的导体上经过或绕过的线路构成连接线迹,或在底层上的类似问题。连接器的插脚引起的电磁辐射情况,接地和电源平面的完整性的效果都是本领域的普通技术人员所能了解的。如图 112a 所示为 PCB 的一个单层的膜,该 PCB 与 4 个连接器相连并与板上的接口连接器或已有的存储器相连。连接线迹的密度以及线迹之间距离的变化都可能与信号的耦合和传播时间的变化有关,且直线线迹的偏离将会导致高频信号下电阻的变化,而电阻的变化将会导致信号波形的失真。如图 112b 所示,为图 112a 所示的层的布局的细节的一个例子的示意图。圆圈代表连接器的插脚,其中黑色圆圈代表与所示层上的线迹相连接的插脚,而浅色圆圈代表穿过该层与另一层上的线迹相连接的插脚。在远离插脚的地方,线迹与相邻线迹之间的距离各不相同,其中一些线迹之间比较近,这增加了信号之间的耦合。在插脚的附近,插脚

之间有成组的线迹,线迹之间以及线迹和插脚之间的距离都变小了。在这个区域中的许多线迹都存在多次的弯曲,这导致其电阻不连续。如图 112c 所示,为两个连接器 A 和 D 之间的连接所通过的通路的示意图,其中,连接线迹必须绕过介于 A 和 D 之间的连接器 B 和 C 的插脚。图中所示的线迹对代表了使用了不同信号情况下的布局。同样还显示了第二层的布局,其中信号可顺着通路传播会使一层上的信号线迹的一部分更靠近邻近层的信号线迹。这样会在板层布局中引起各种问题。

[0382] 如图 113 所示,为一种减轻电路板布局的限制的方法的示意图。在本例中,将板上每一个连接器 1200 进行了旋转,使其与多数存储器板上平行排布的连接器(或类似应用的设备)之间形成了大约 30° 的角。当连接器以图示方法成歪斜排布时,线迹 1300 可以没有或只有很小弯曲的情况下在绕过连接器插脚 1400 并布置在相邻的连接器之间。上述歪斜的角度取决于插脚行的数目和各行之间的距离。在这种排布下存在一个以上的最小弯曲通路,如图 114 所示,为在一组具有 240 个插脚的连接器上具有最小弯曲的两条通路。相邻板子之间的空间或是插脚空间的一个很小的改变可能可以进一步的减小线迹的弯曲程度。

[0383] 如图 115 所示,为可容纳 84 个歪斜放置的 DIMM 的主板上的连接器的排布图。当需要排布很多连接器时,板上的空闲空间将缩减到最小,而上述的空闲空间通常会用于布置板子的接口,存储控制器,辅助设施,滤波器或类似的设备。上述连接器的歪斜方式可用于本具体实施例中所描述的任何连接器和模块的排布。为了表示的简便,在其他例子中的连接器还是以水平方式排布。

[0384] 大型存储器板可以使用本处描述的设备和方法进行排布,如图 116 所示,为 82 个 DIMM 模块的排布图,其中 64 个为具有 RAID 等级 5 错误纠正能力的 DIMM 存储器模块,上述 RAID5 以 4+1 方式配置,并具有 2 个备用 DIMM 模块。存储器板支持 2 个树结构模块组,并与树的叶子节点连接。对剩下的数据和控制通路通过合适的重新配置后,树中任何两个模块的故障都不会影响对其他模块的访问。因此,在数据不可恢复的丢失前需要对存储器板进行维护,并将故障模块通过热交换换成新的模块。热交换发生在仅当一个 DIMM 失效时,上述配置防止了当维护技术员将故障模块替换走时的数据丢失。在本例中,图 116 中为逻辑排布图,图 117 中为相应的在板上的排布图。

[0385] 在图 117 中,实线表示基本树结构中的链路,虚线表示在基本树结构中没有用到,但当某一模块故障时对树结构进行重新配置后在保证根和剩余模块相互连接的链路。虚线所示的链路的通道吞吐量可为实线链路的一半。在实线链路的树结构的第 0 层,每一个树都有一个根节点并与存储控制器(图中没有显示)相连接。另一方面,所示排布可以认为有 4 个子树,其中每一对子树组成一个树。在上部的树中的每一个子树都是另一个子树的镜像对称,其中对称线通过子树的根节点。位于图中下部的树结构相对相关的根节点也有相似的镜像对称结构。每个树中作为冗余结构提供辅助链路,两个树结构中的叶子节点通过辅助链路相互连接,并指定与 MC 相连的作为冗余树结构的根模块,其中该 MC 可以是冗余 MC。

[0386] 如图 118 所示,为另一个具有两个根节点的由 82 个模块组成的树结构的示意图。上述树结构将作为一个例子以描述大型互连系统中的主板上的排布技术。上述技术可以用于底层、半导体电路内或冲模的相互连接。术语“主板”的使用仅仅是为了描述的方便。

[0387] 其中,每一个树结构的深度为 5,第一种配置方法中使用根节点 R1 和 R2 与存储控

制器 MC 连接。冗余配置方法中使用 R1' 和 R2' 中的一个来代替故障的根节点以继续进行操作而不丢失数据。图中实线表示第一种配置方法中的链路,虚线则代表了当两个或少于两个的模块故障时的替换配置中的链路。主板上的排布可以认为是两个相似的相互连接的树结构。与 R1 用实线连接的模块为上部的树,而下部的树则为与 R2 用实线连接的模块。

[0388] 如图 119 中所示,为上部的树连接的物理排布和相应的树结构的逻辑排布的示意图。模块 R1 在第 0 层,在本例中处于连接器行的中间,并且与存储控制器 MC 通过一个链路相连。图中没有显示 MC,该 MC 可以布置在任何合适的位置,可以在主板上也可以不在。本二叉树中模块的连接用阴影表示,共有三层阴影。为了便于表示,上述阴影分别用不同的颜色表示,蓝色(图中为最深的颜色)、红色(最浅的颜色)和绿色(介于上述两种深浅之间的颜色)。每一条宽的阴影线代表处于其末端的两个模块之间的一条链路。穿过模块的下面而没有在该模块终结的线条则代表没有将该模块进行连接。上述链路具有之前描述的链路的特性,而宽线可代表双向线路,单向线路组,通道化的线路,以及用于模块和控制器间连接的类似物。

[0389] 大部分的链路都用直线表示,只有很少甚至几乎没有链路需要穿过其他的链路。这种简化的板子的排布方法可以减少信号的耦合,并减少 PCB 板的层数。图中分别标记为 a 和 b 的链路在树中用于连接根模块和第一层模块,该链路用绿色标记。通过对板上有代表性的方面的描述,本领域的普通技术人员可以明白链路 a 和 b 可以以更短的长度和更少的弯曲布置在 PCB 的红色层。此处仅仅表明可对排布和相互连接可进行简化,板子的排布方式是可以被进一步优化的,上述排布和相互连接是指大量用连接器连接的存储器的相互连接和排布、或布置于电路板的存储器的相互连接和排布、或与底层上与集成电路集成的或封装在一起的存储器的相互连接和排布。

[0390] 如图 120 中所示,为主板上包括位于 1-1 之间的模块的横截面的示意图。三个相关的横截面 A、B、C 中表示了 3 组连接到各模块的链路线迹。其中,宽线示意性的表示了链路中的线路,并与均匀或非均匀分布在电路卡上的连接器的插脚相接。进一步的,不同颜色的链路可能彼此重叠分布,为了使描述清楚在图中将其分开绘制。图 120a 为主板的俯视图,图中指示了各横截面的具体位置。为了便于描述,将各电路卡用数字 0-7 标识,这种标识还同时适用于图 120 中的其他图。在 120b 中,显示了图 120a 中的横截面 A;横截面 B 显示在图 120c 中;而横截面 C 显示在 120d 中。在横截面 A 中,显示了蓝色、绿色和红色连接。如图所示,在本例中,印刷电路板使用了 PCB 技术并具有两个地平面层和一个 VCC(电源)层同时板上还布置有链路线迹,因此每条连接的部分线迹(线路)布置在了接地或 VCC 层的反面。这将可能增加传输和接收线迹之间的隔离程度。水平阴影线代表连接器之间的链路的一种可能的位置,上述链路彼此之间没有交叉,可以以直线方向进行配置。如图所示,每一条链路都终止于一个与模块的连接器连接的插脚,该插脚代表了一组插脚,而垂直阴影线与连接器插脚相连。

[0391] 在图 120c 和 d 中,横截面 B 和 C 的路由具有相似的特性。

[0392] 例子中使用了带插脚的连接器,但是也可以使用表面帖装技术(surfacemount technology, SMT)连接器。对于 SMT 连接器,向上接往连接器的水平线迹使用通孔而不是插脚进行连接。但是,并不需要每一个孔都完全穿过整个厚度以到达要连接的线迹所在的层,孔可以通过背钻去掉板层上的镀层,使该板层离连接器比离要连接的线迹更远。这减少了

垂直传导元件的长度,也就减少了线迹之间的辐射和耦合,同时避免了“盲”孔。此外,在互连中涉及的层的数目更少,则其中对校准可以有一定的公差,这可以使在地和 V_{cc} 平面上提供的救济最小化,也就增加了屏蔽效果。在本例中用三种颜色表示了用于链路的线迹,也可以仅用两种颜色对排布在板上的线迹进行表示,不过这将会导致增加一些线迹的弯曲。

[0393] 还可以有很多别的主板的布局方法,以下示例了其他两种布局方法。图 121 为 82 个 DIMM 的逻辑排布图,其中树结构排布成 3 行连接器。可以发现上述树结构中具有更进一步的冗余特性。

[0394] 如图 122 所示,为图 121 中所示的相应的逻辑树结构的排布图,其中使用了与图 119 相同的符号。对相邻连接器之间的空间进行了调整。例如,增加了根模块或冗余根模块之间的空间以适应当模块作为根节点时增加的能量分配。在行的中间的模块之间靠的更紧,以便为诸如模块控制器、CPU 或类似元件提供设置的空间。一方面,根模块可以被没有存储器的 CSE 或只有很小的存储能力的 CSE 代替以减少模块作为根节点时的能耗。

[0395] 可容纳 82 条 DIMM 的主板上布置满了 DIMM,其中 62 个存储器模块配置为 RAID 5 模式,若每条 DIMM 上插有 2G 的存储器,则主板的容量为 164G。这样的主板的尺寸大约为 15.4×21.4 英寸。当然,在类似的方法中也可以采用其他 DIMM 存储器和其他存储器类型。本例仅仅是为了描述,在本领域中还有许多不同的可用于 DIMM 模块的冗余方法,当存在如上所述的相互连接和操作时,DIMM 模块组也可用于实现冗余操作。

[0396] 如图 123 所示,为另一 82 条 DIMM 的逻辑排布图,其中树结构布局成 3 行连接器的形式,并利用如图 109 中所示的从前到后的冷却排布方法。图 124 所示为相应的主板上的排布图。

[0397] 主板上的连接种类只有几种形式,其中各种形式以及与模块的关系如图 125 中所示。其中标记为从 A 到 H 的单元代表了在二叉树结构中会使用的模块的类型。在上述单元中垂直的线表示了连接器,而垂直的线上的三条水平的短线分别代表一个链路或任何之前所描述过的链路的类型。链路中的线路可组合成组,并分成不同的通道或单独的路由,上述单元的种类仅仅表明了相对于其他连接器的链路的方向。此外,对各单元的标记并没有限制通过哪个侧面的连接器的链路线迹与存储器模块相连。图中所示为具有 16 个连接器的布局的例子,但是在该例中并没有使用全部的单元的类型。图中还表明有与特别的连接器相关的单元的类型。当连接器与树的链路少于 3 个时,即有些链路少于 3 的单元类型(图中没有显示),也可以进行图示的布局。

[0398] 以树结构排布的存储器,其时钟可能是本地的、区域的或系统的时钟、RAMBUS 时钟或类似的以及时钟技术的组合。举例来说,每一个 DIMM 模块都有本地时钟用来管理存储器和完成本地处理功能。中央同步时钟可用来将中央时钟传递给存储器模块组,而这些存储器模块 CSE 的时钟的相位可能并不知道,但是假定该相位的变化是很缓慢的。在上述情况下,当一个链路处于活动状态时仅需要对相位进行重新同步化,则启动时间将变得较短。可能会需要重新获得链路的每一条线路上的时钟,但是一组线路的相位会比较相近并且可作为一个组进行调整。另一方面,时钟可能会分配在地址线路上,上述线路中的一条在所有时间和所有链路之间将会保持活动状态。可替换的,当数据、命令或地址数据的间隔时间没有用于在一预先确定的时间段向邻近模块发送或接收数据时,线路中的一条可能会用于传输诸如 NOP 命令的信号。

[0399] 另一方面,一个存储控制器 MC 一方面可接口到一个或多个根模块,并且另一方面可直接或间接的与其他的通讯总线或计算元件连接。在 MC 和根模块之间的接口可以通过根模块用来与树结构中的其他模块通讯的链路类型来连接。可选择的,在 MC 和树中的模块之间可以有一个或多个 CSE, CSE 的其中一个完成根模块或单个节点的功能,这样在树中其 CSE 具有最高的通讯量的点可以为具有较低功耗的模块,举例来说,MC 可以完成去歪斜、寻址、帧控制、数据和命令在上行和下行方向的缓冲、初始化(包括歪斜判定、模块完整性和其他接通测试)以及当模块故障时树结构的重新配置的功能。MC 也可以管理开电状态下的功耗,使得板上的存储器的能量的最大消耗量与相关的能量供应能力相适应。

[0400] 虽然对本发明提供的方法、装置以及系统通过上述例子进行了描述,但是本领域的普通技术人员应当可以理解,本发明并不局限于上述的具体实施例,还存在在不偏离本发明的思想下的对本发明的不同的改变或修改,这些改变和修改应仍然为本发明所涵盖。

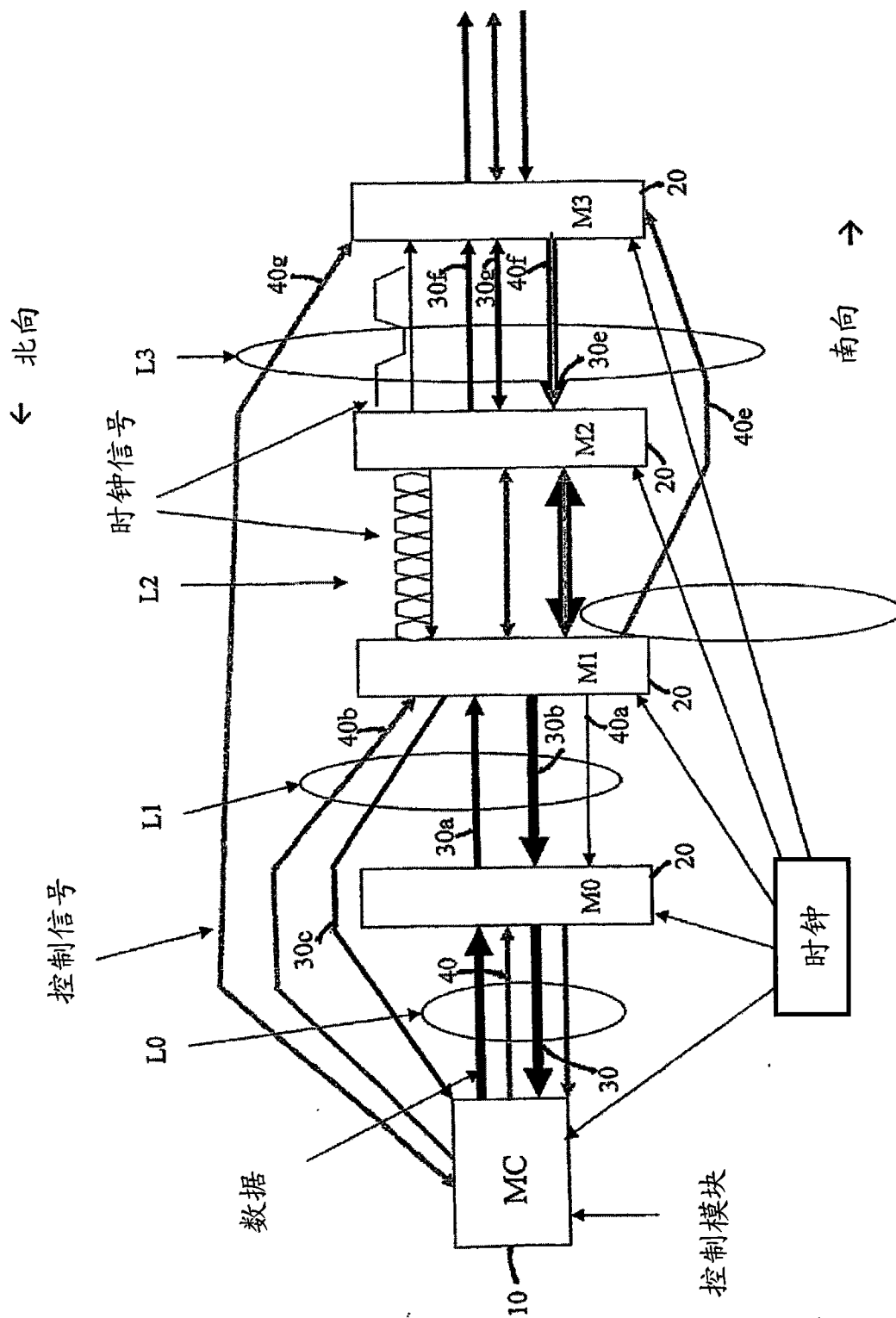


图 1

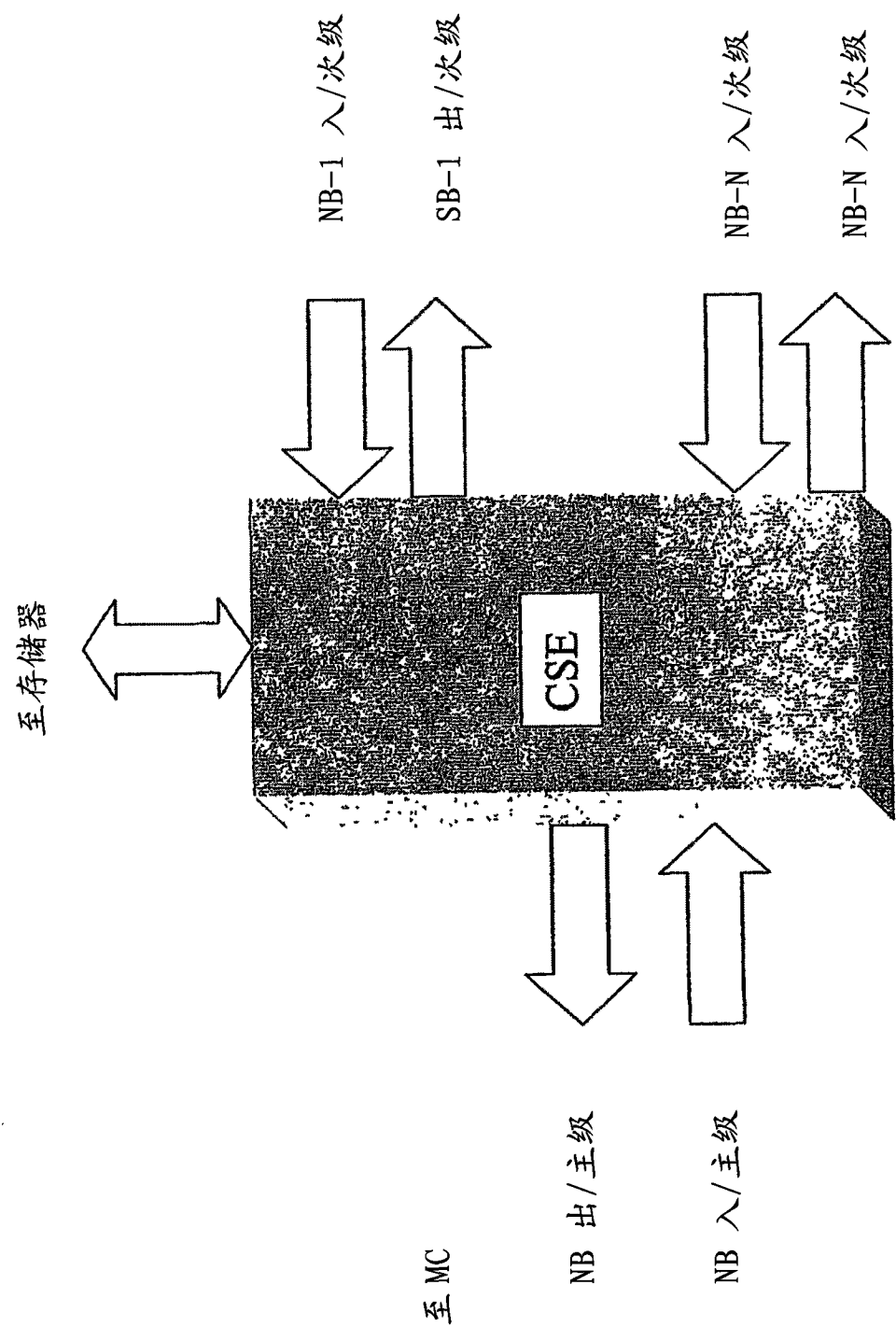


图 2

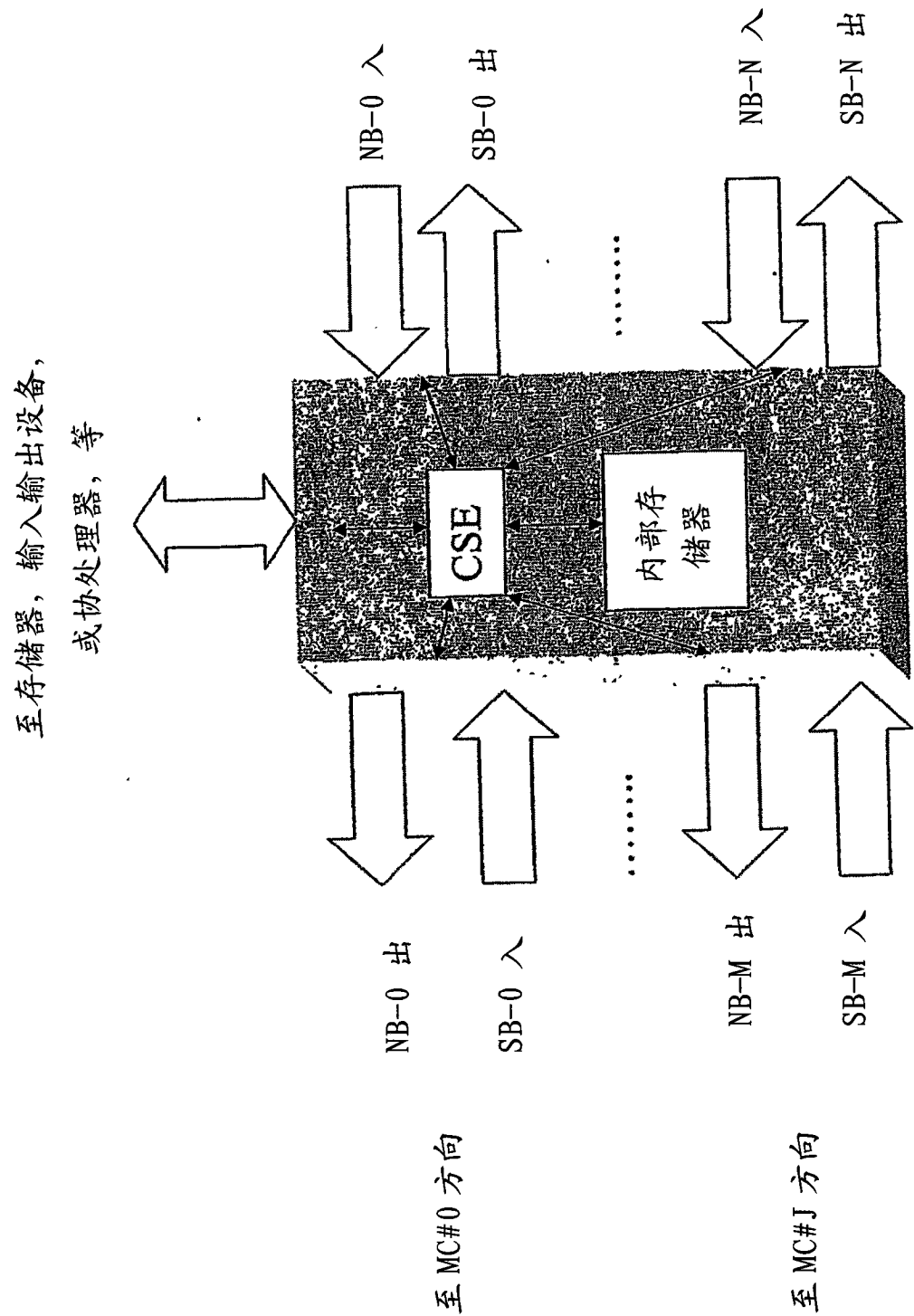


图 3

至存储器, 输入输出设备,
微处理器, 图形处理器, CSE,

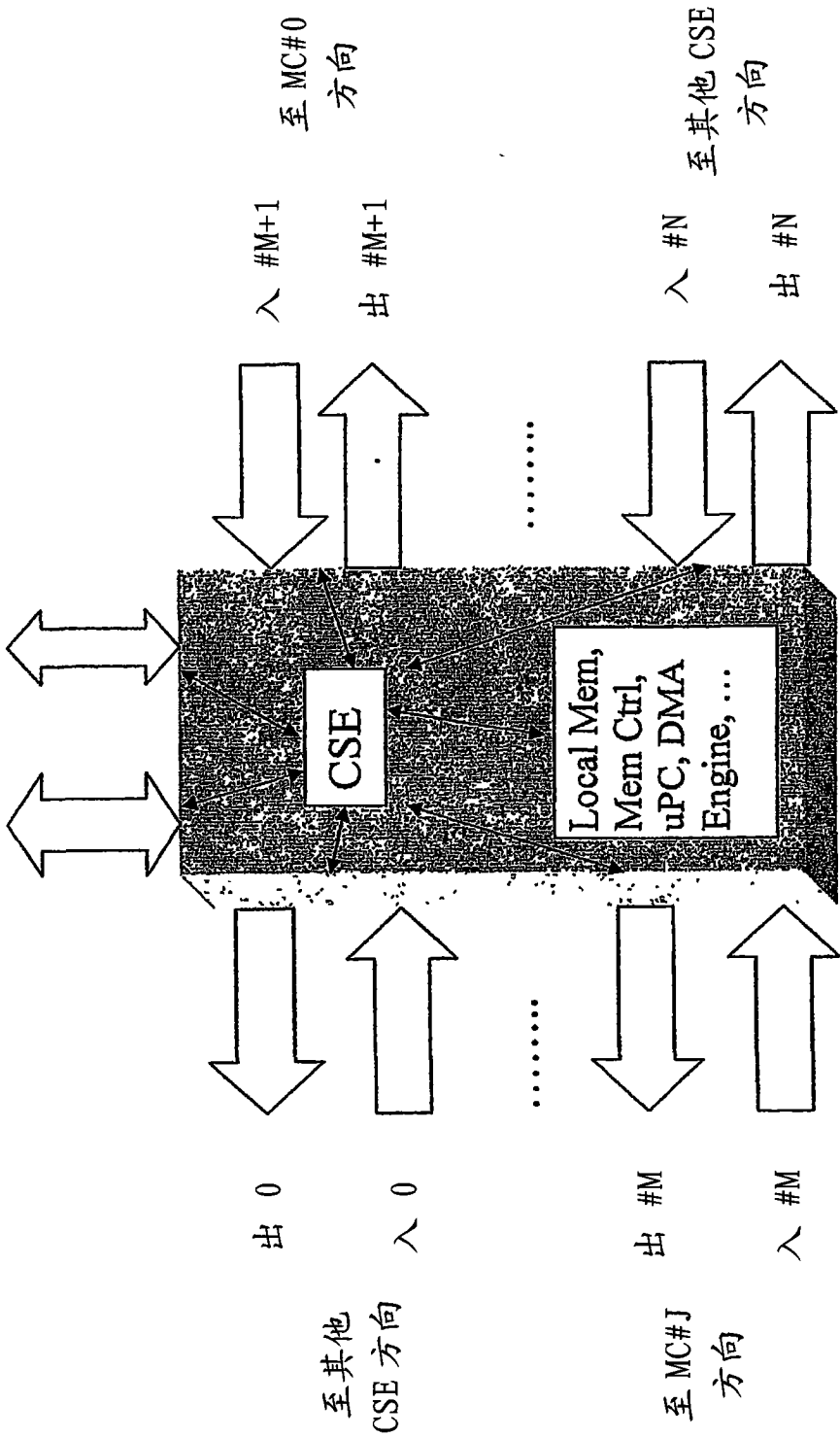


图 4

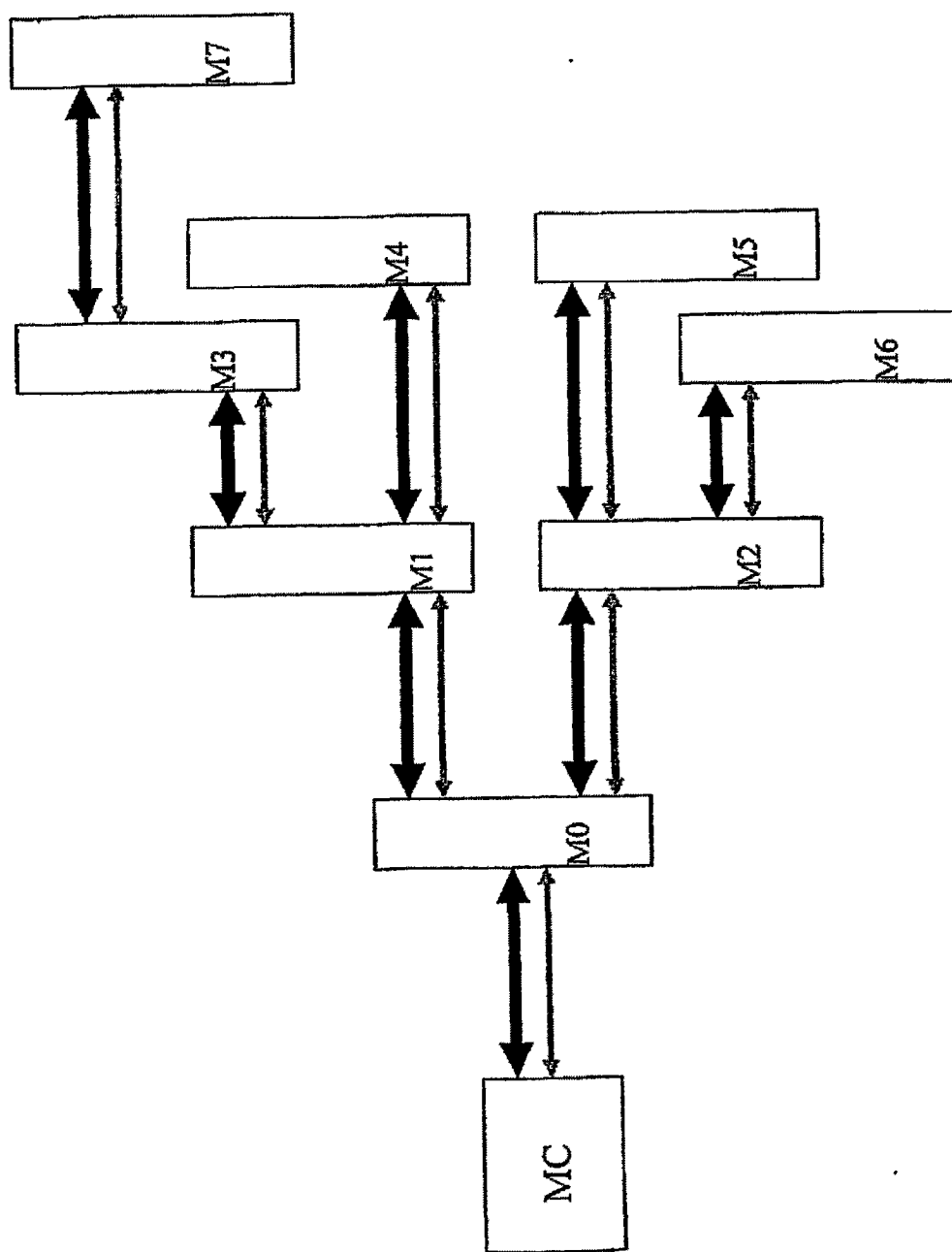
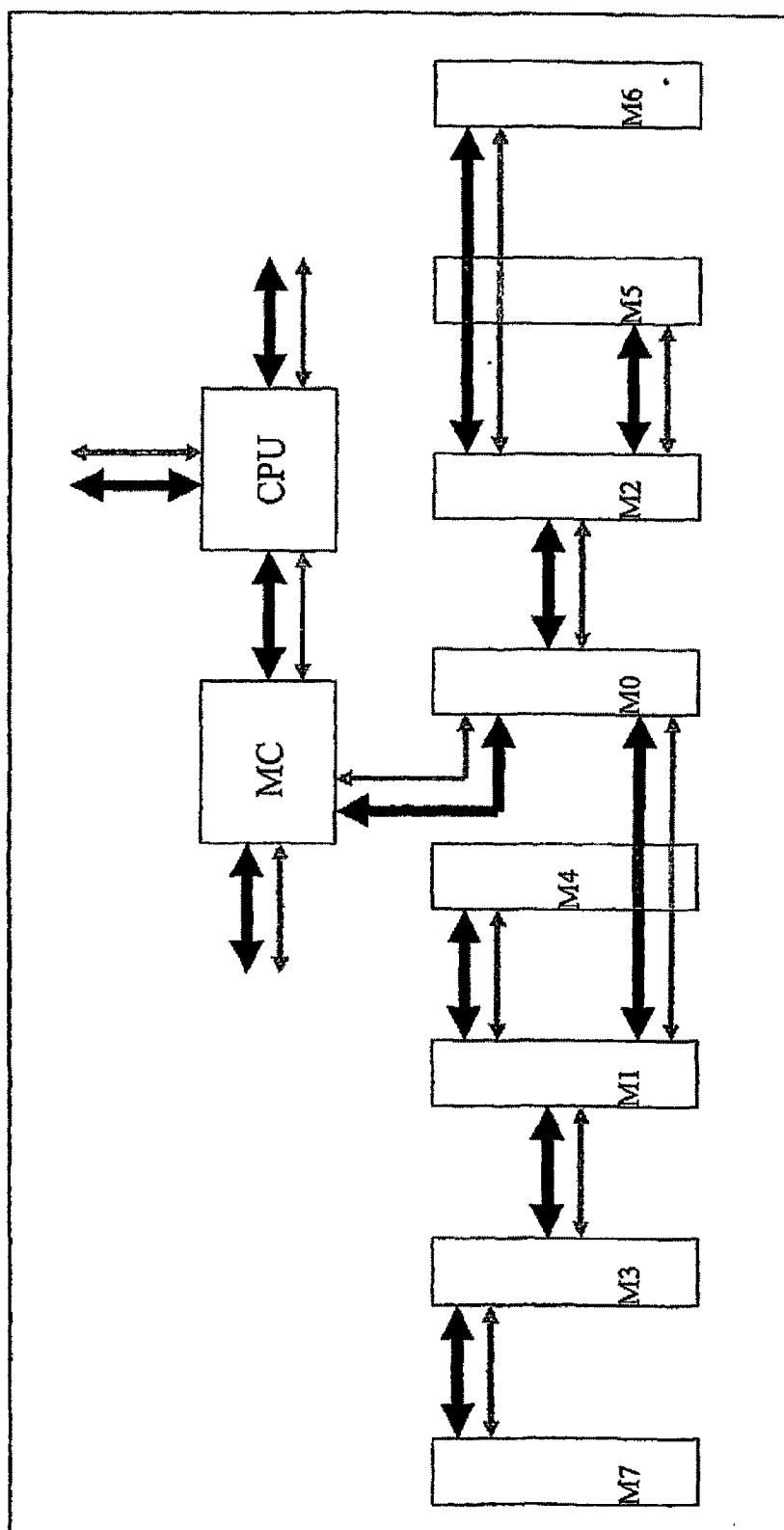


图 5



6

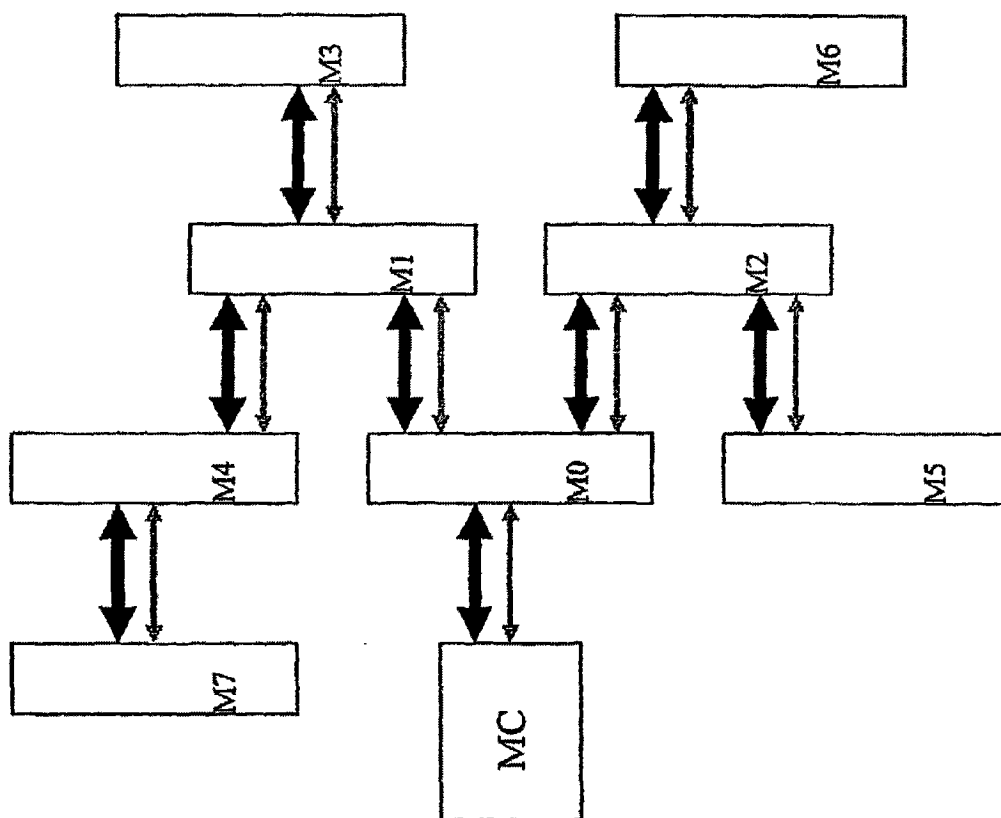


图 7

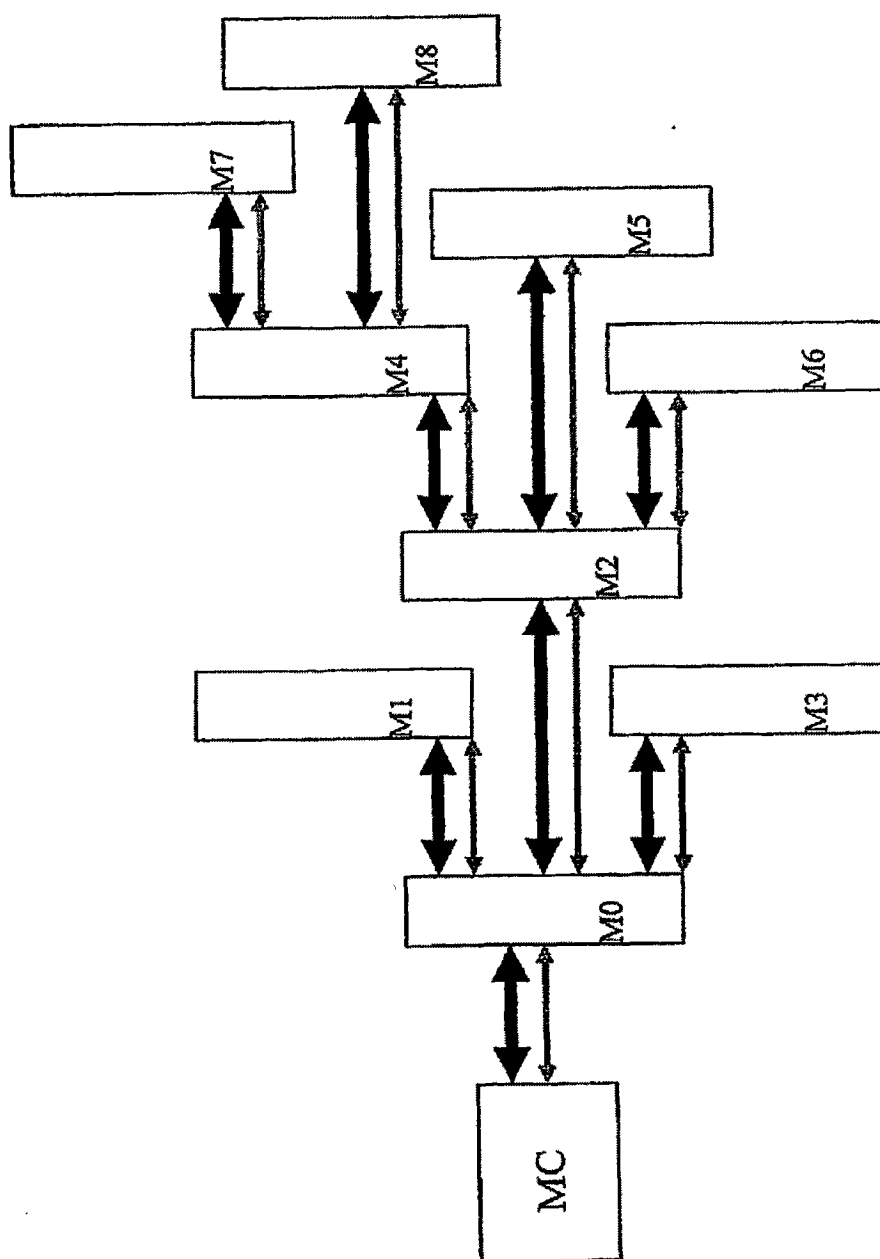


图 8

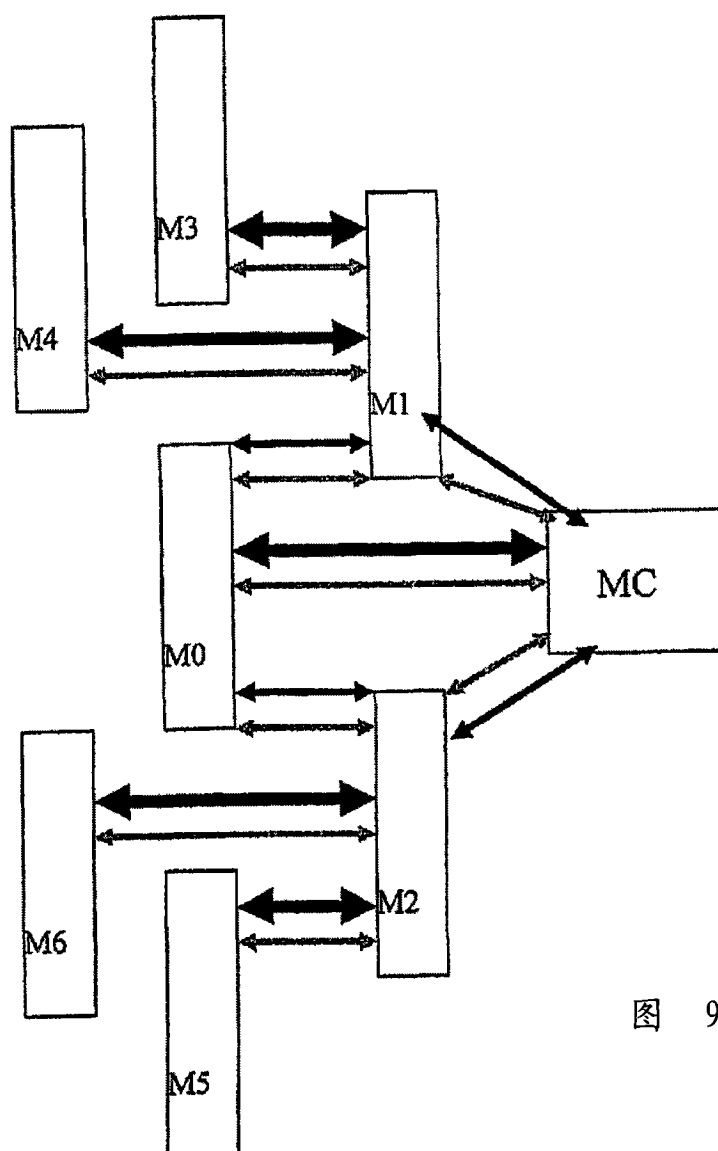


图 9

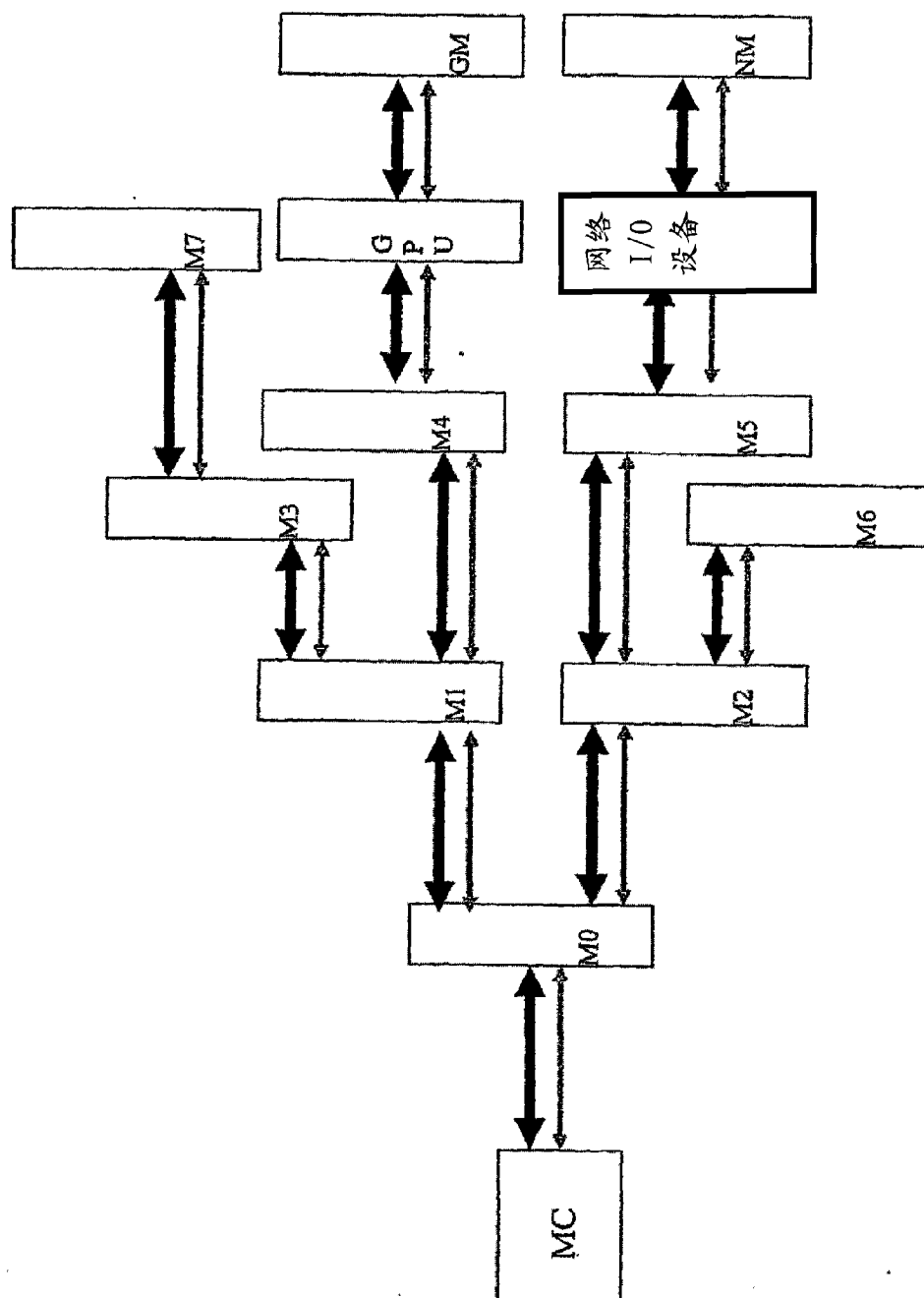
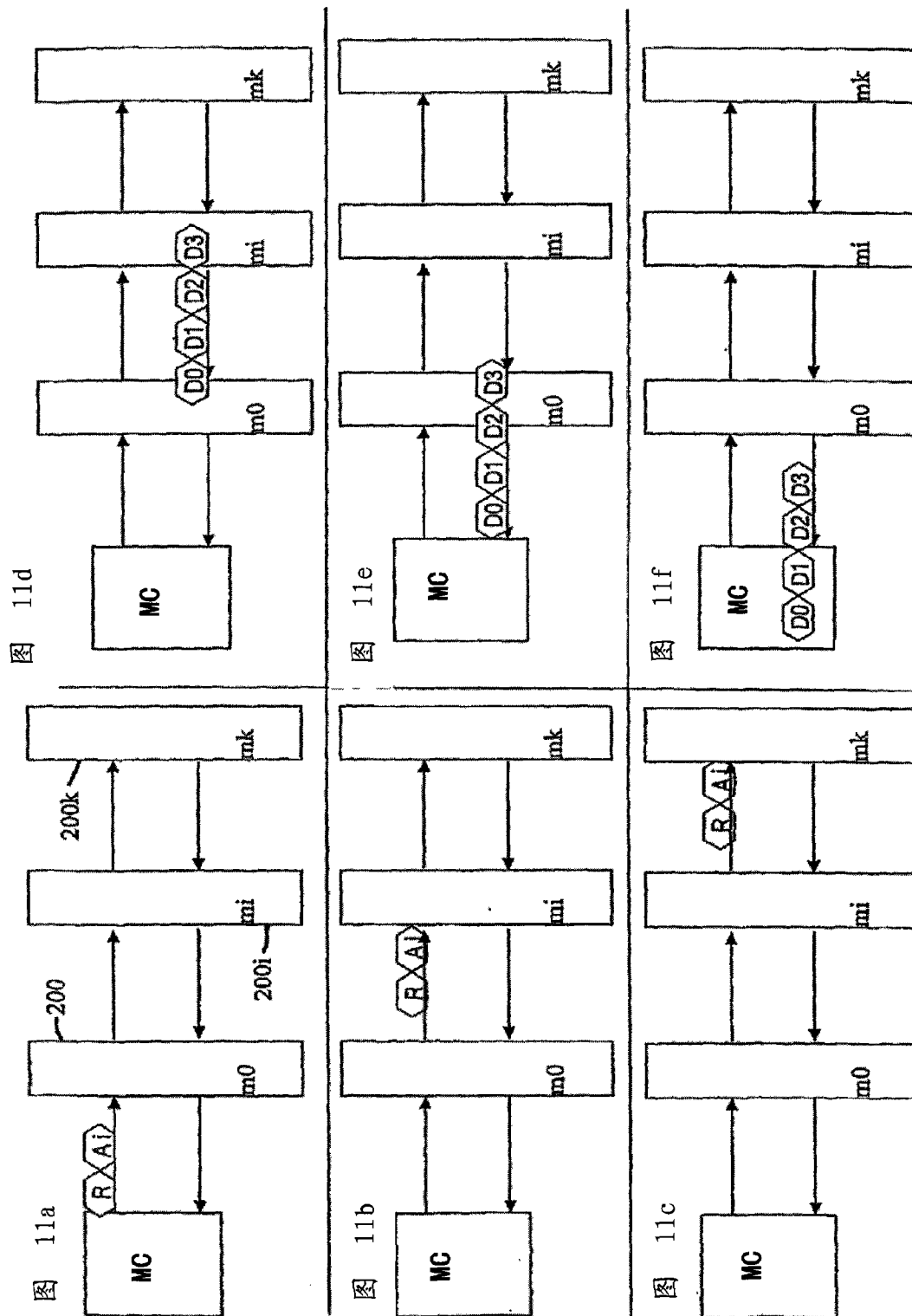
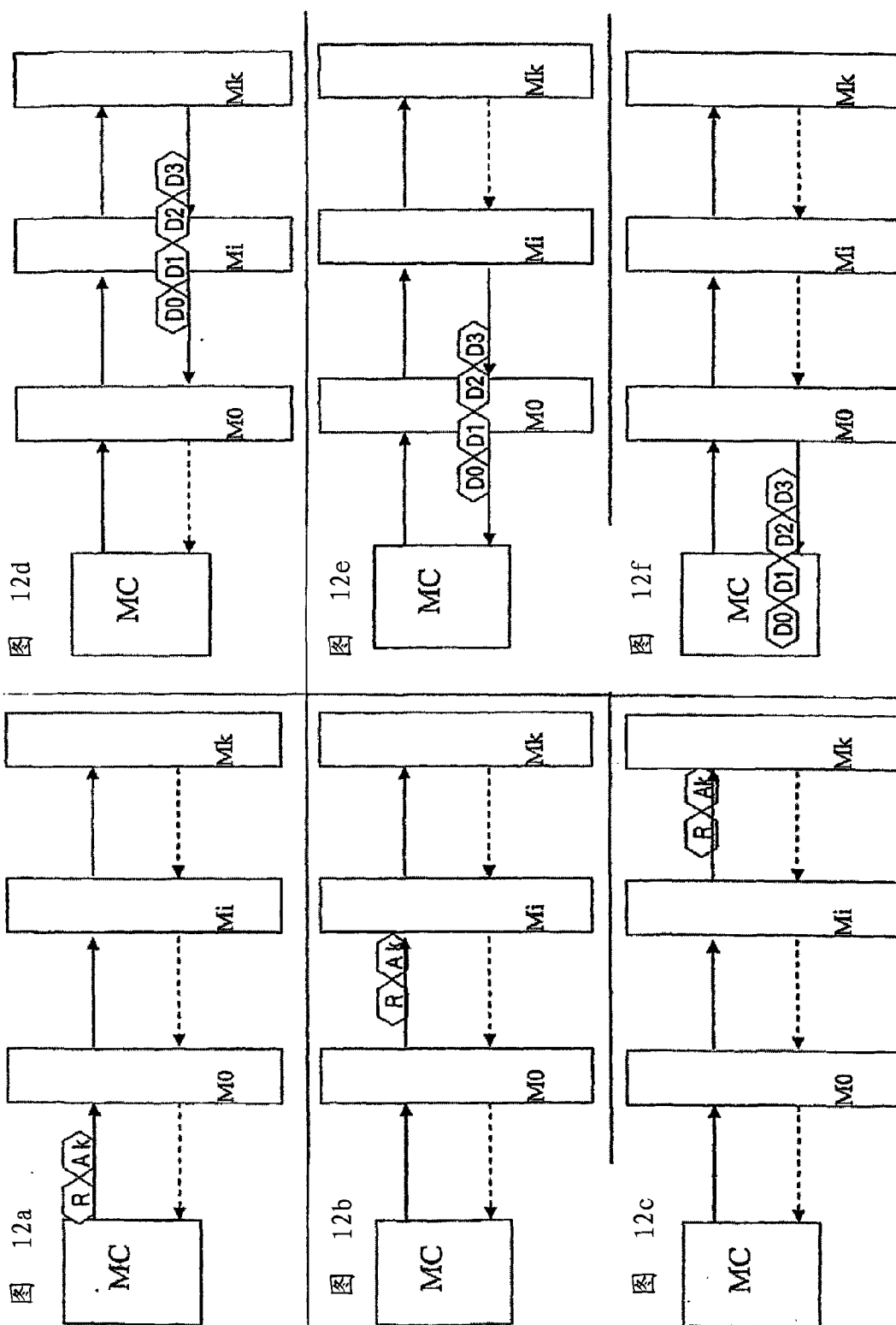
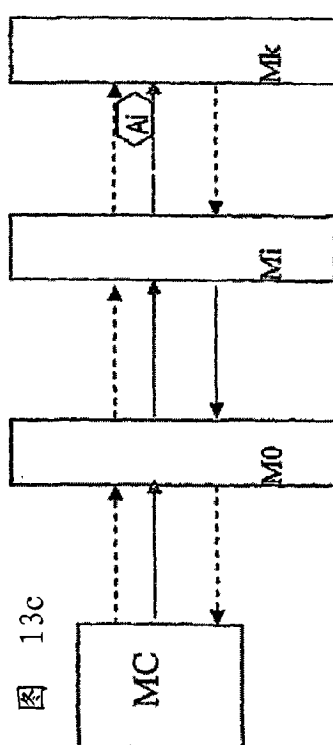
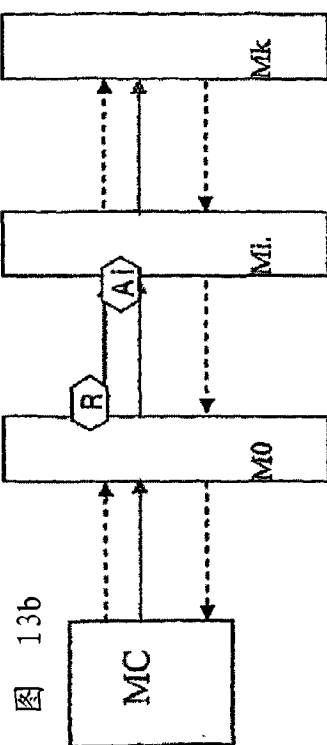
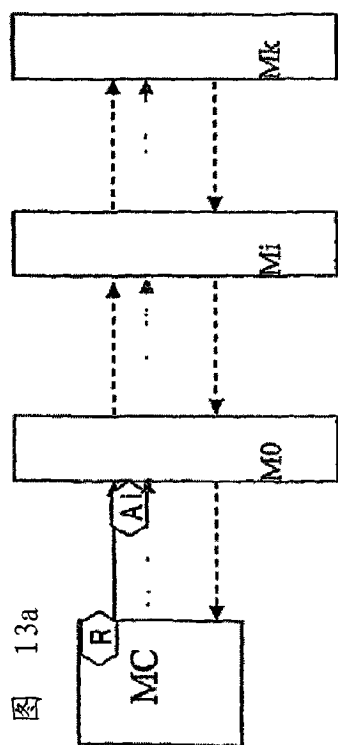
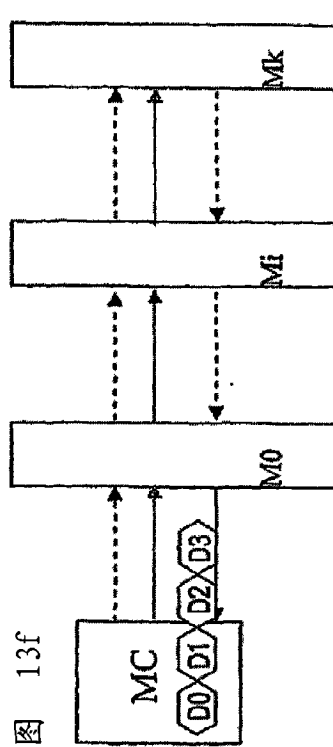
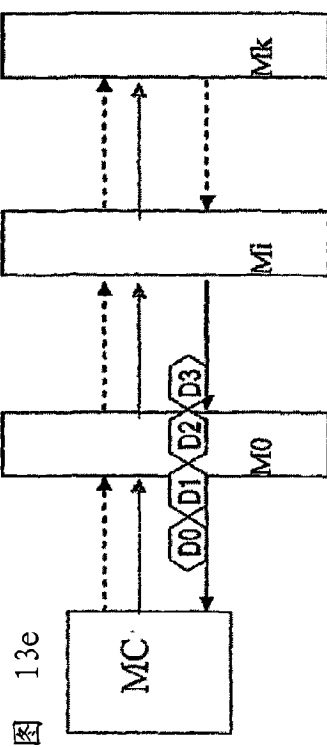
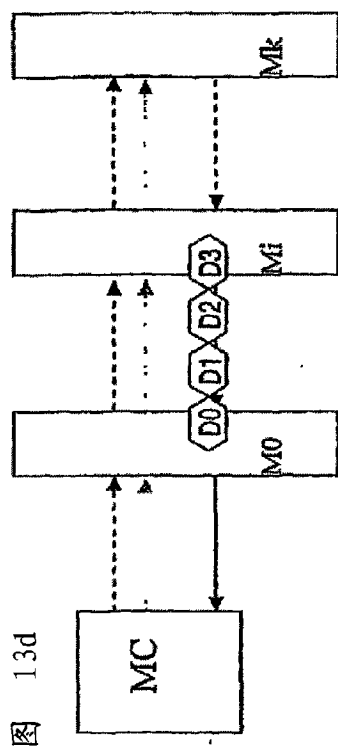


图 10







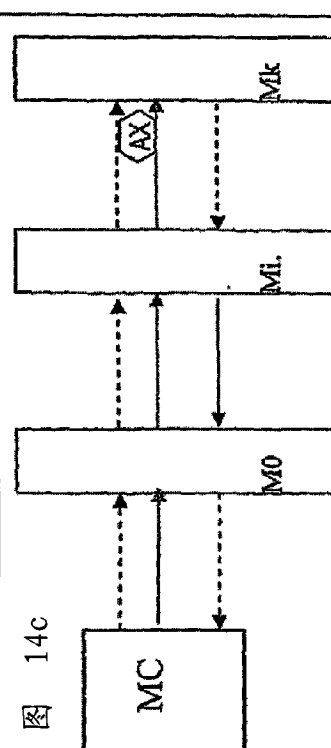
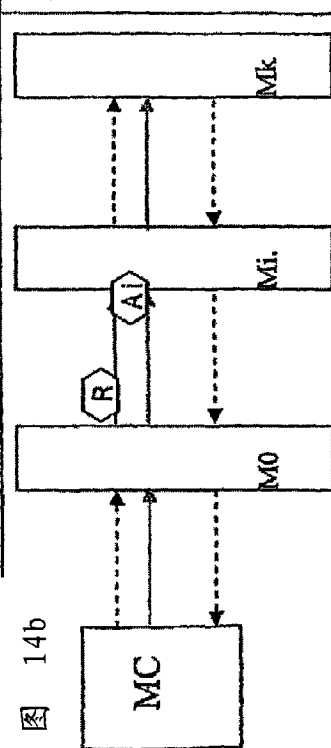
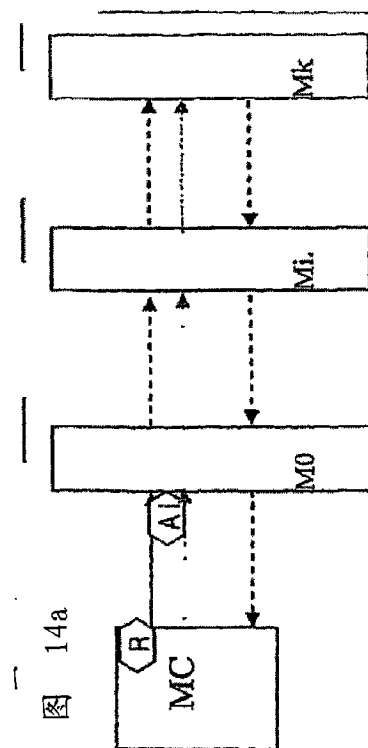
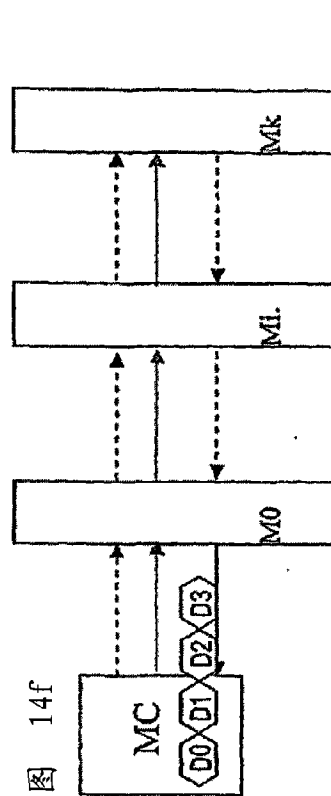
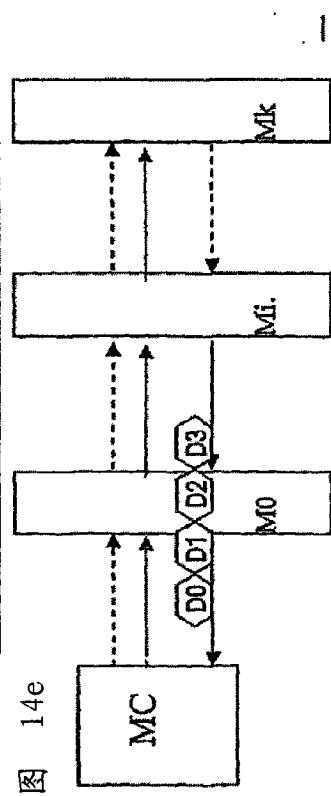
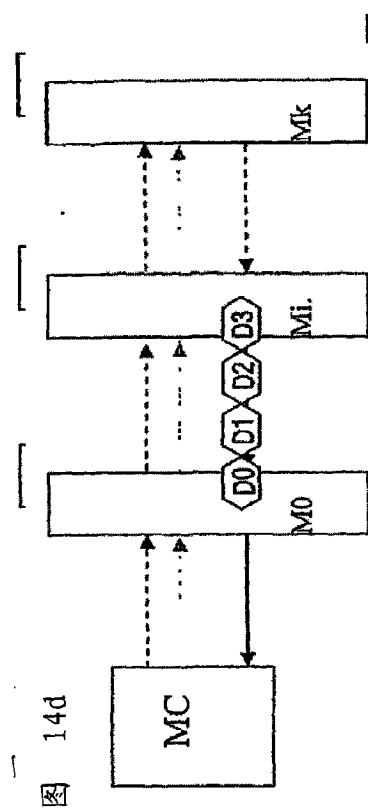


图 15d

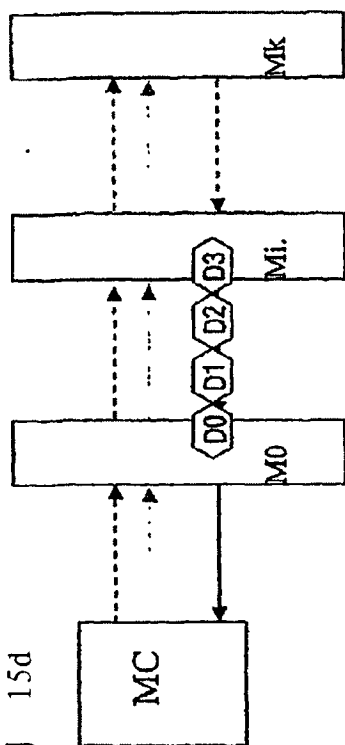


图 15e

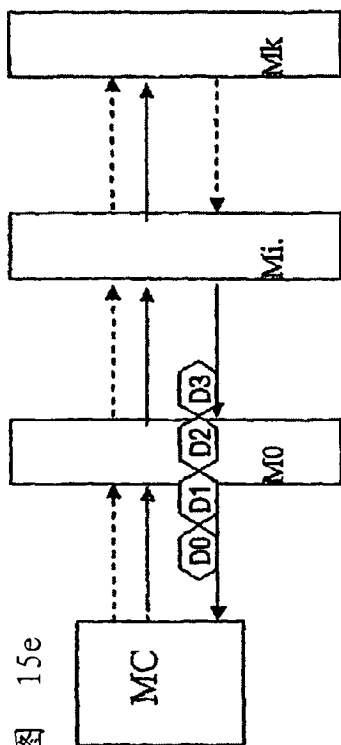


图 15f

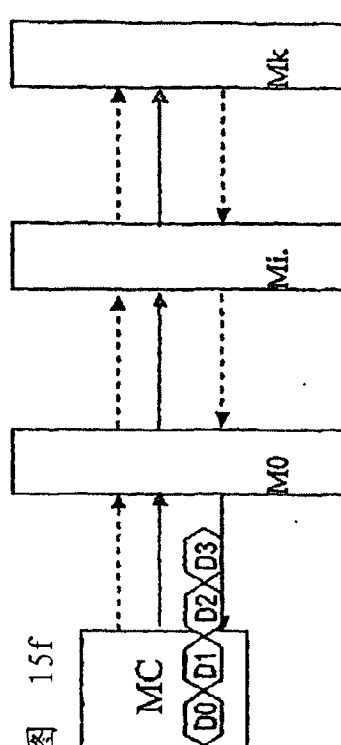


图 15a

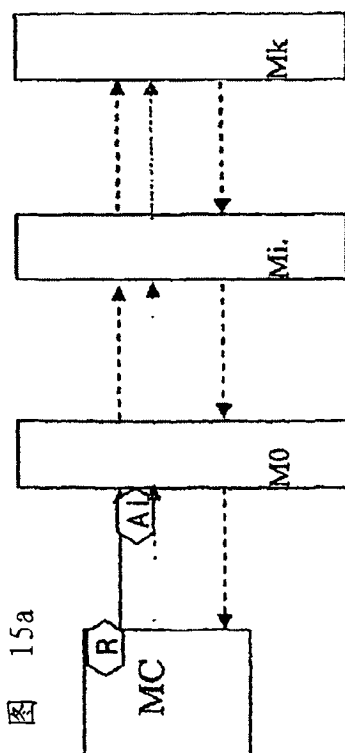


图 15b

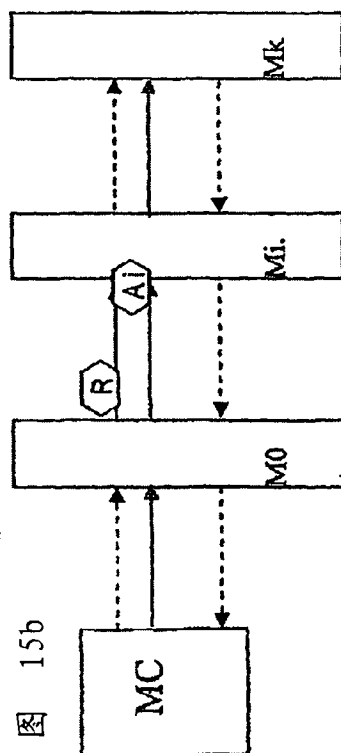
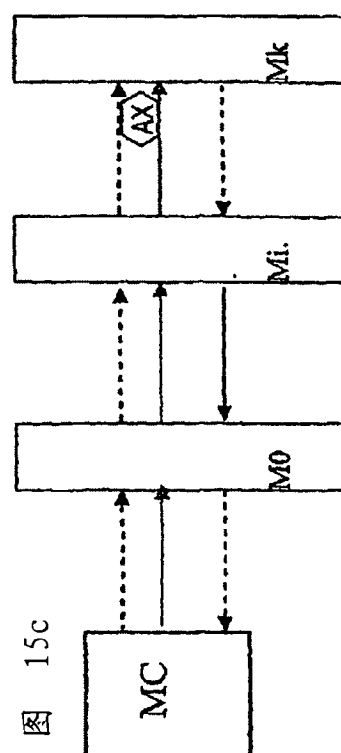
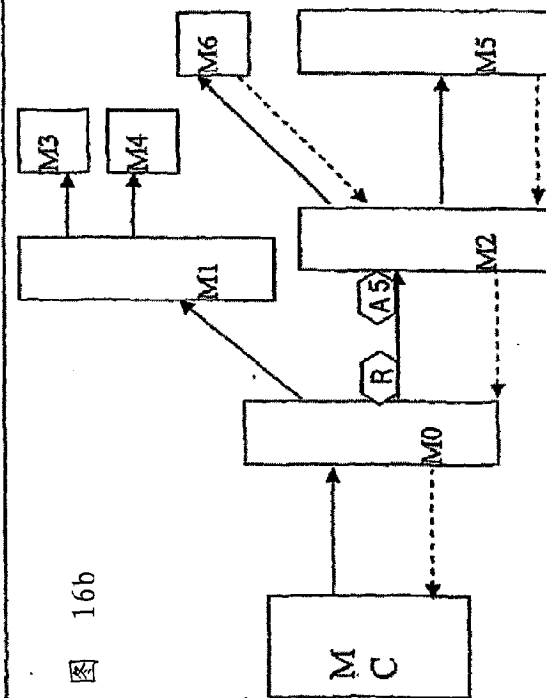
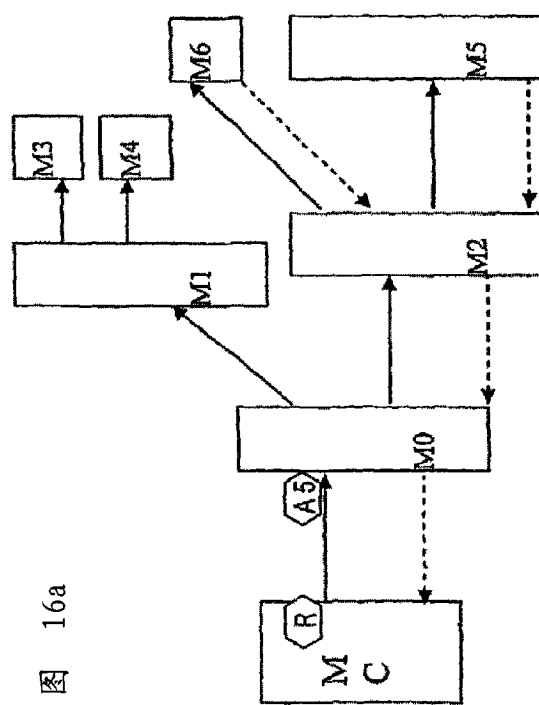
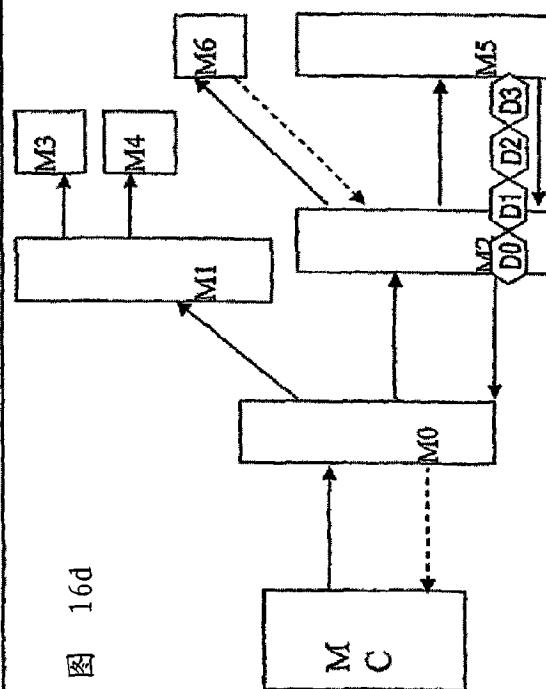
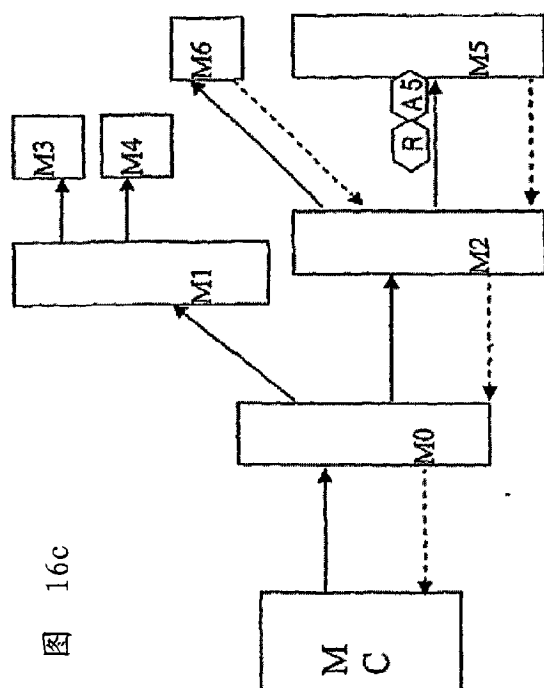


图 15c





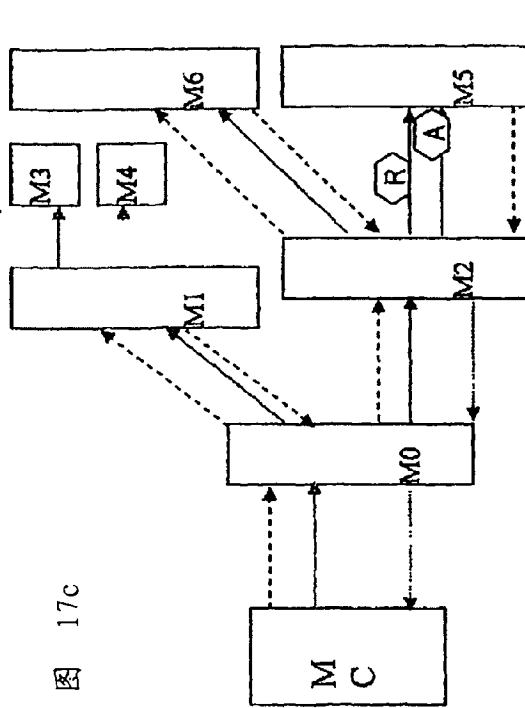


图 17c

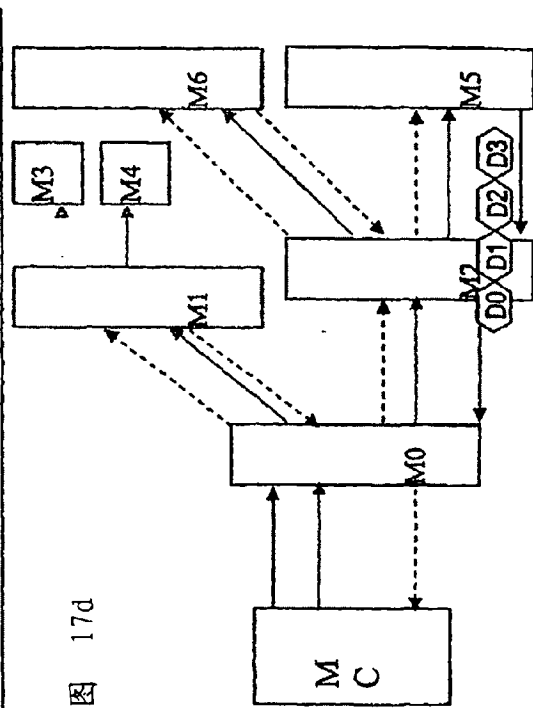


图 17d

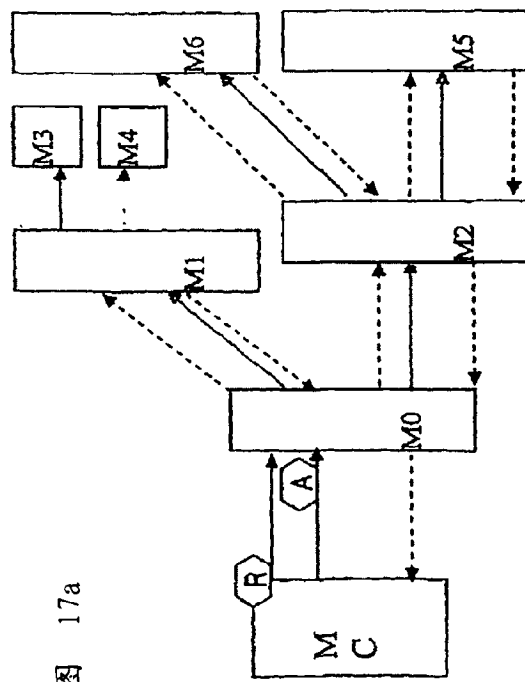


图 17a

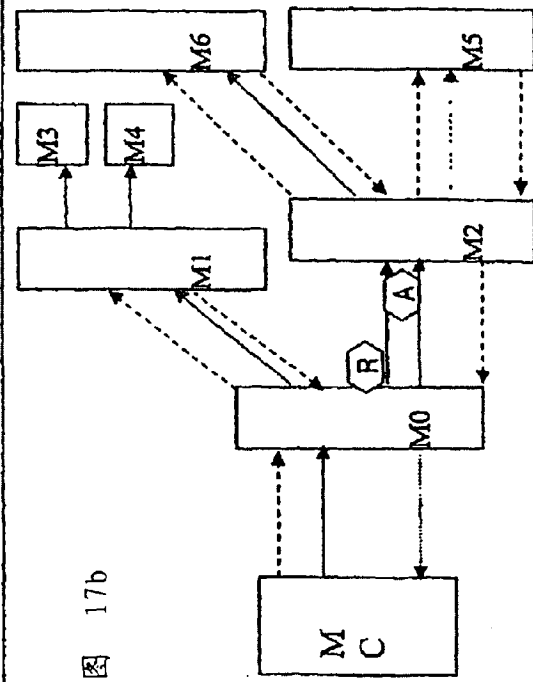


图 17b

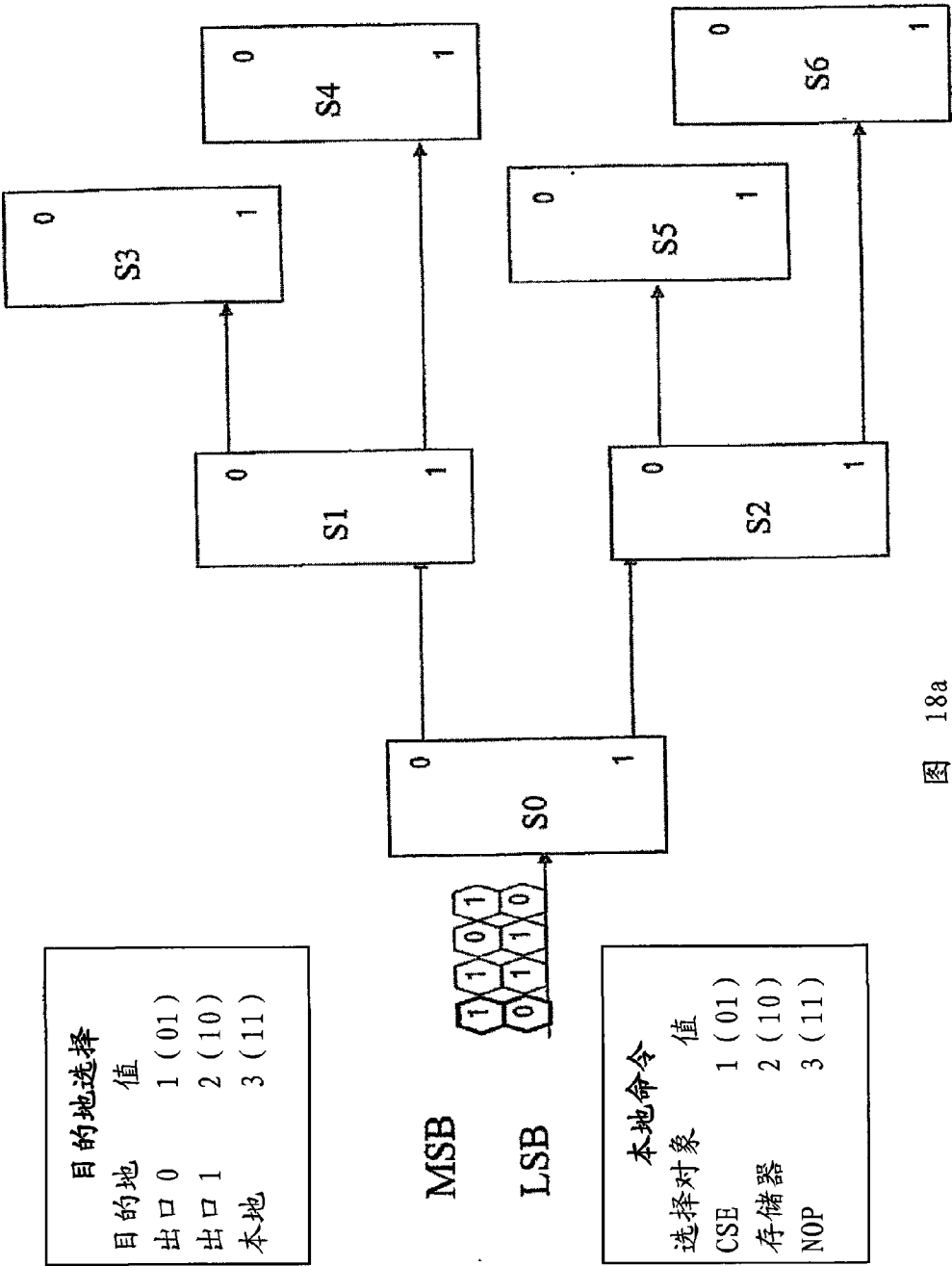
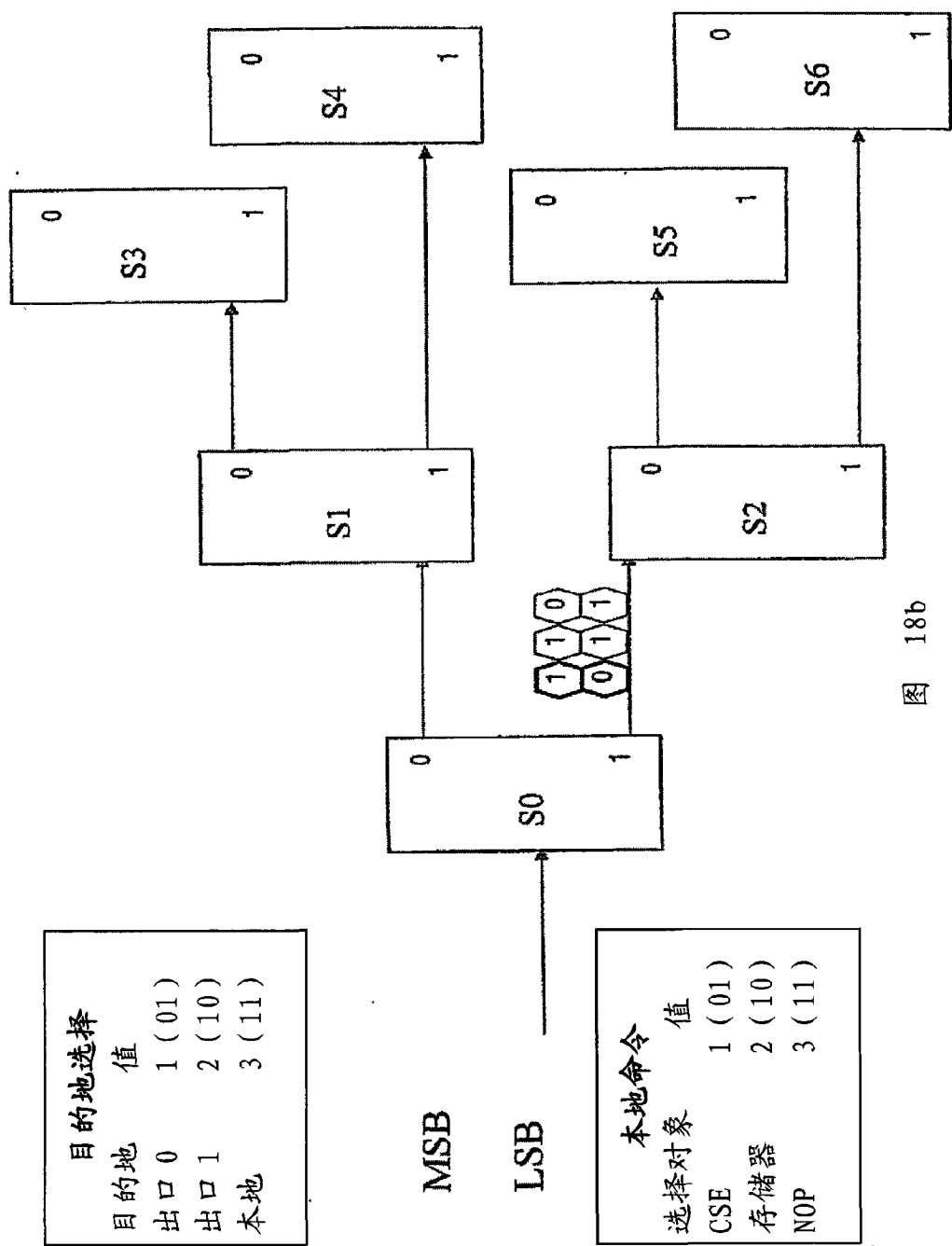


图 18a



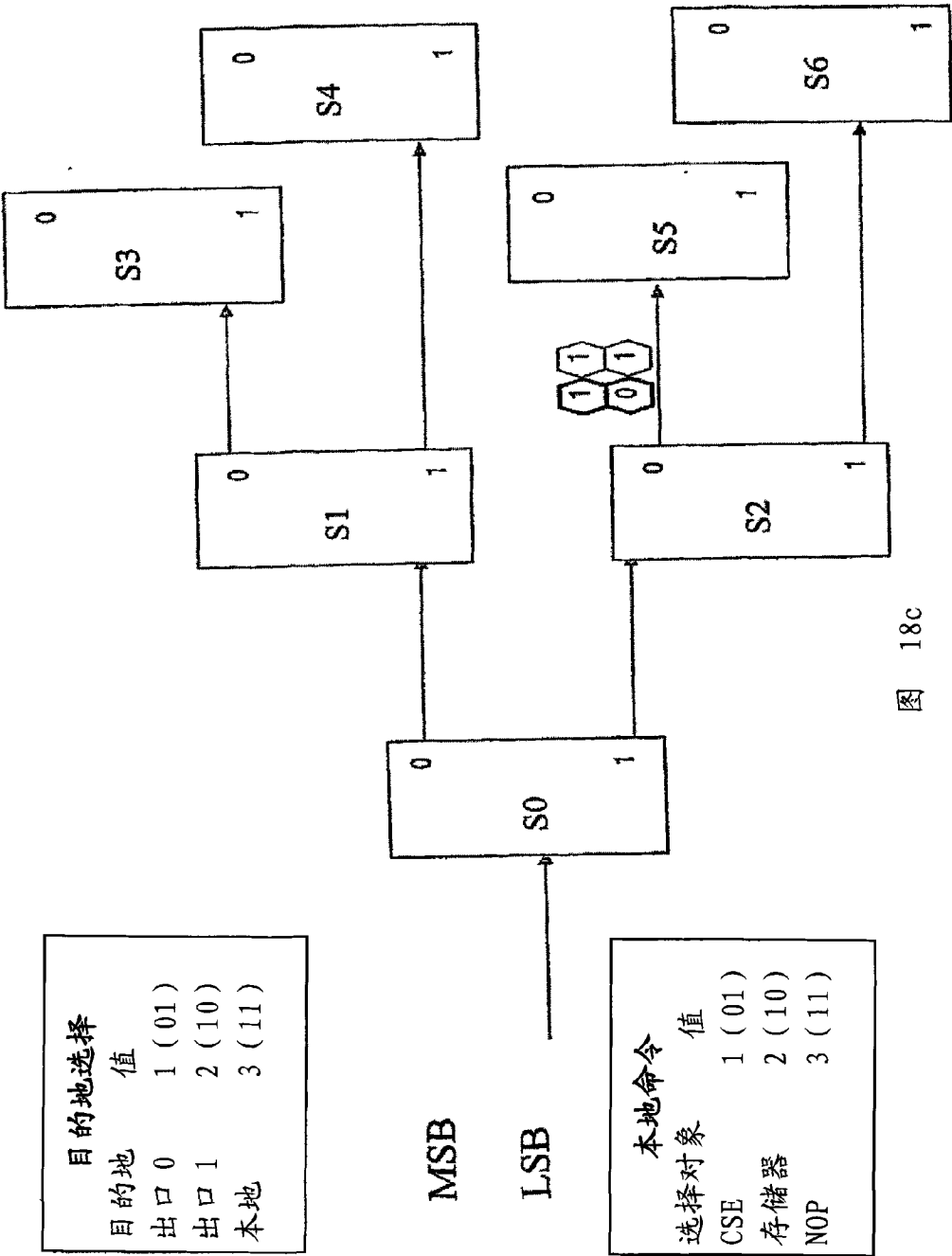
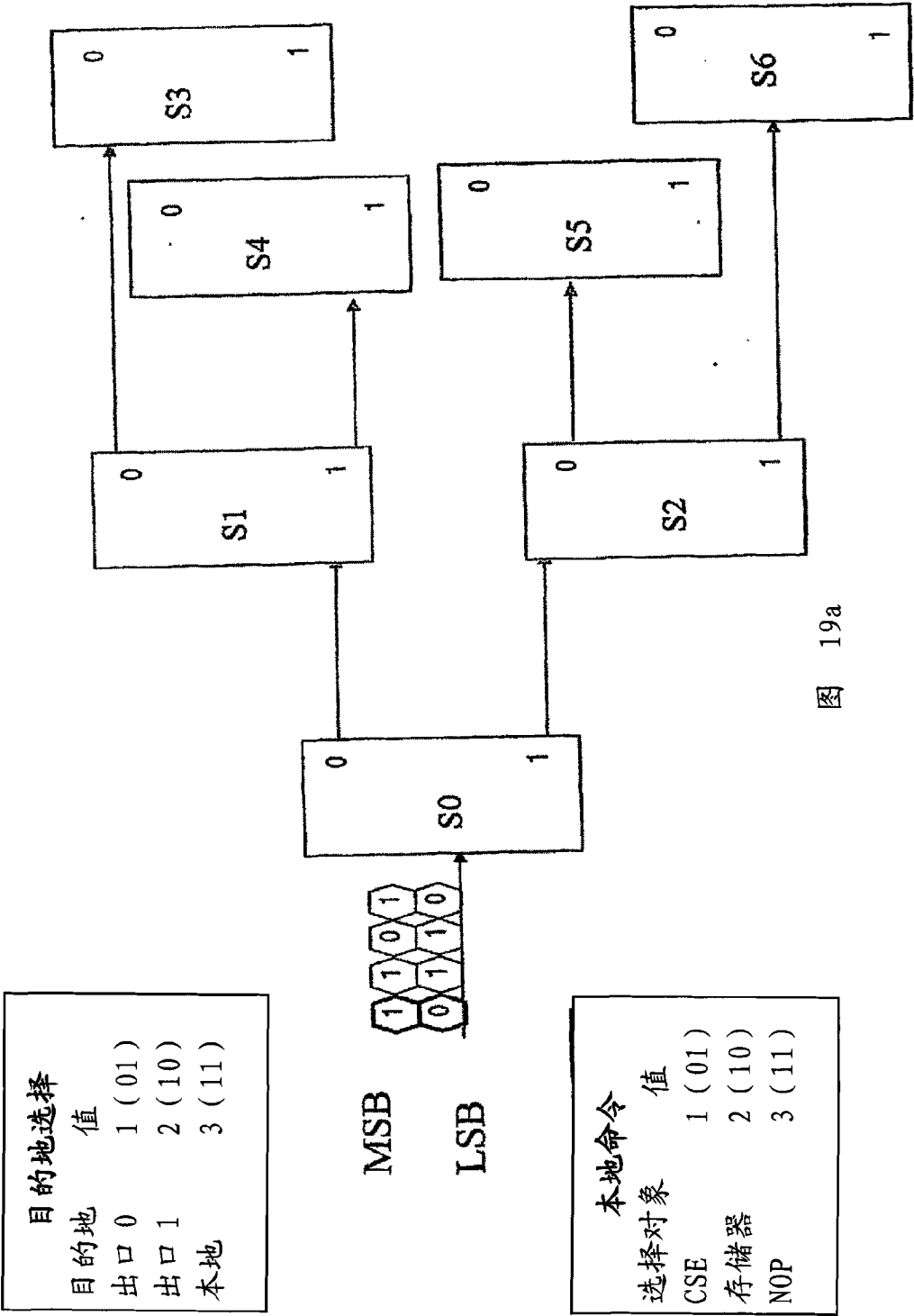


图 18c



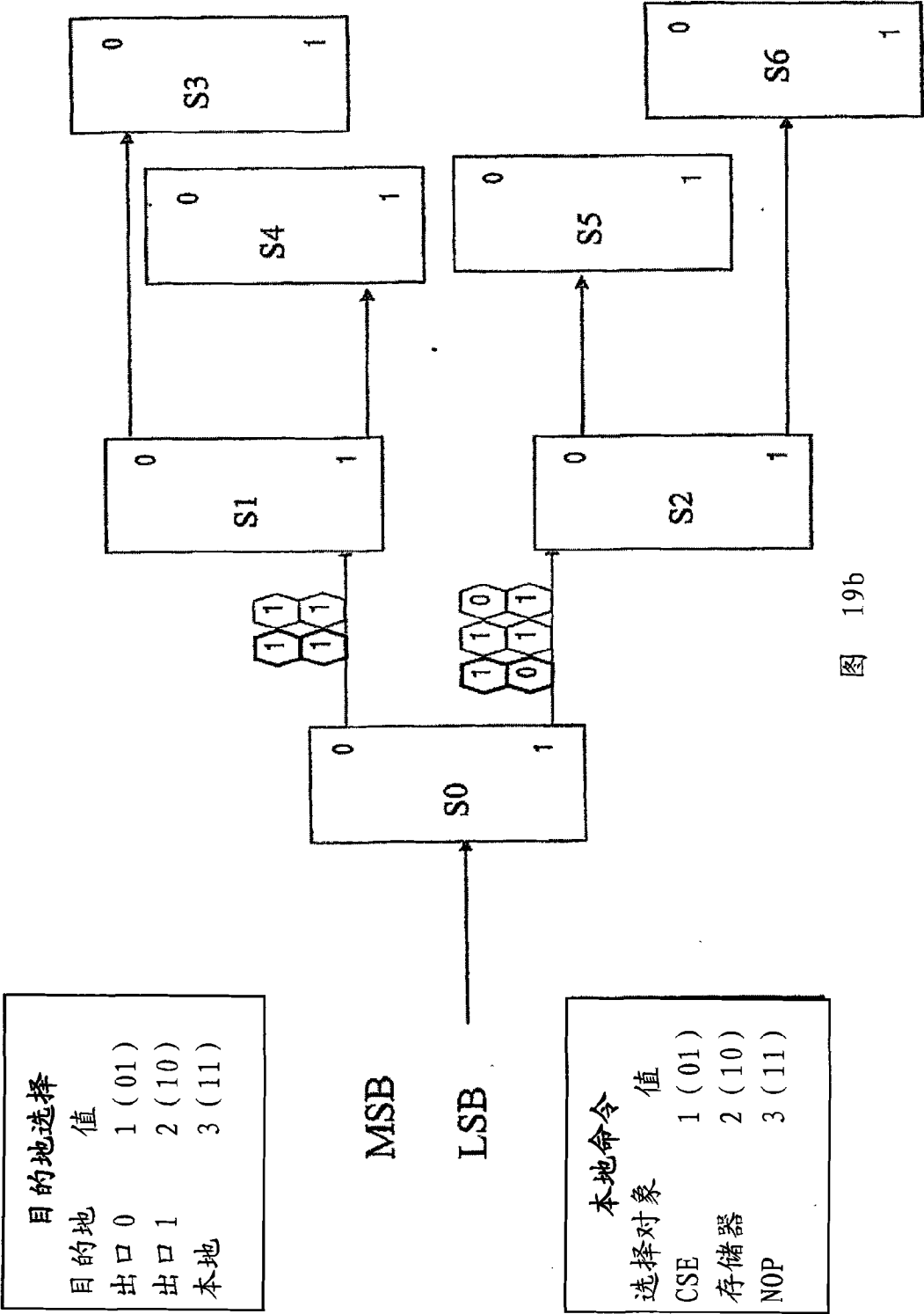
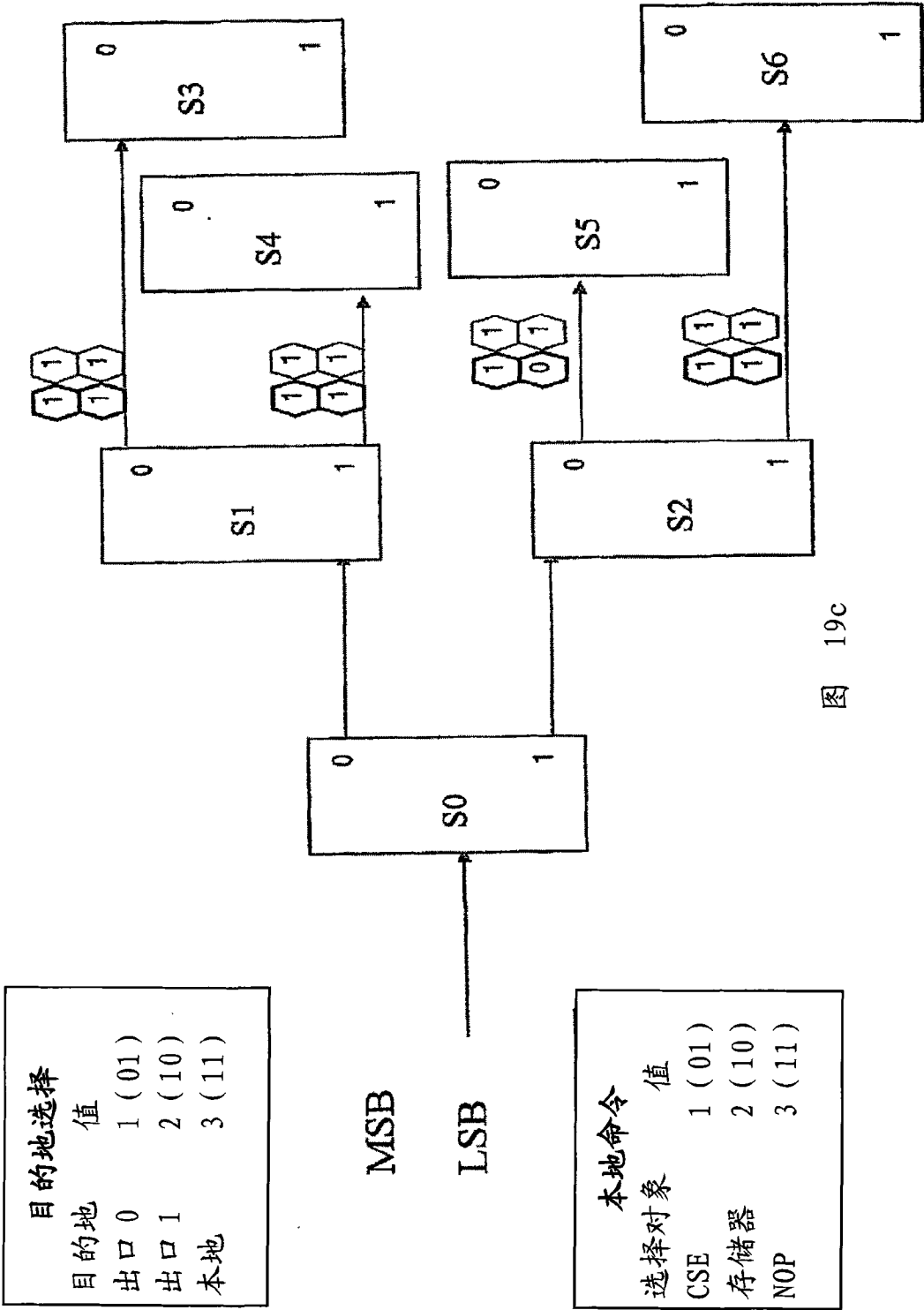


图 19b



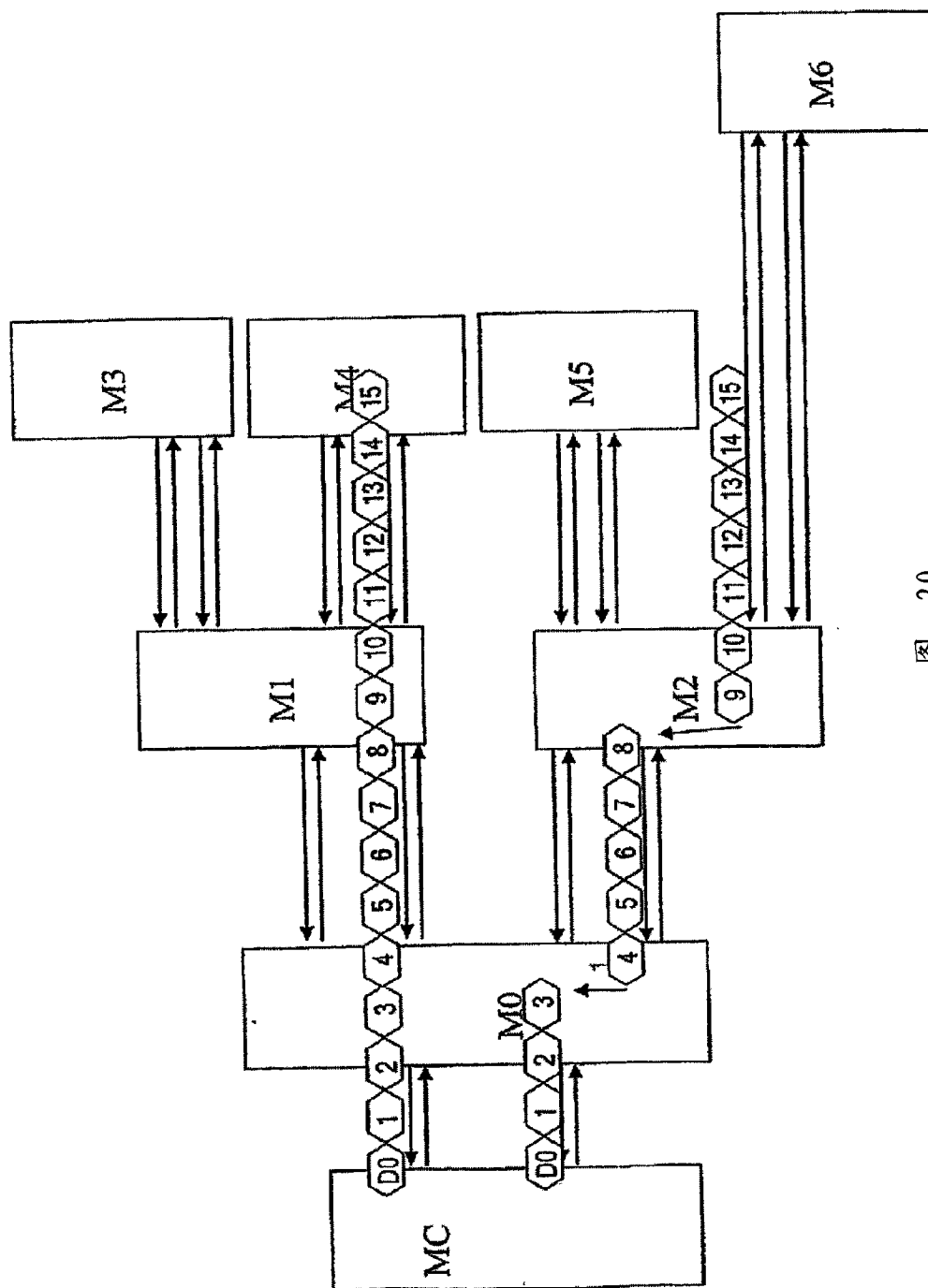


图 20

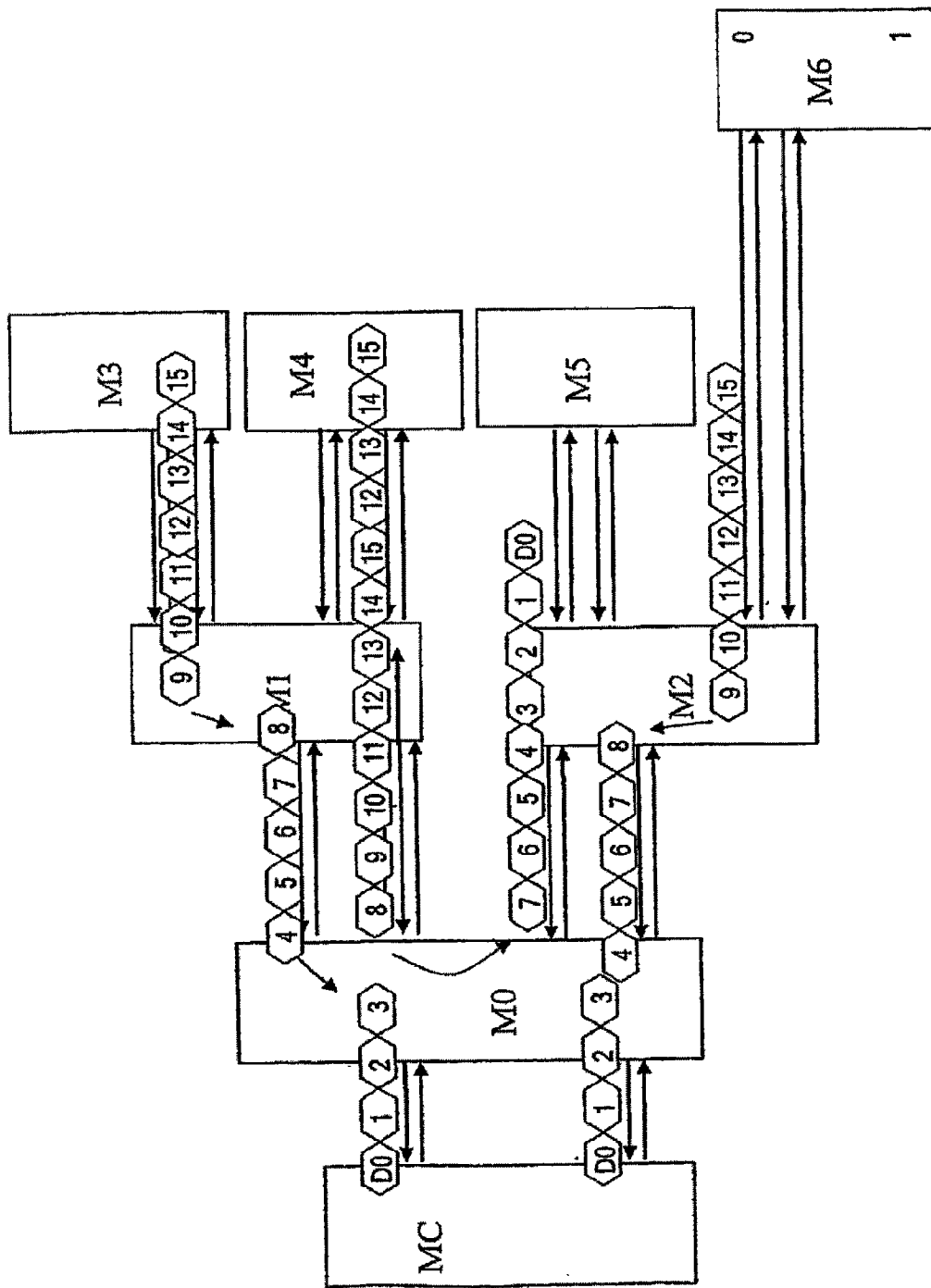


图 21

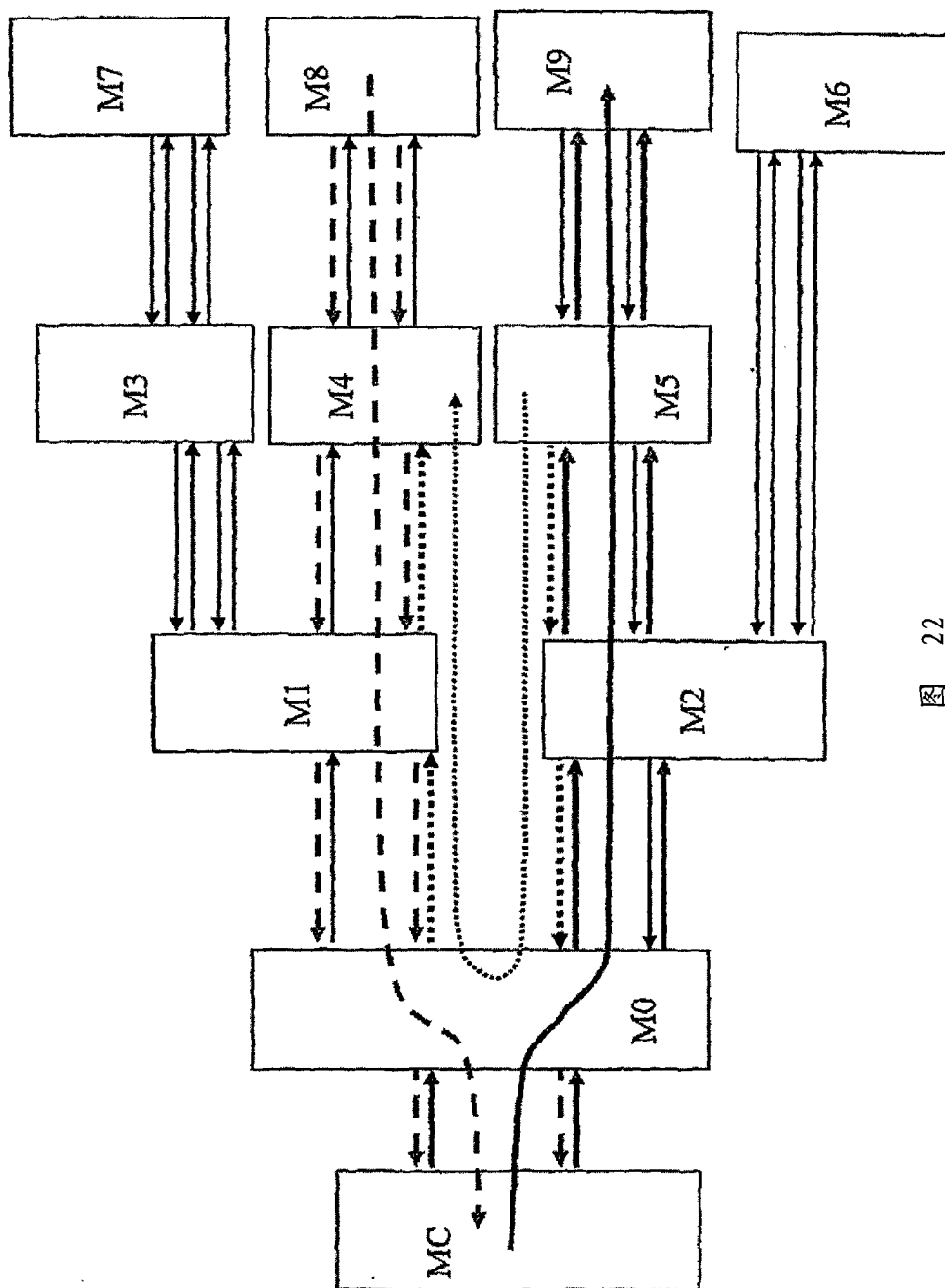
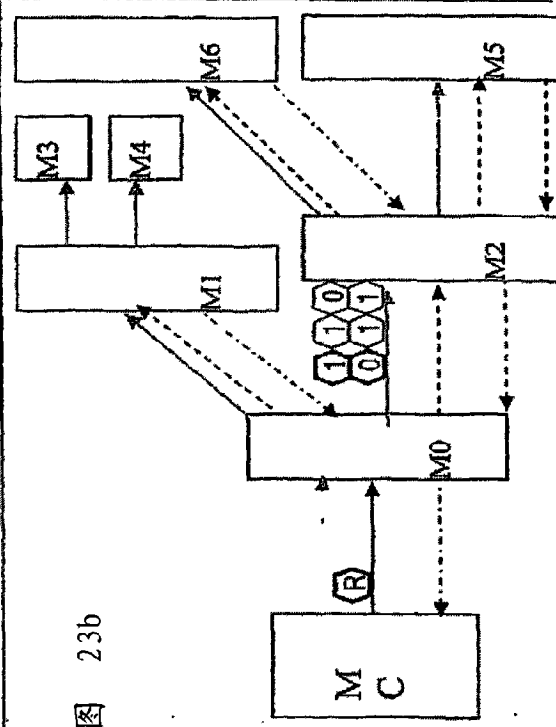
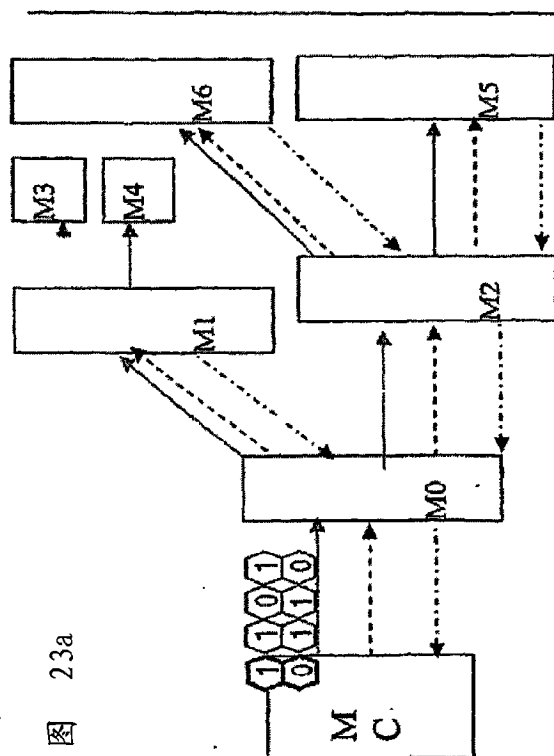
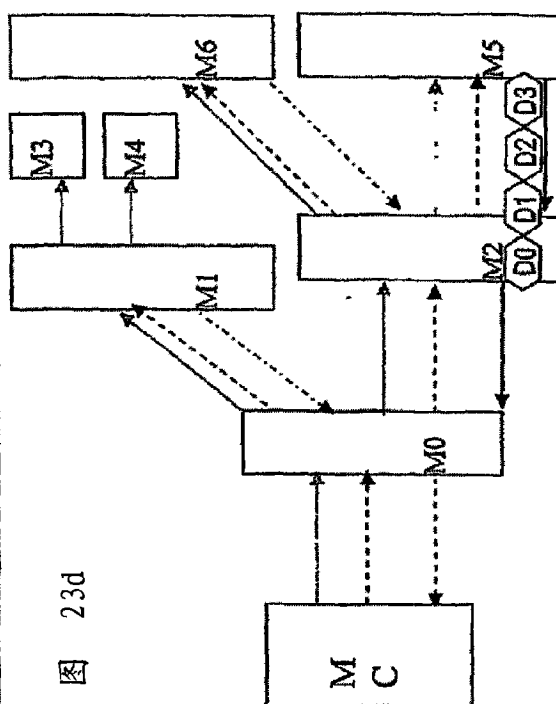
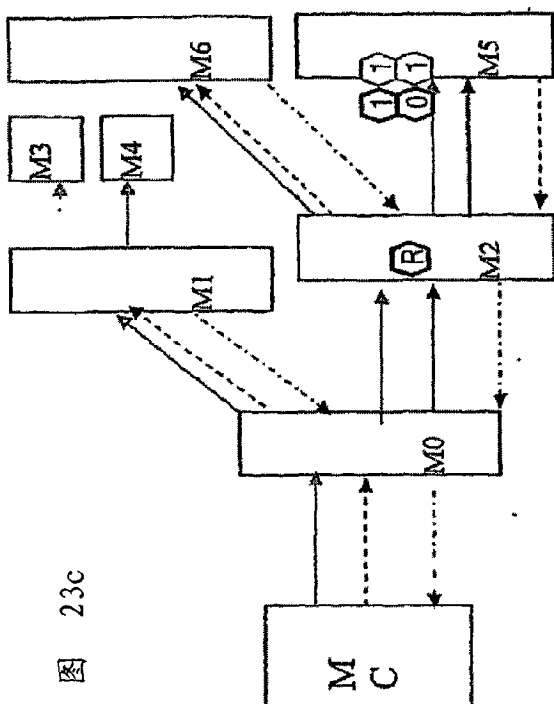
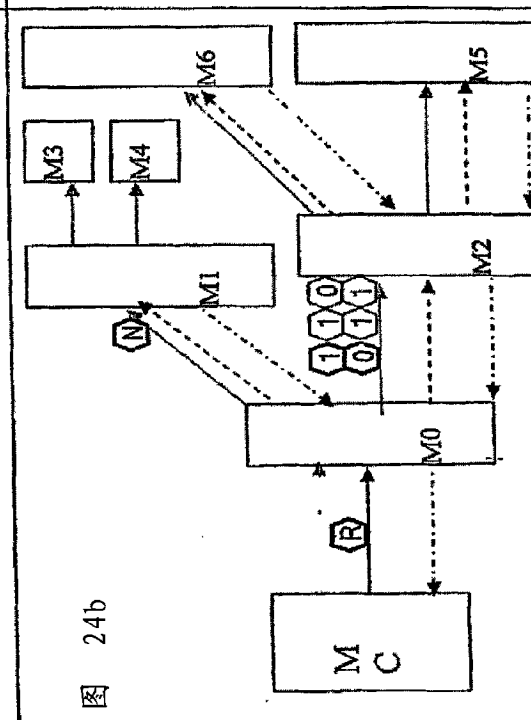
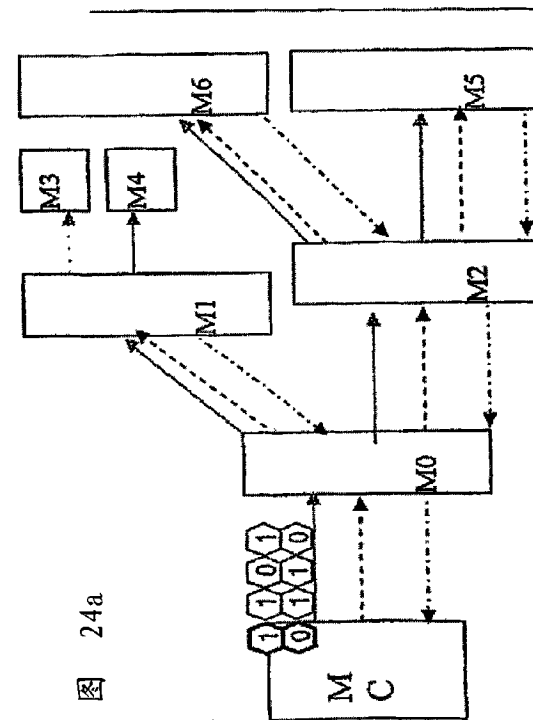
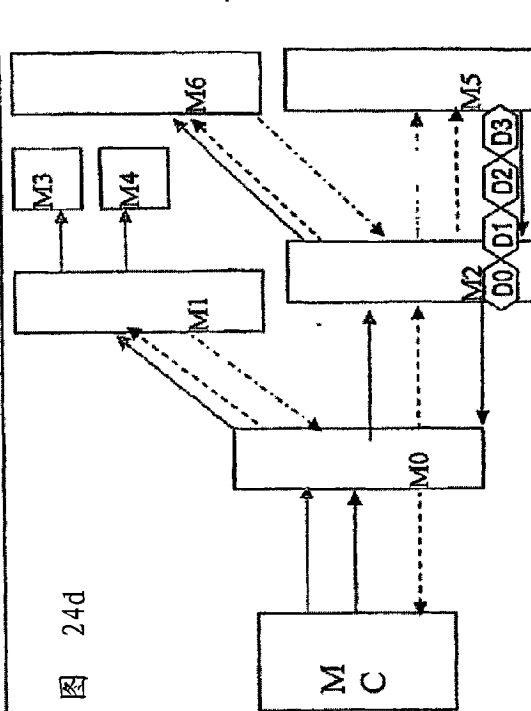
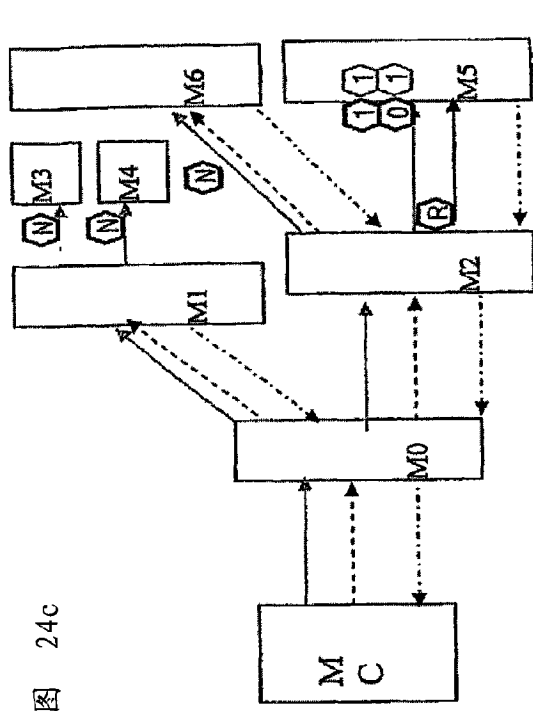
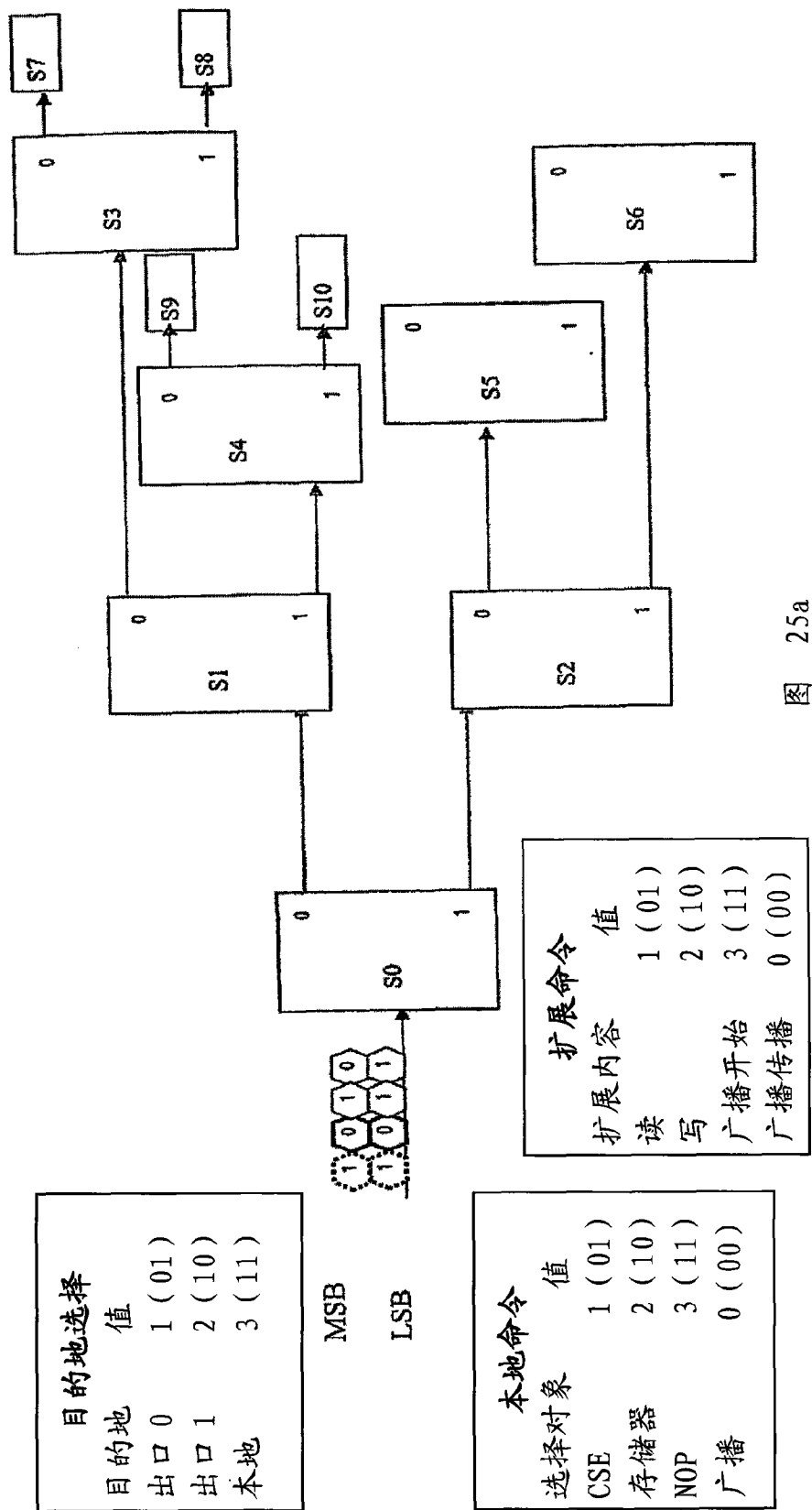
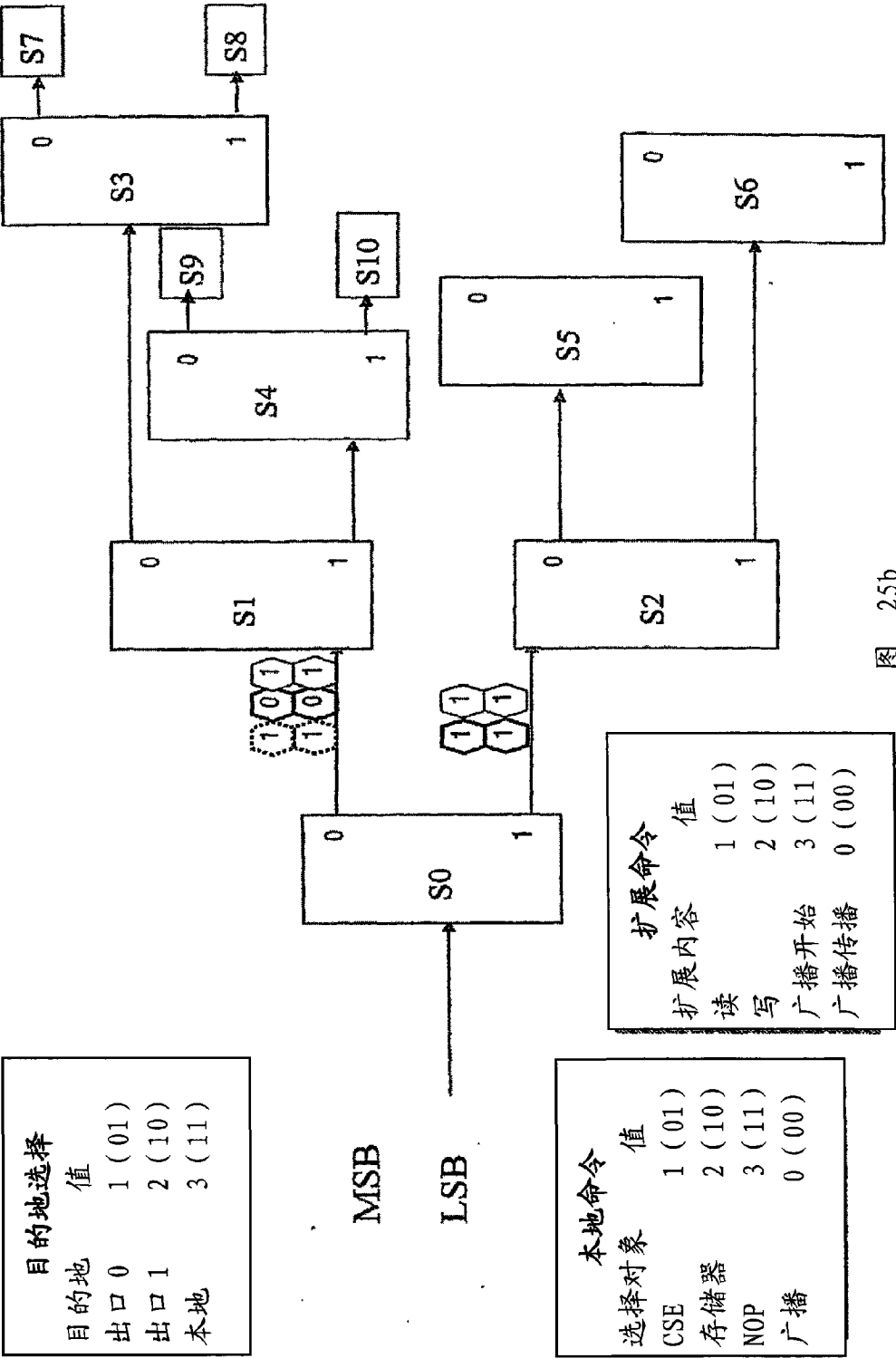


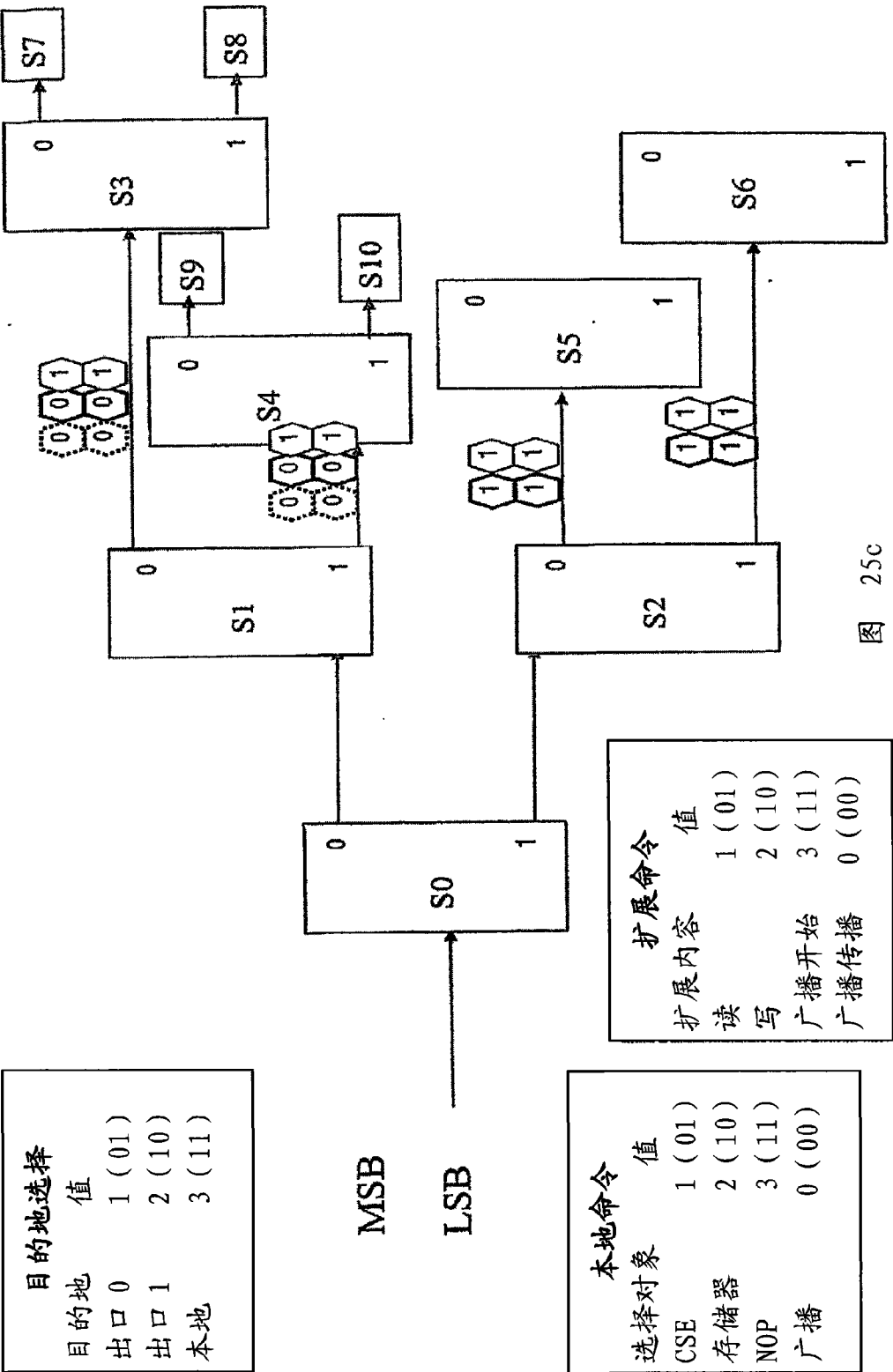
图 22











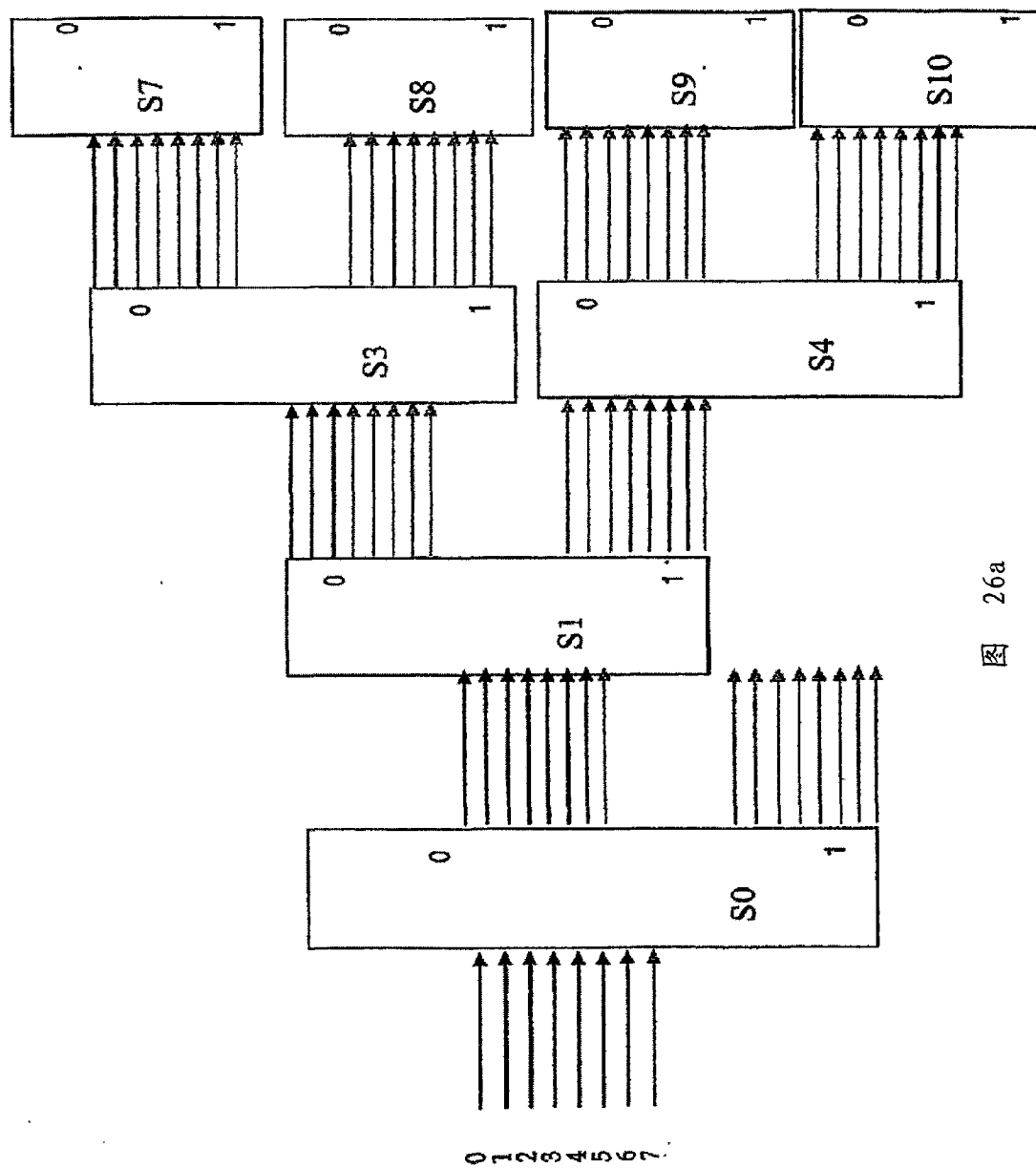


图 26a

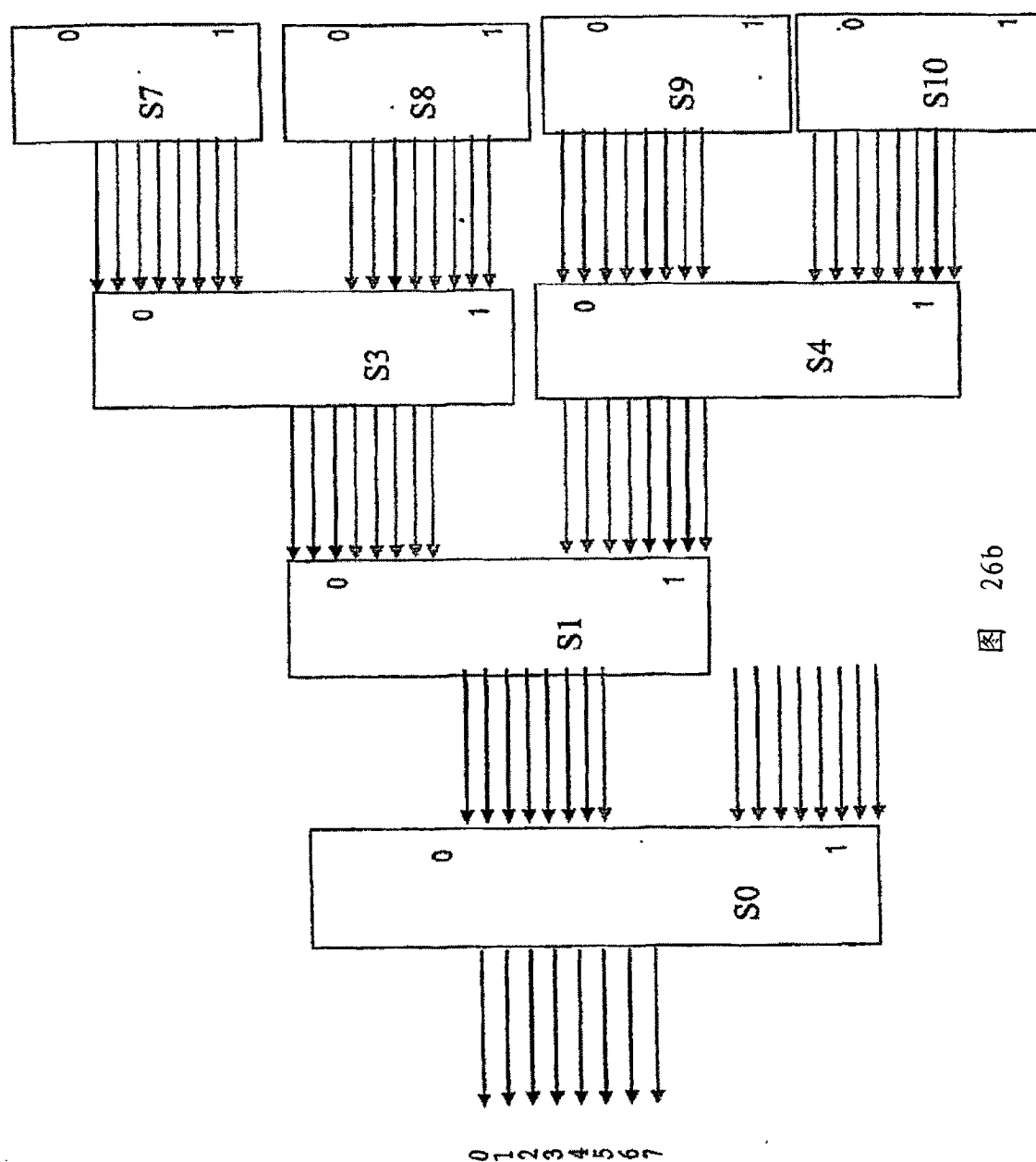


图 26b

图 27a

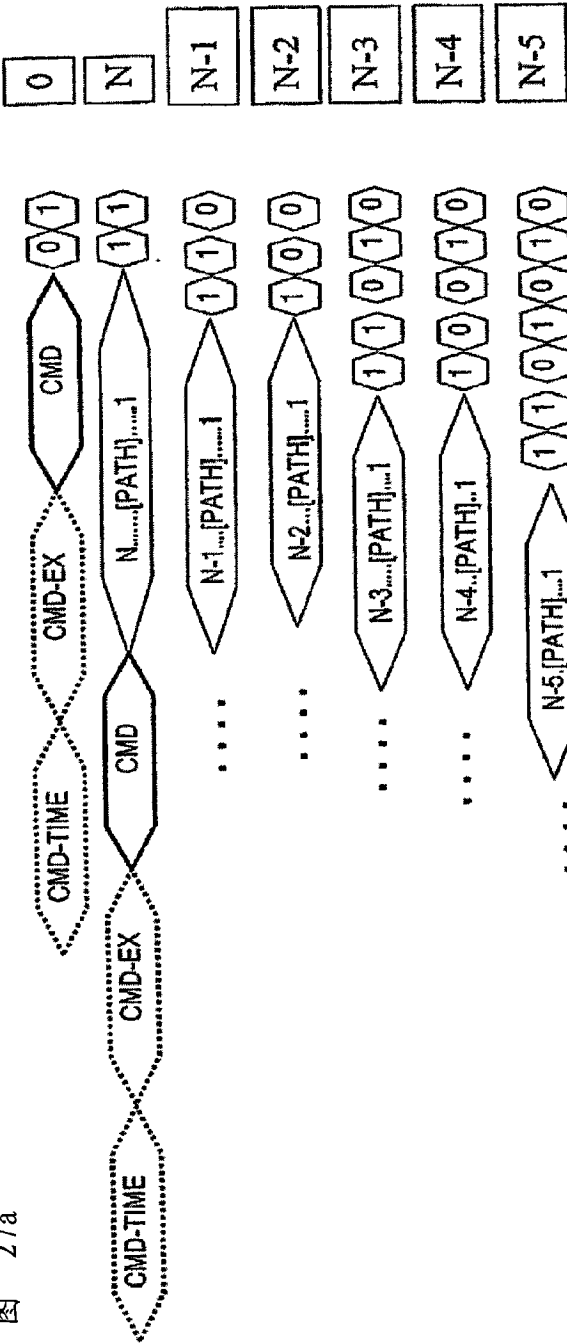


图 27b

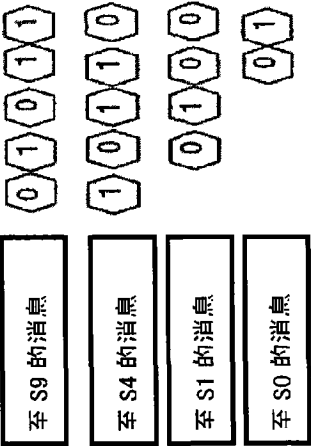


图 28a

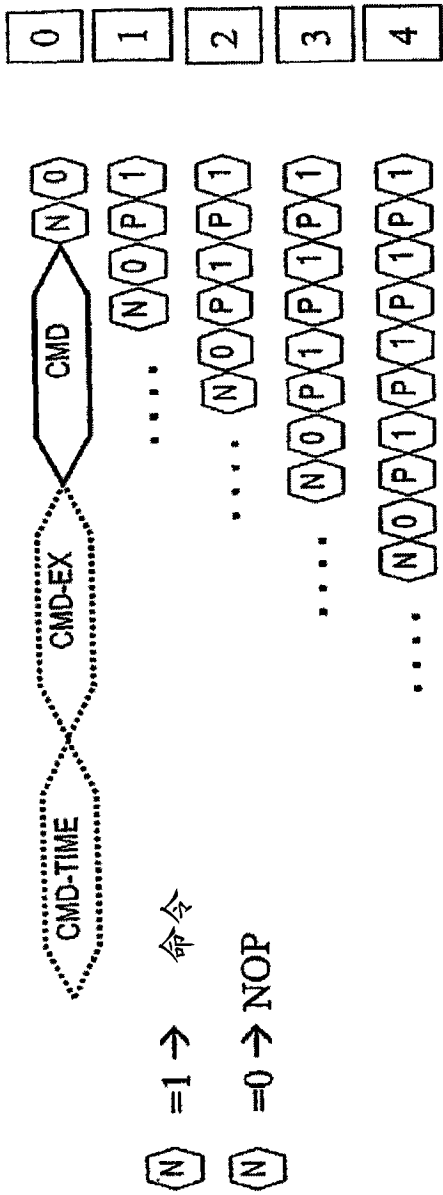
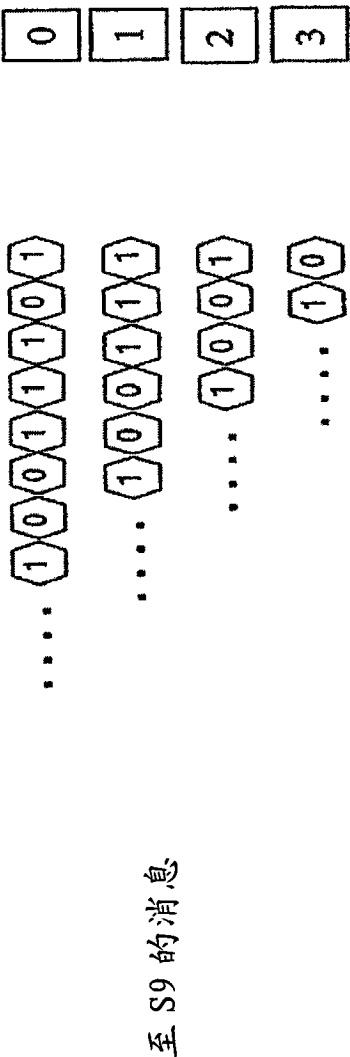


图 28b



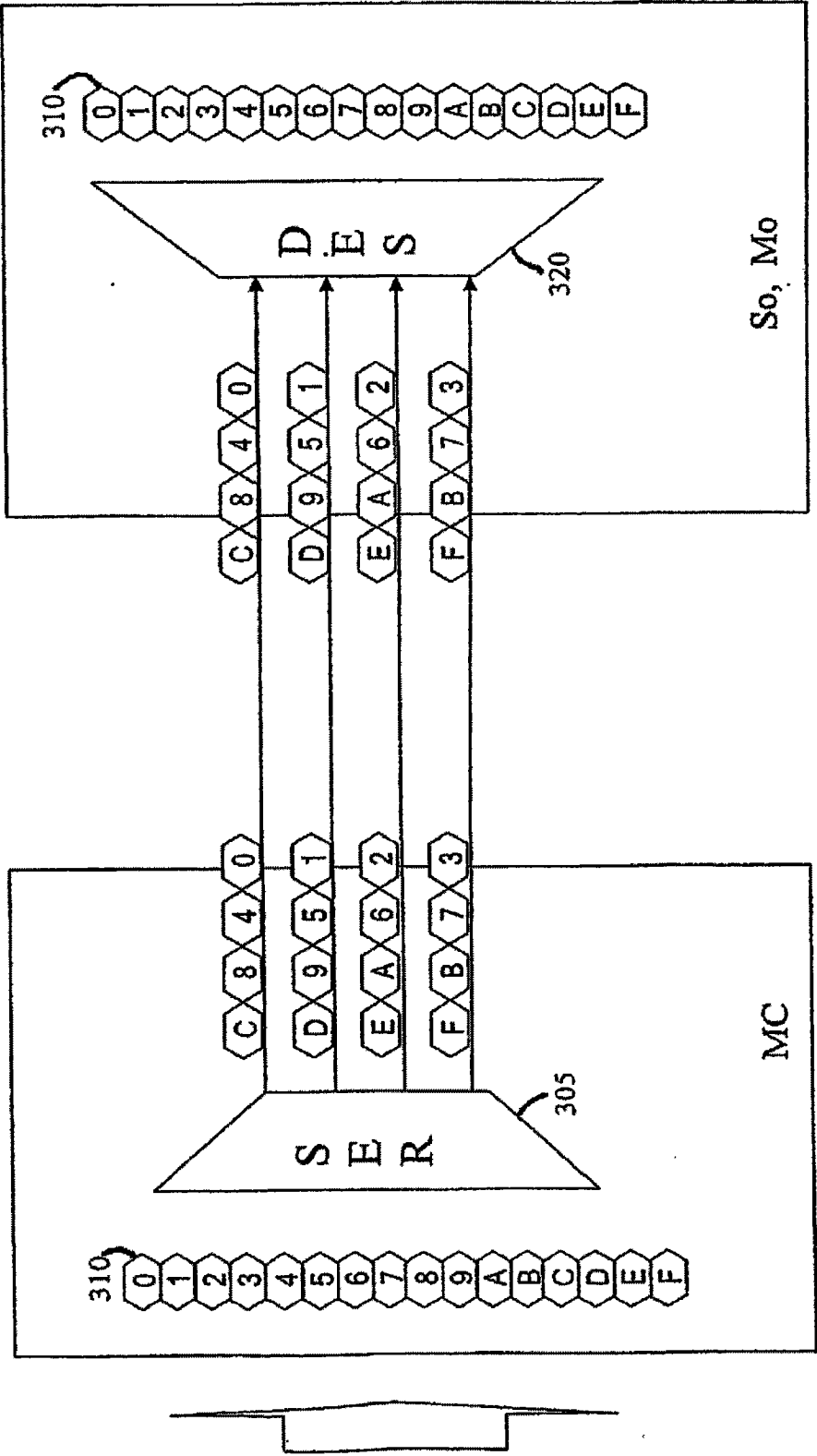


图 29a

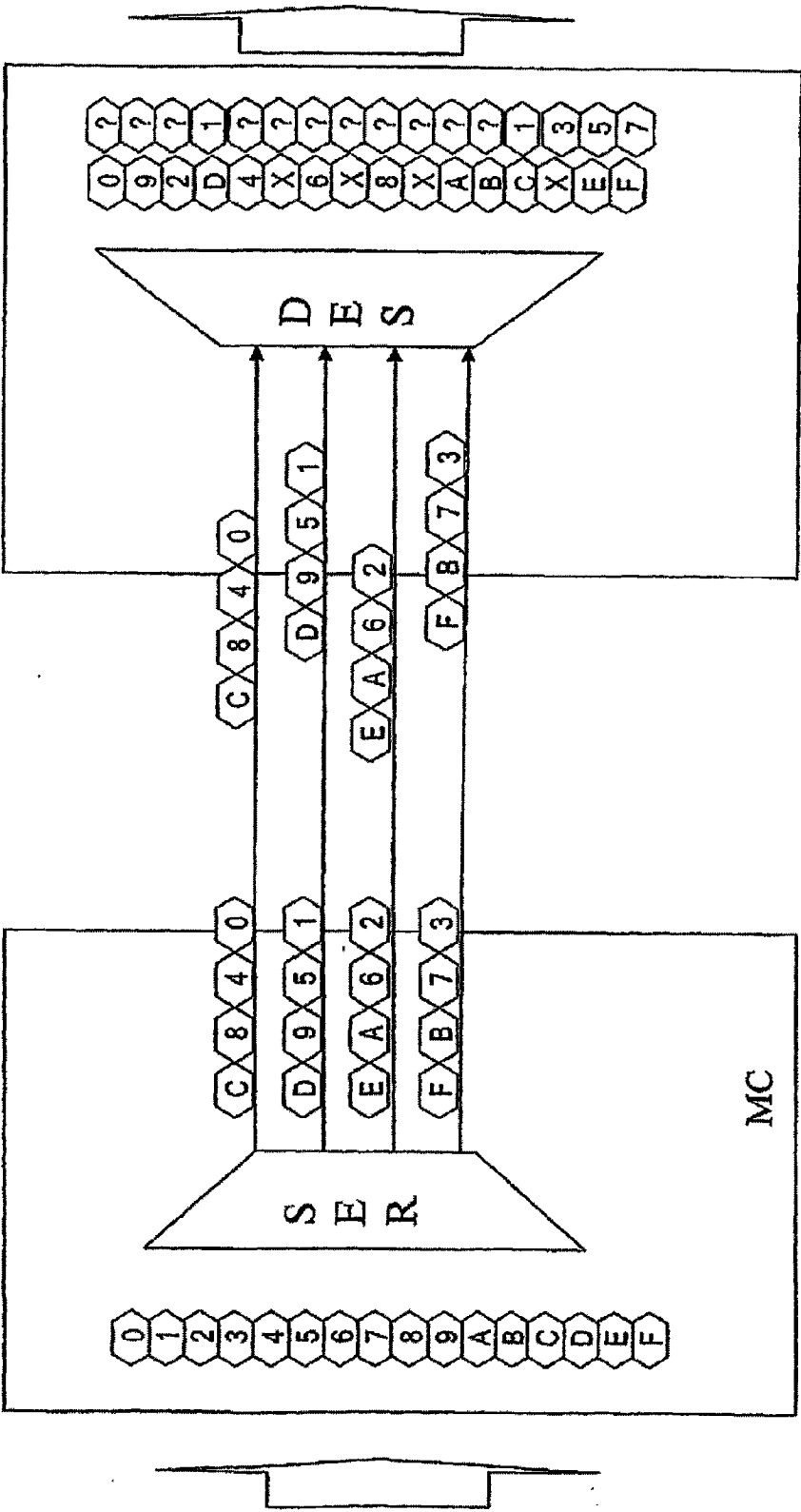


图 29b

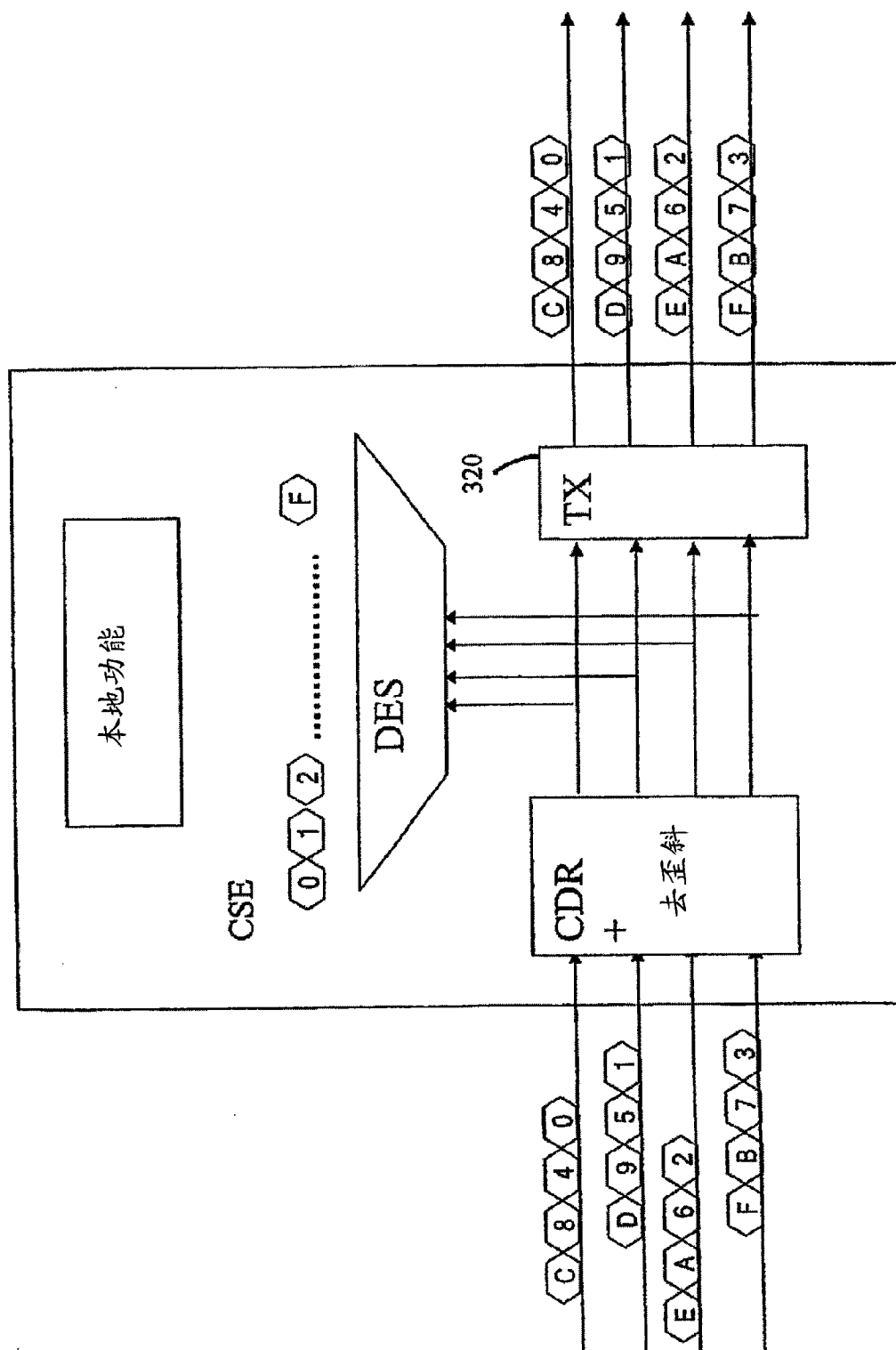


图 30

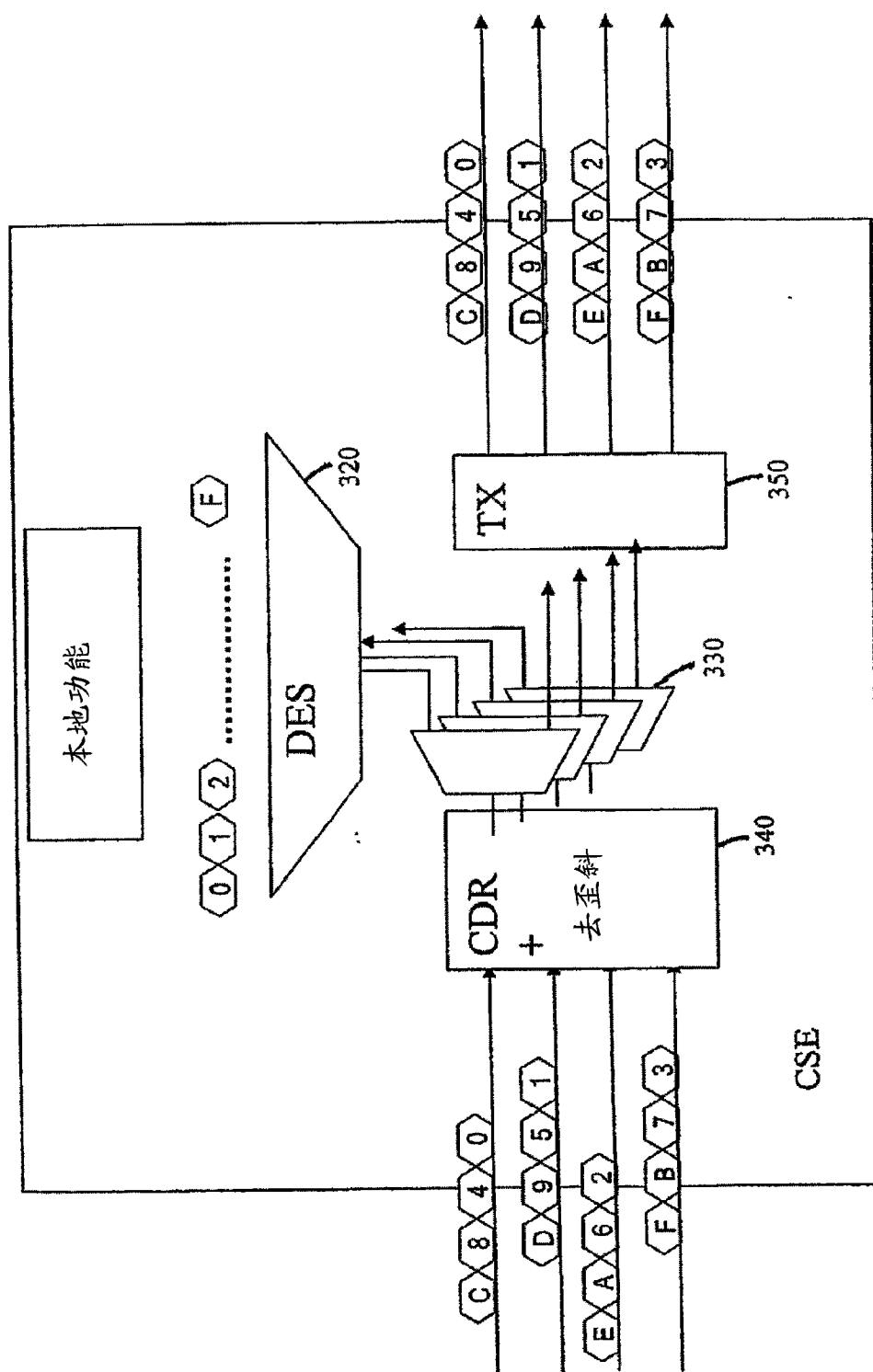


图 31

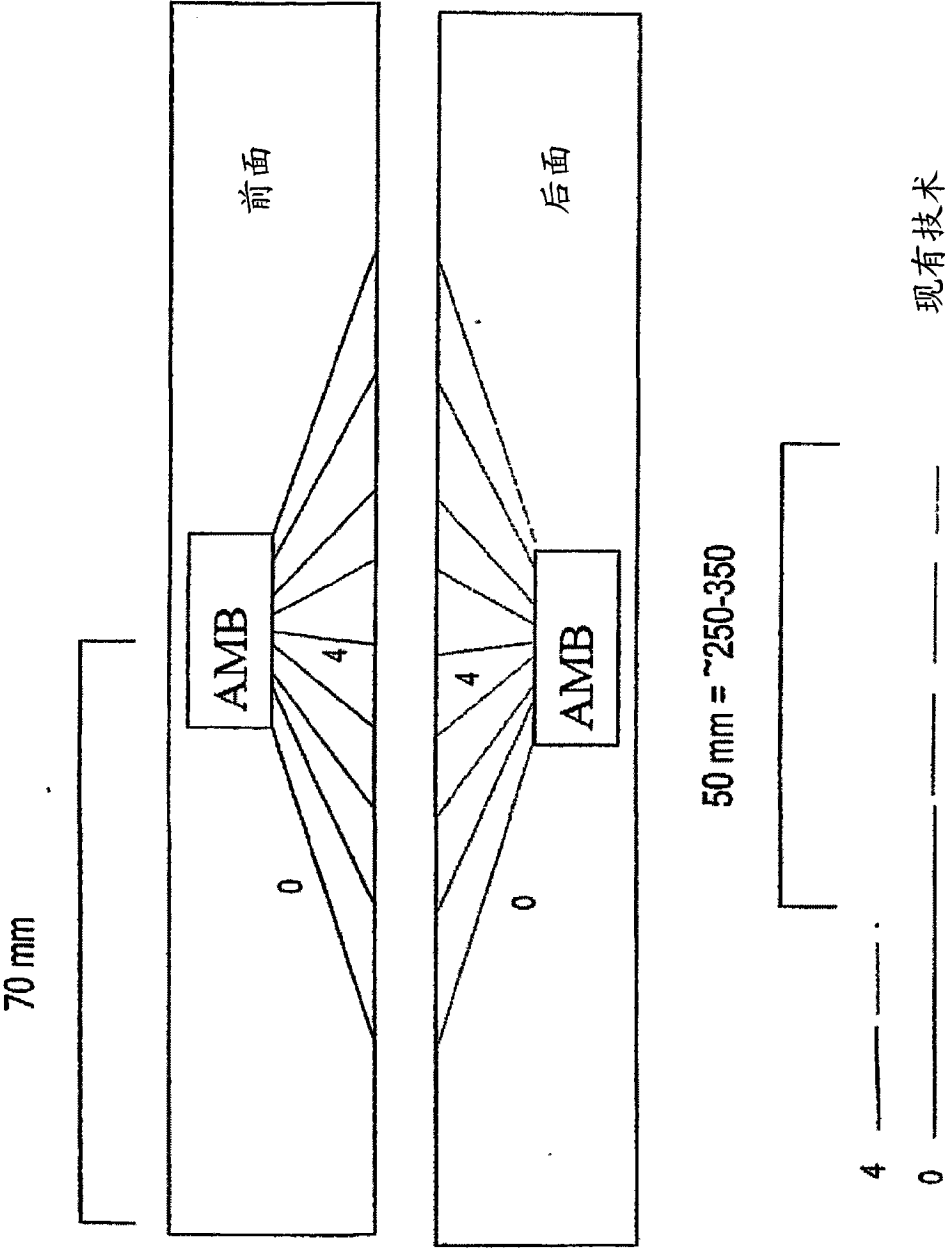


图 32

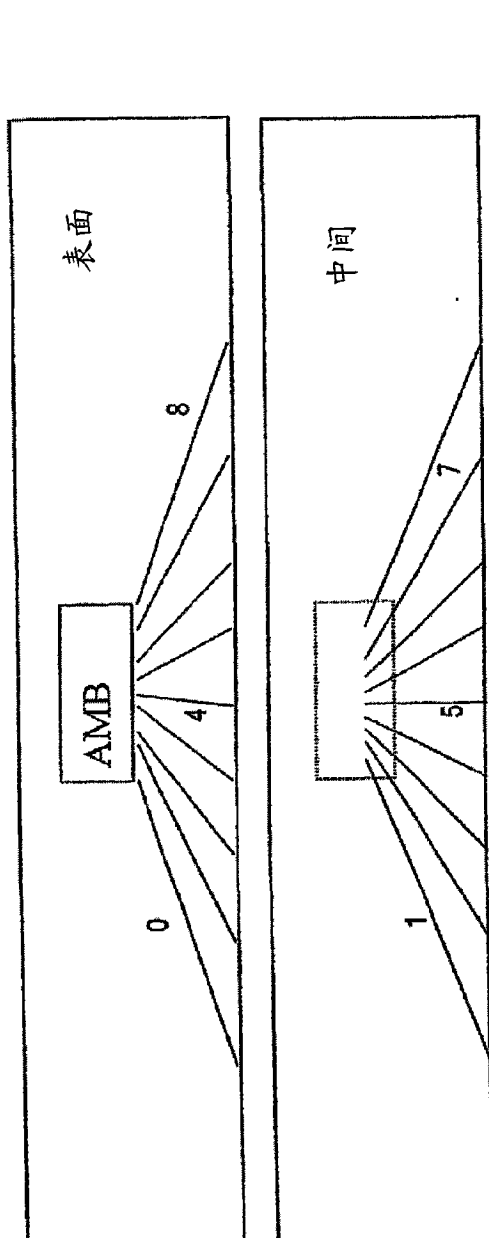


图 33a

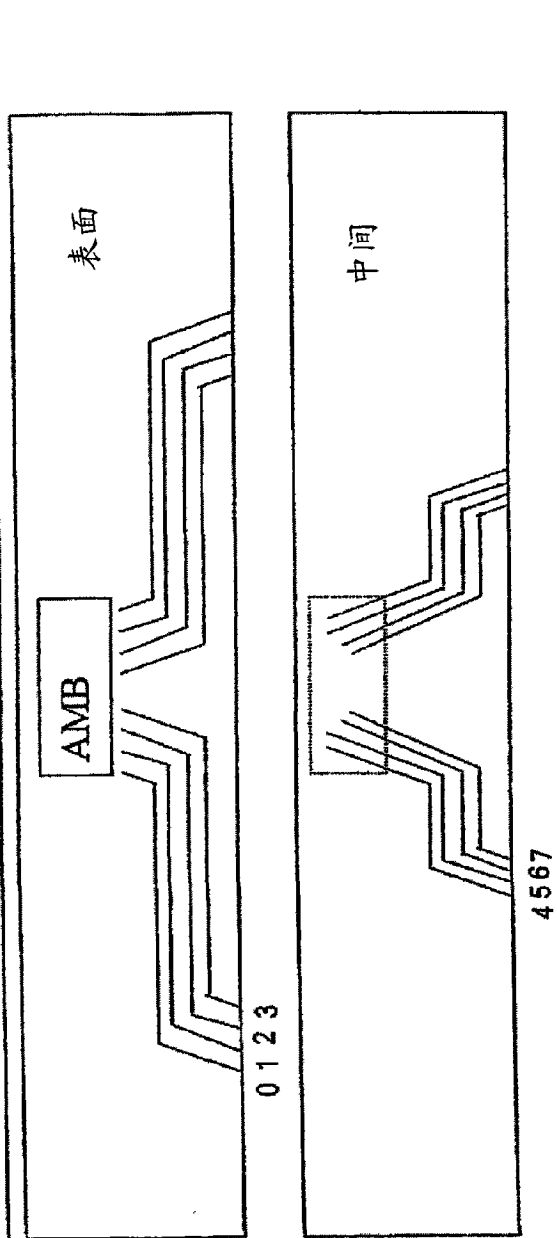


图 33b

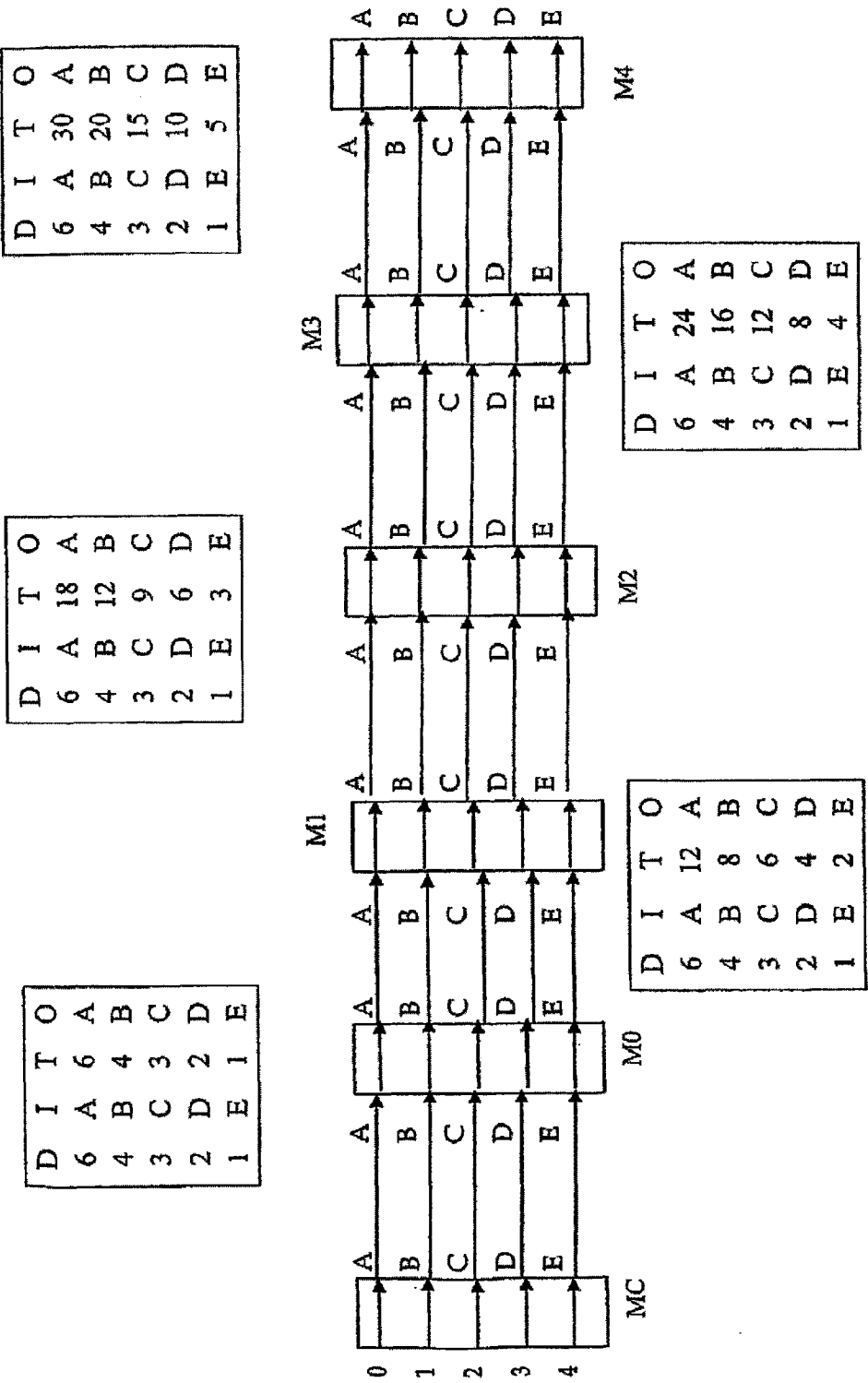


图 34

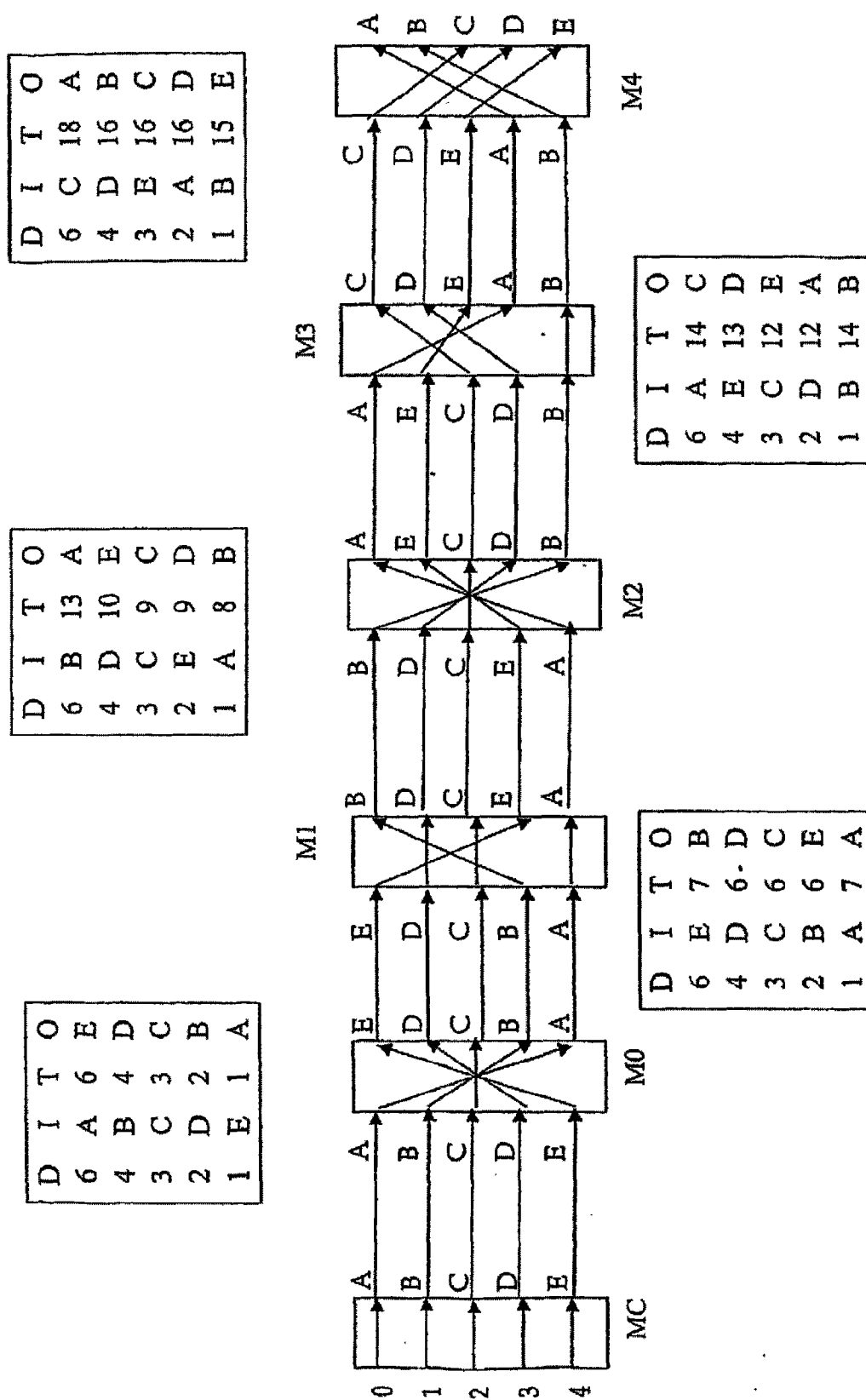


图 35

图 36a

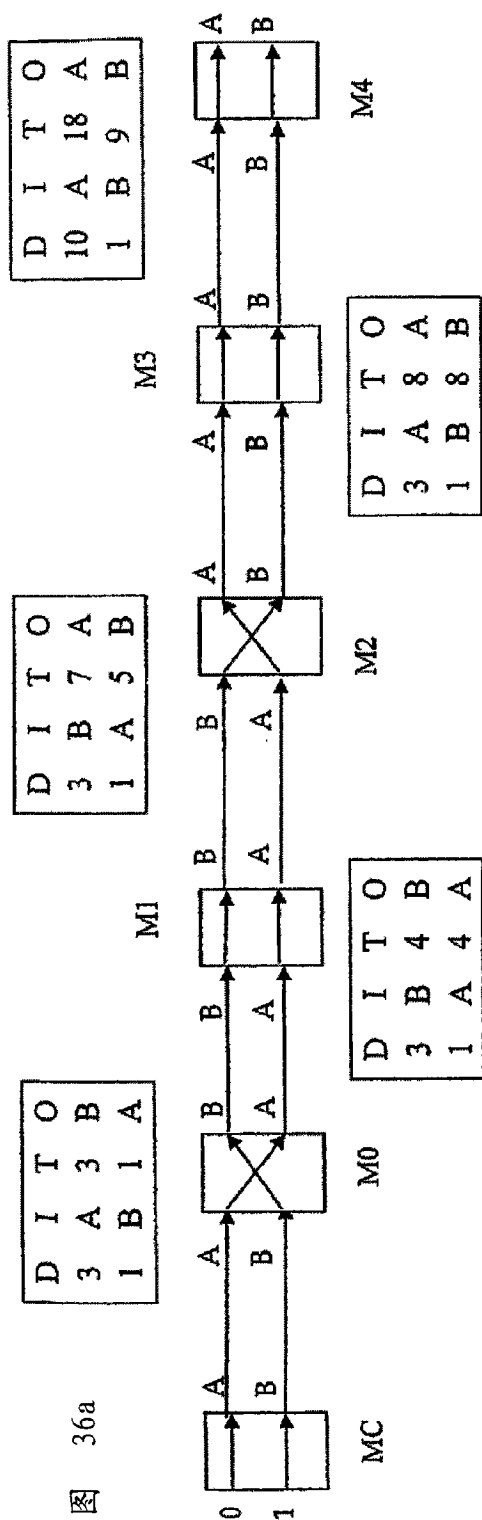
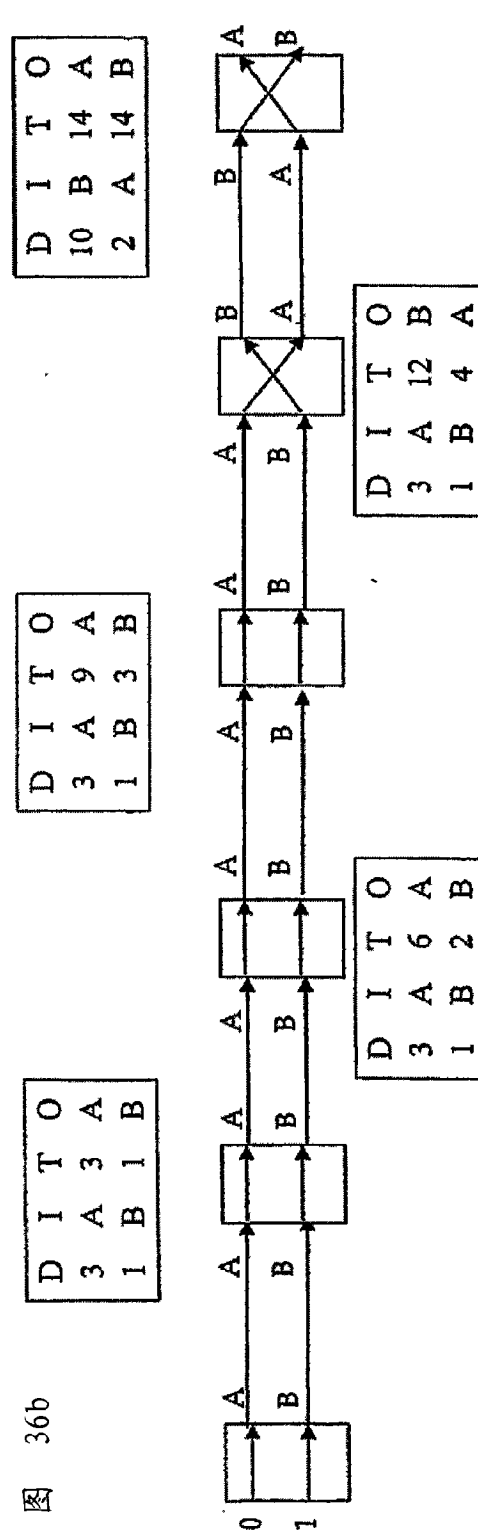


图 36b



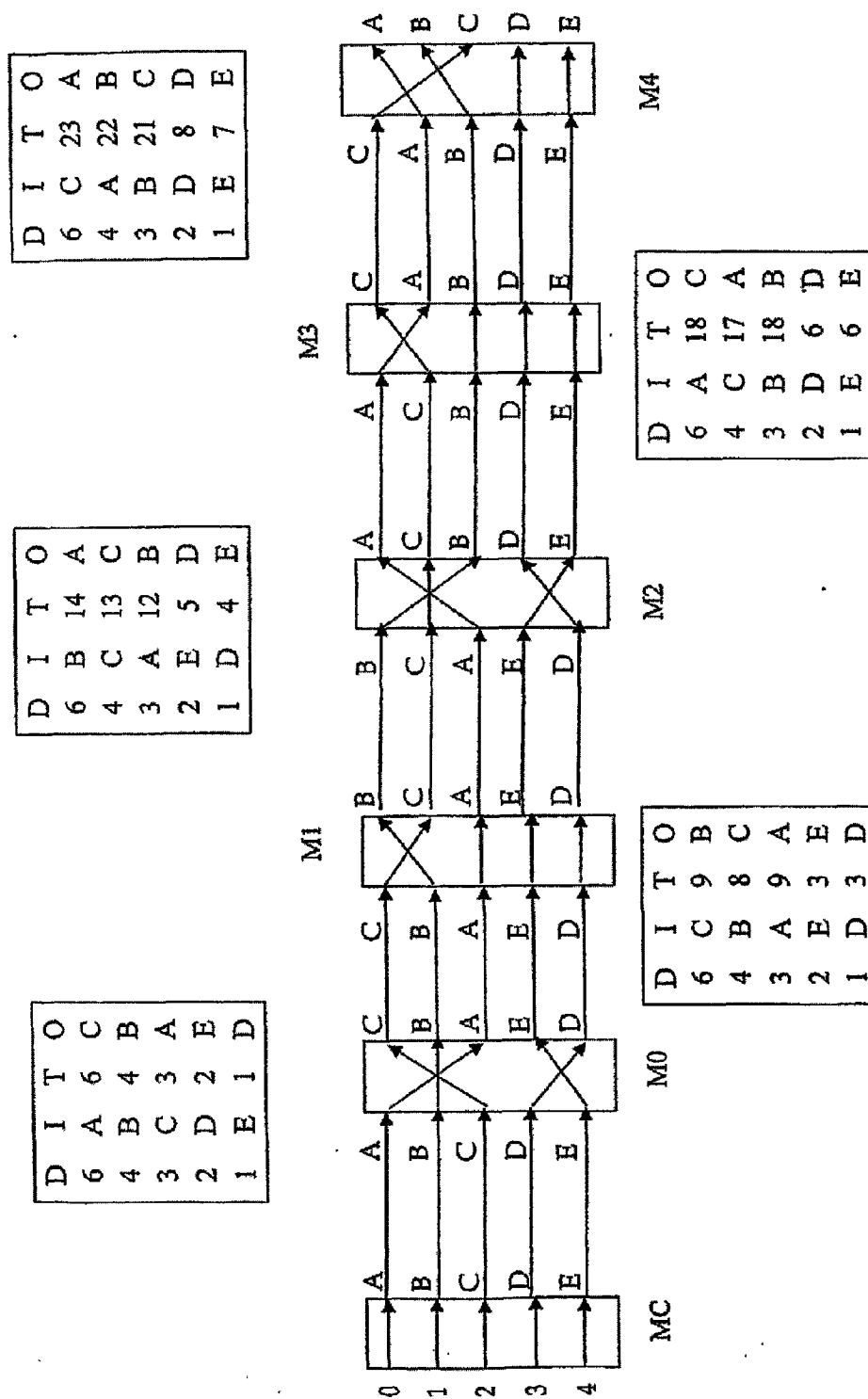


图 37

图 38a

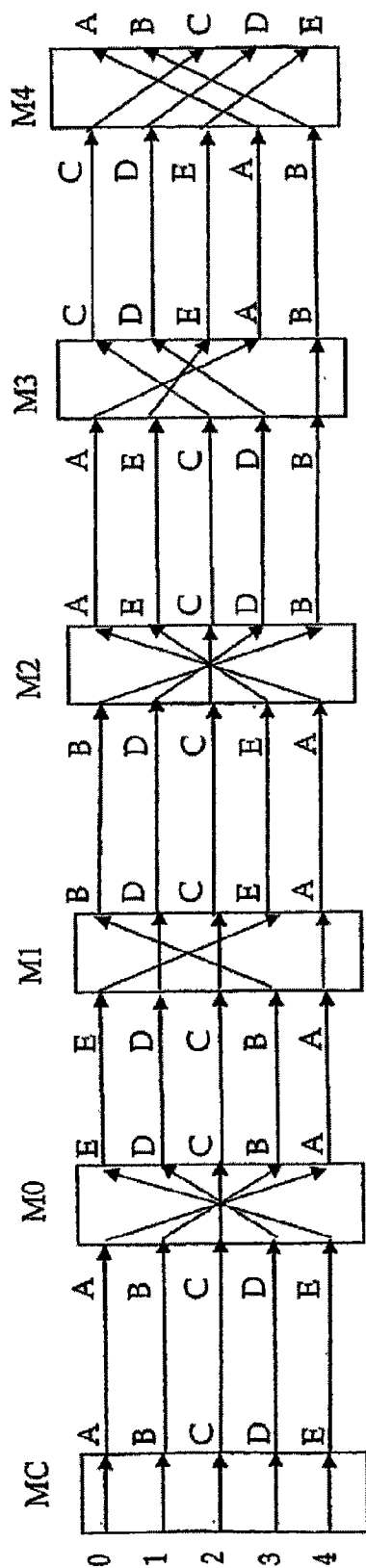


图 38c

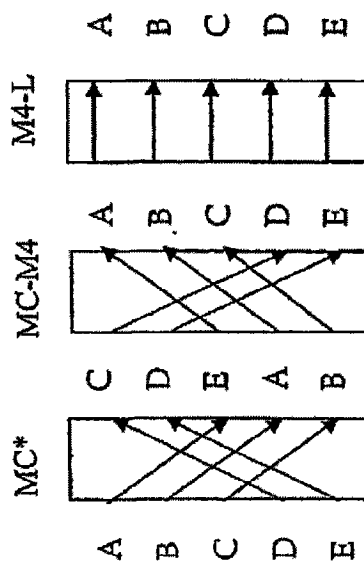


图 38b

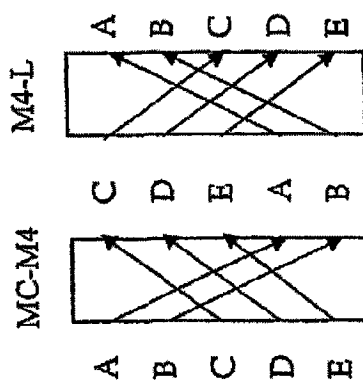


图 39a

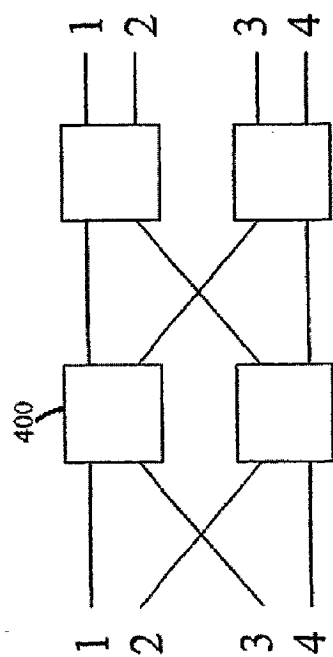


图 39b

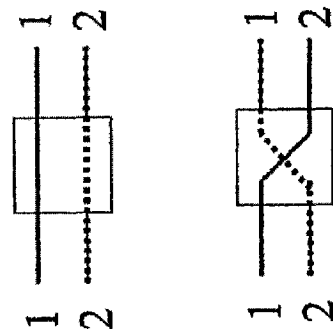


图 39c

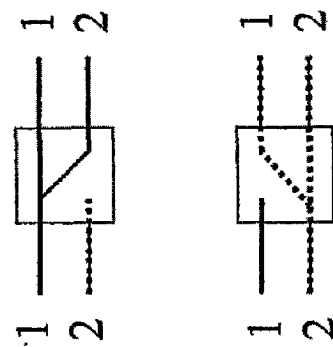


图 39d

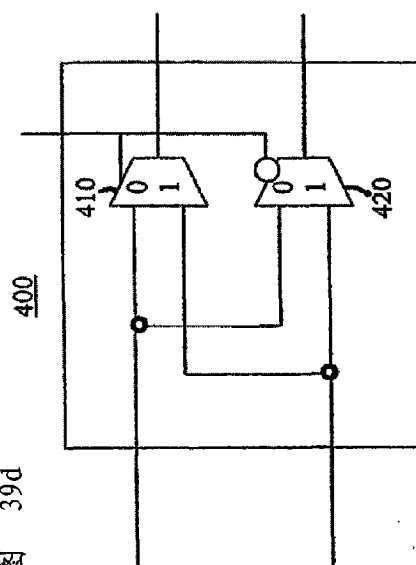


图 39e

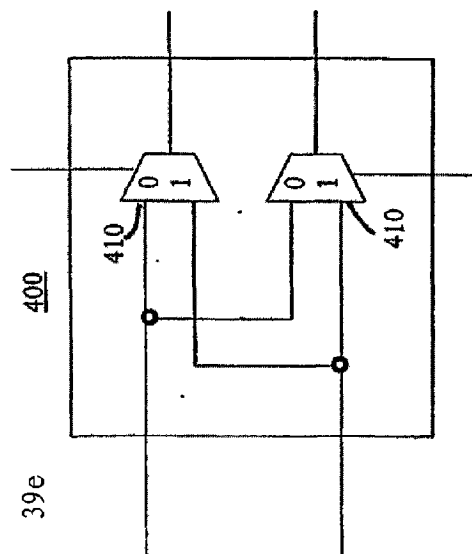


图 40b

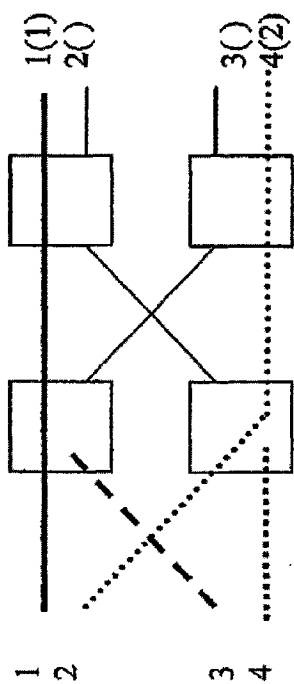


图 40a

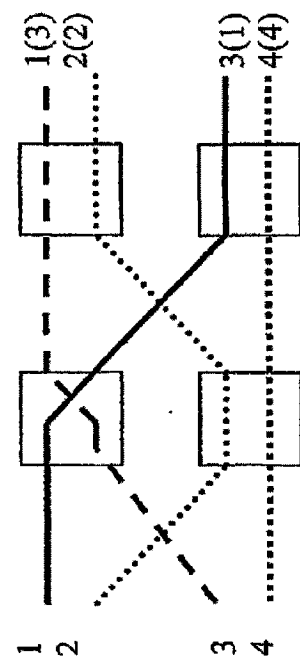


图 40d

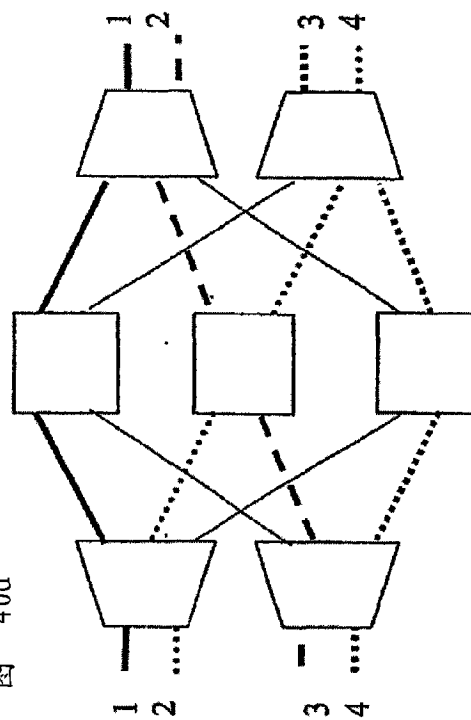
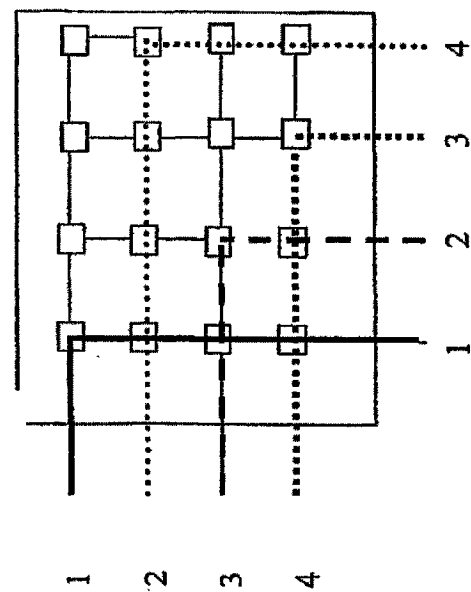


图 40c



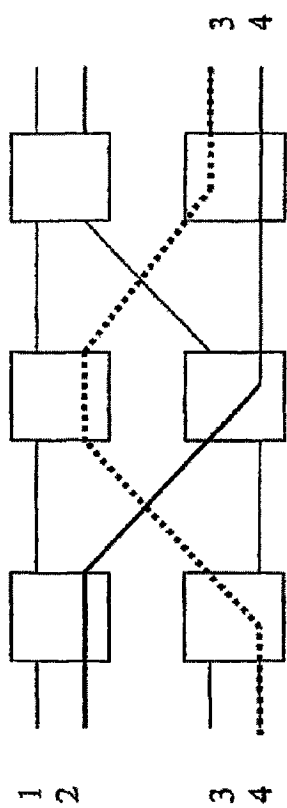


图 41a

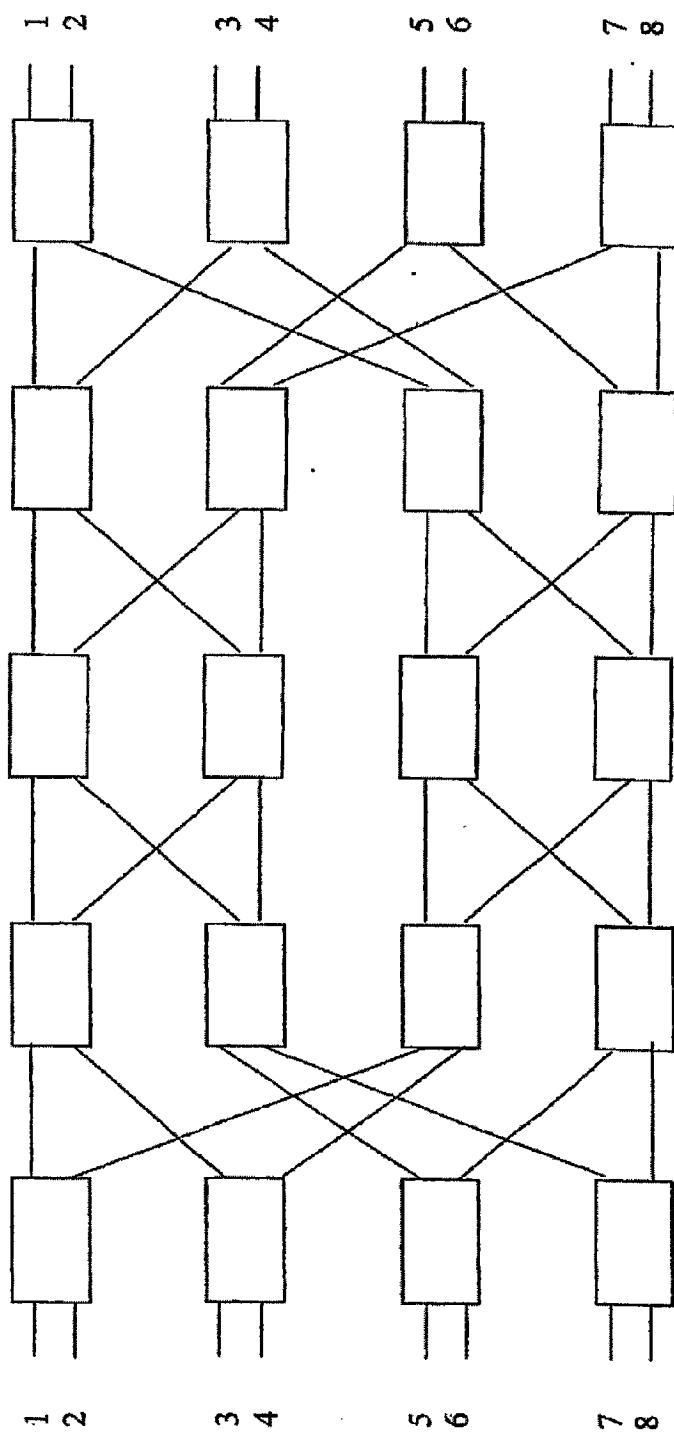


图 41b

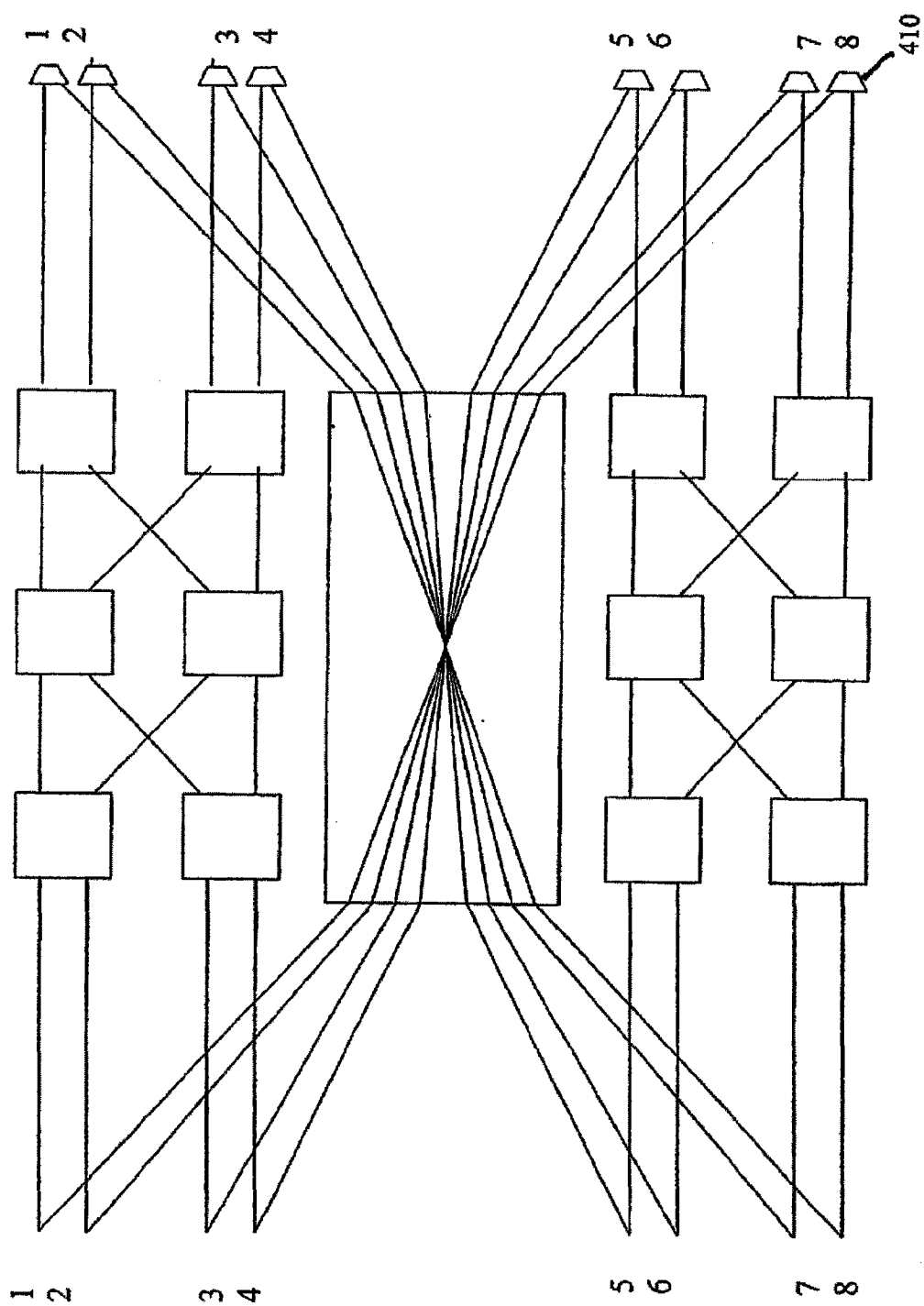


图 42

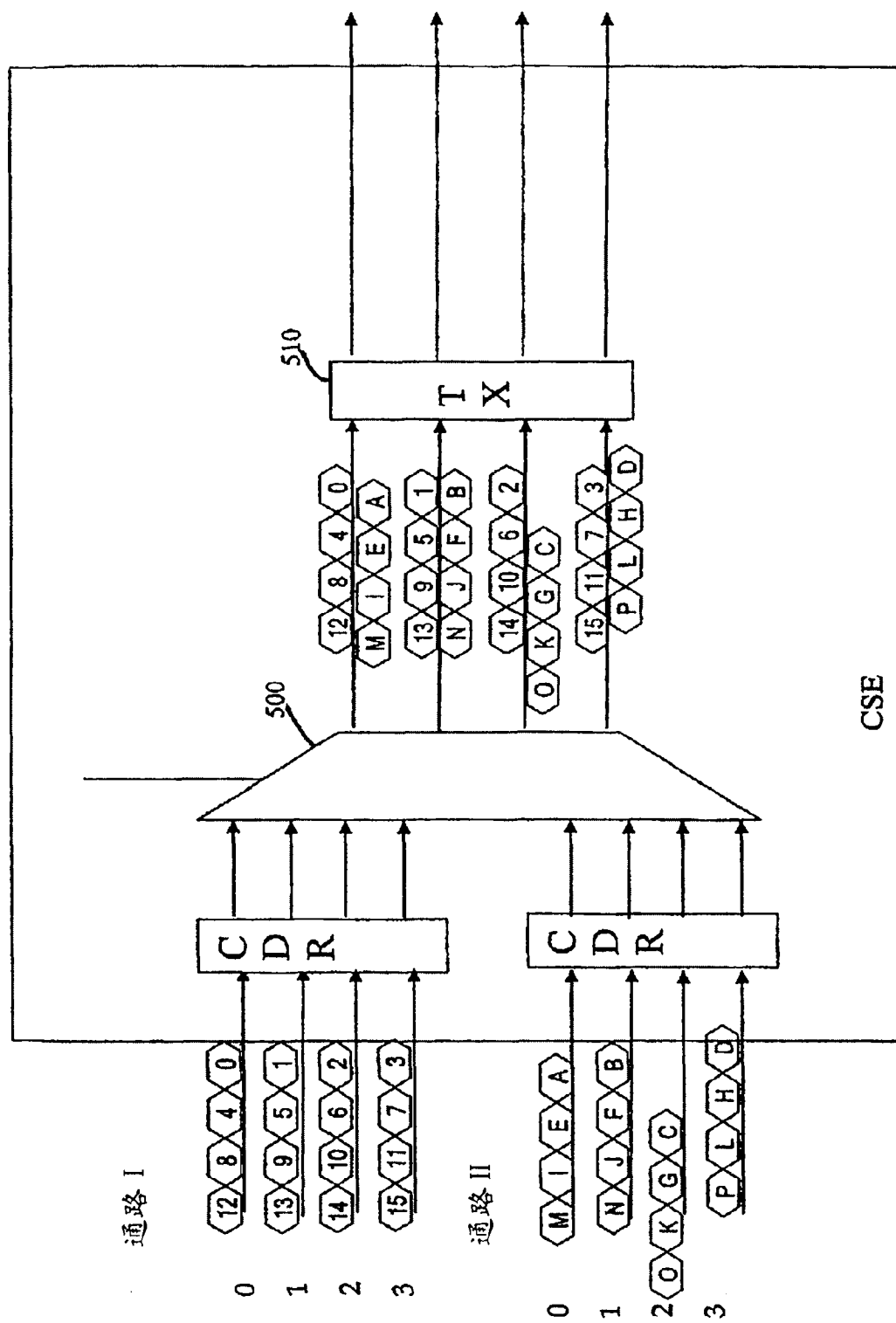


图 43

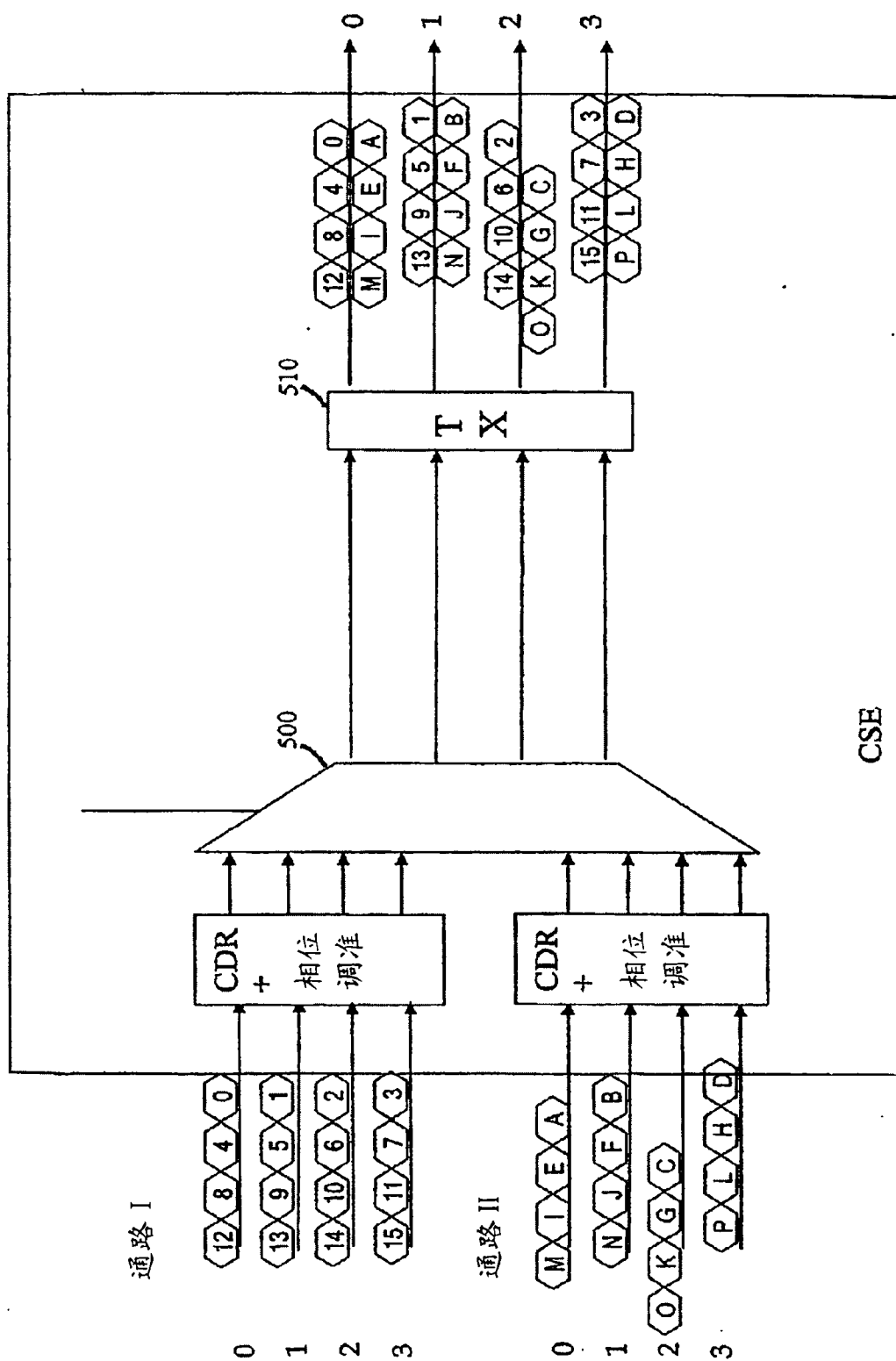


图 44

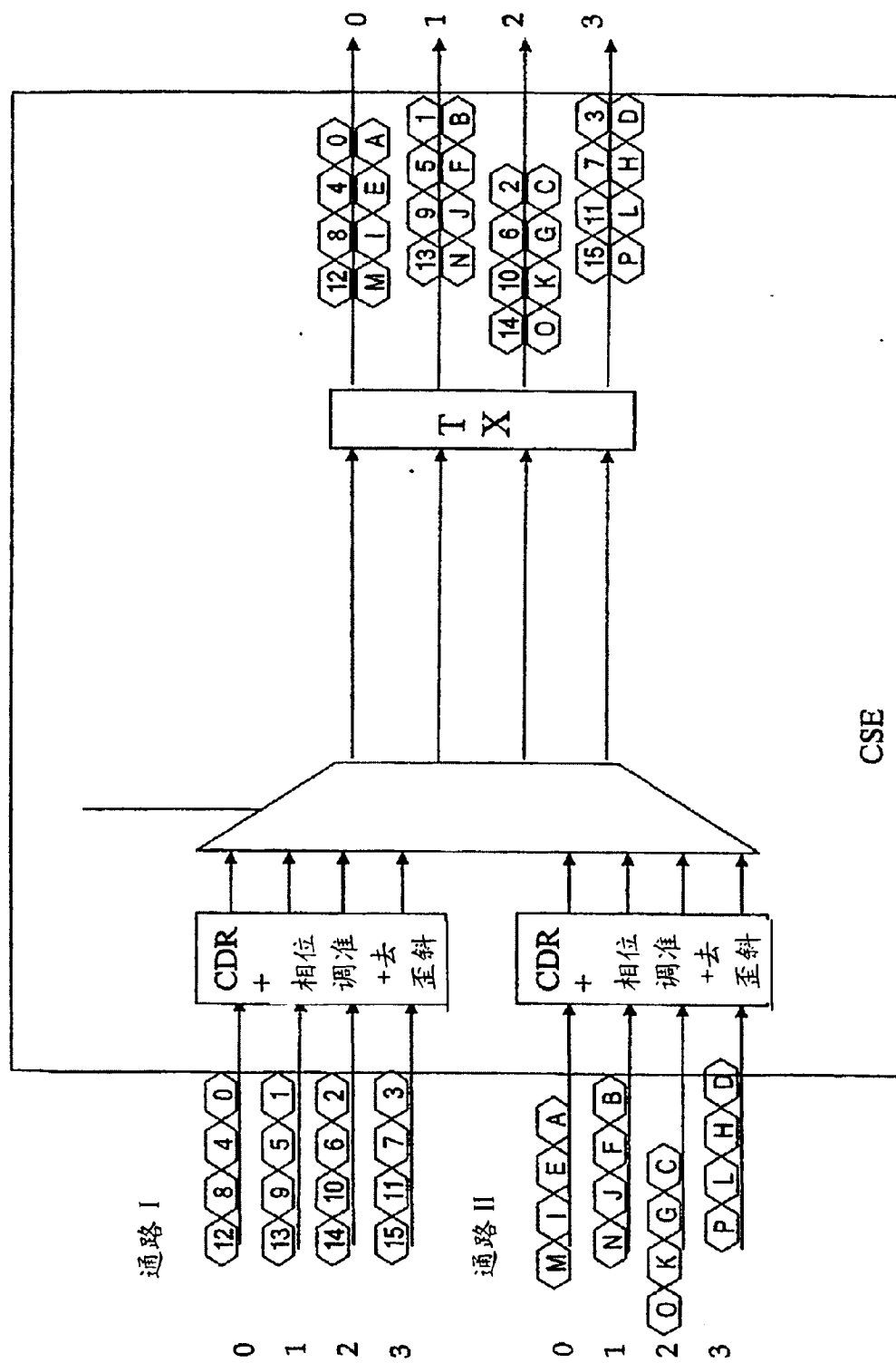


图 45

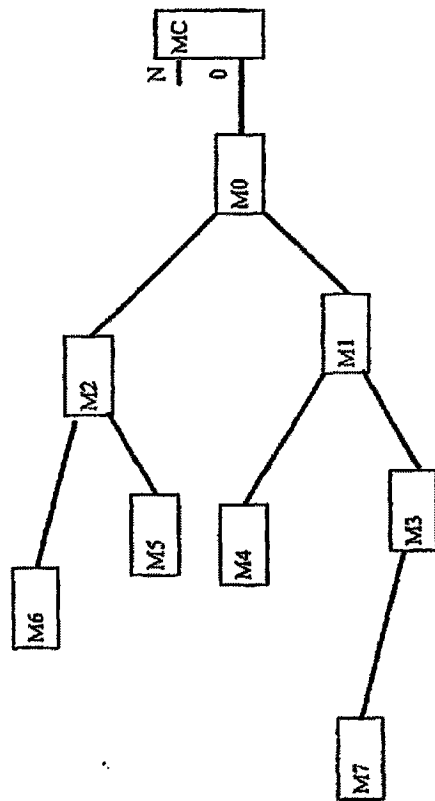


图 46a

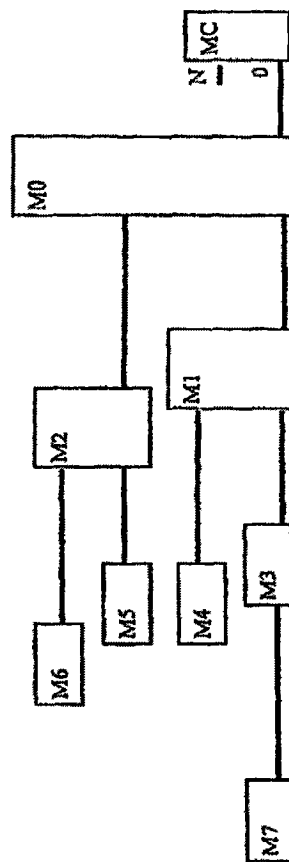


图 46b

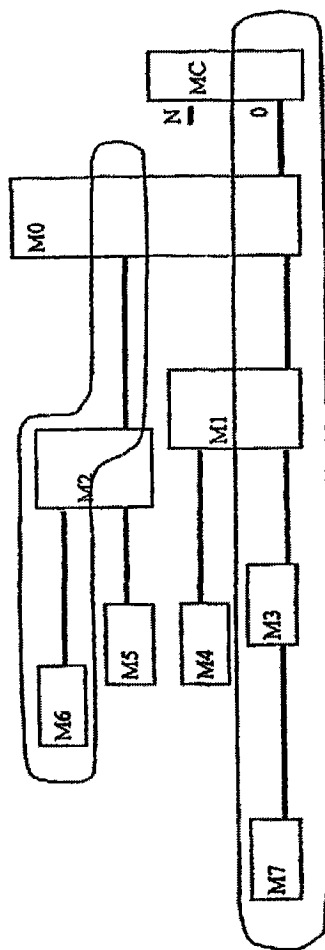


图 47a

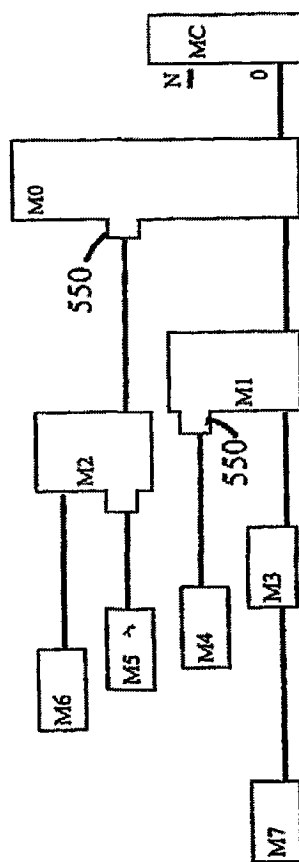


图 47b

dM_i_N=信号从M_i的TX I/O 端口发出, 经M_i到MC的#N端口线路, 到达MC的RX I/O端口的延迟
 $t_{M_i_N}(t)=1$ 比特信号在t时刻从M_i发送, 经M_i到MC的#N端口线路, 到达MC的时刻

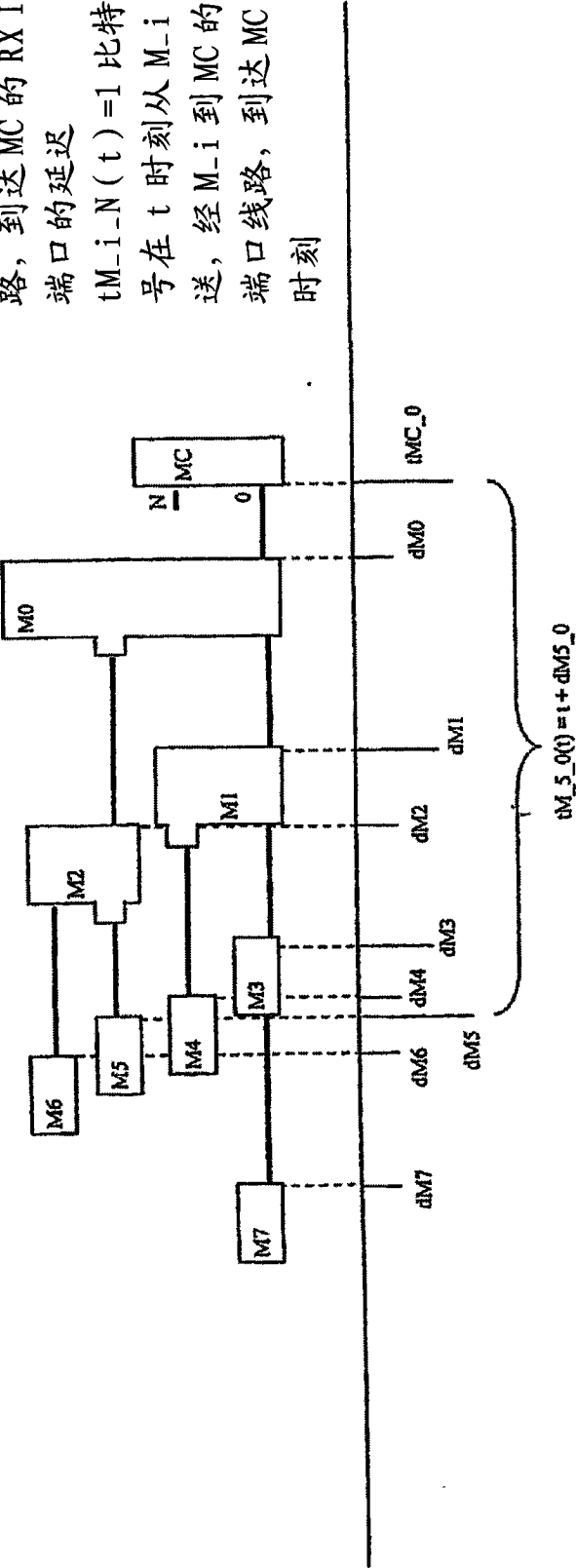
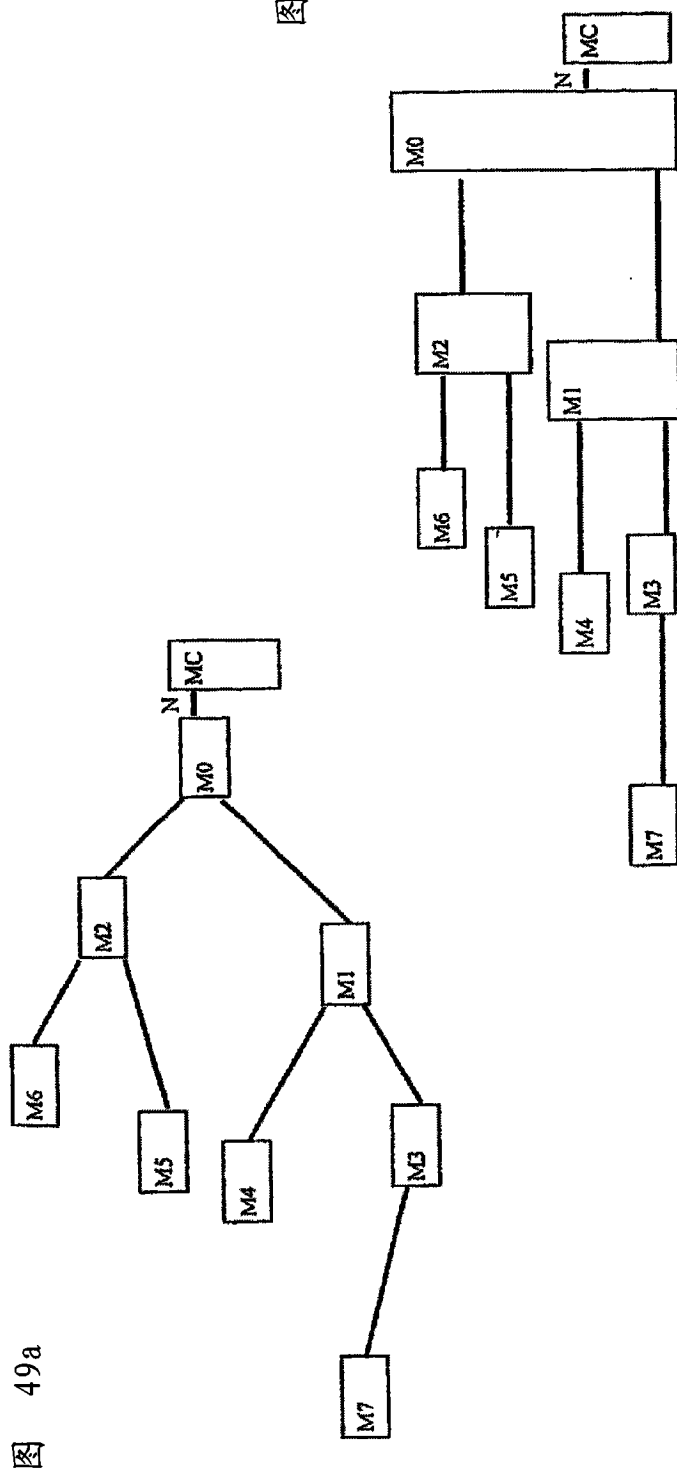


图 48



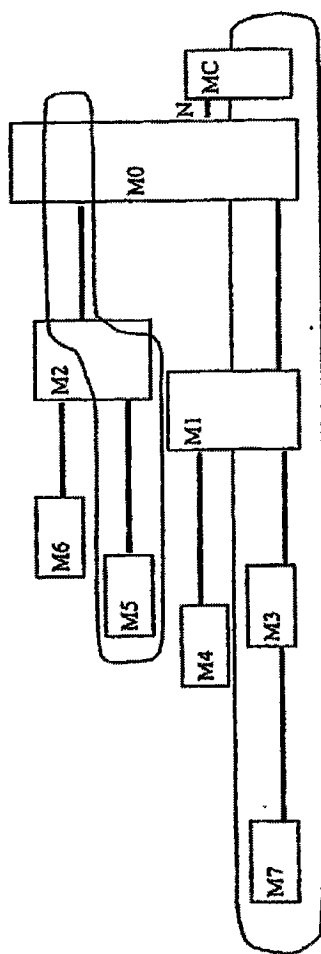


图 50a

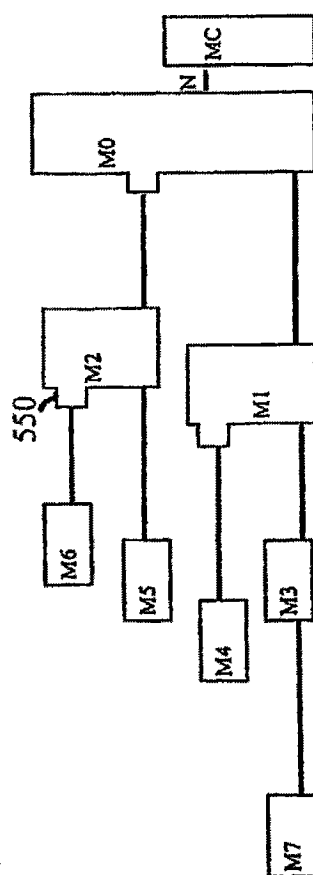


图 50b

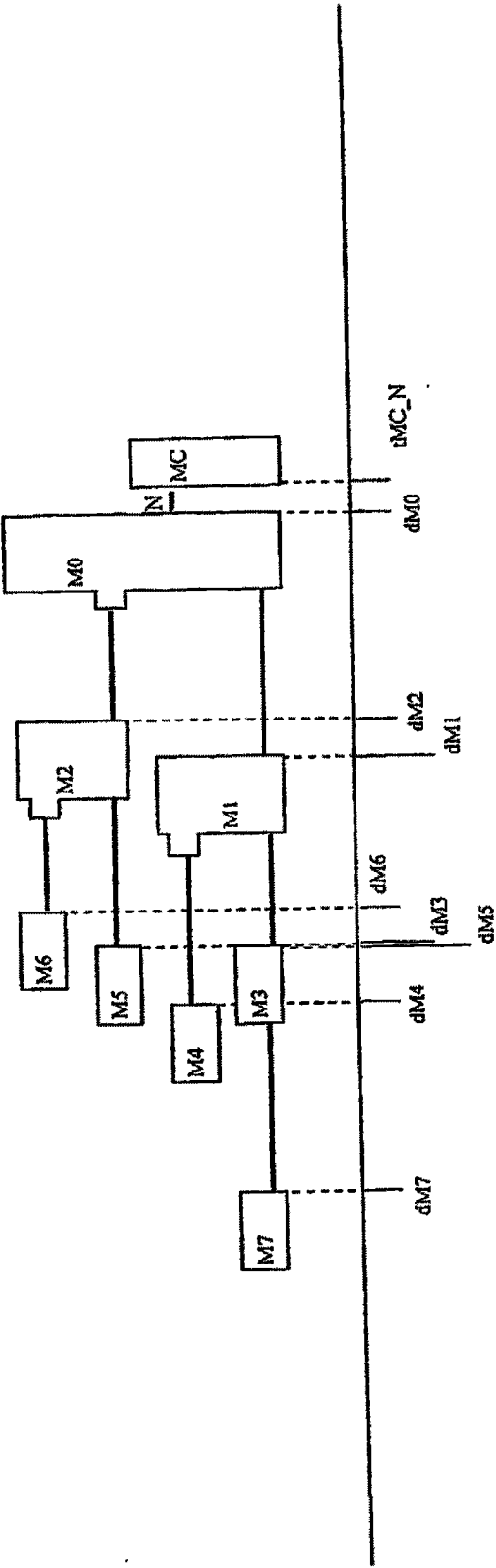
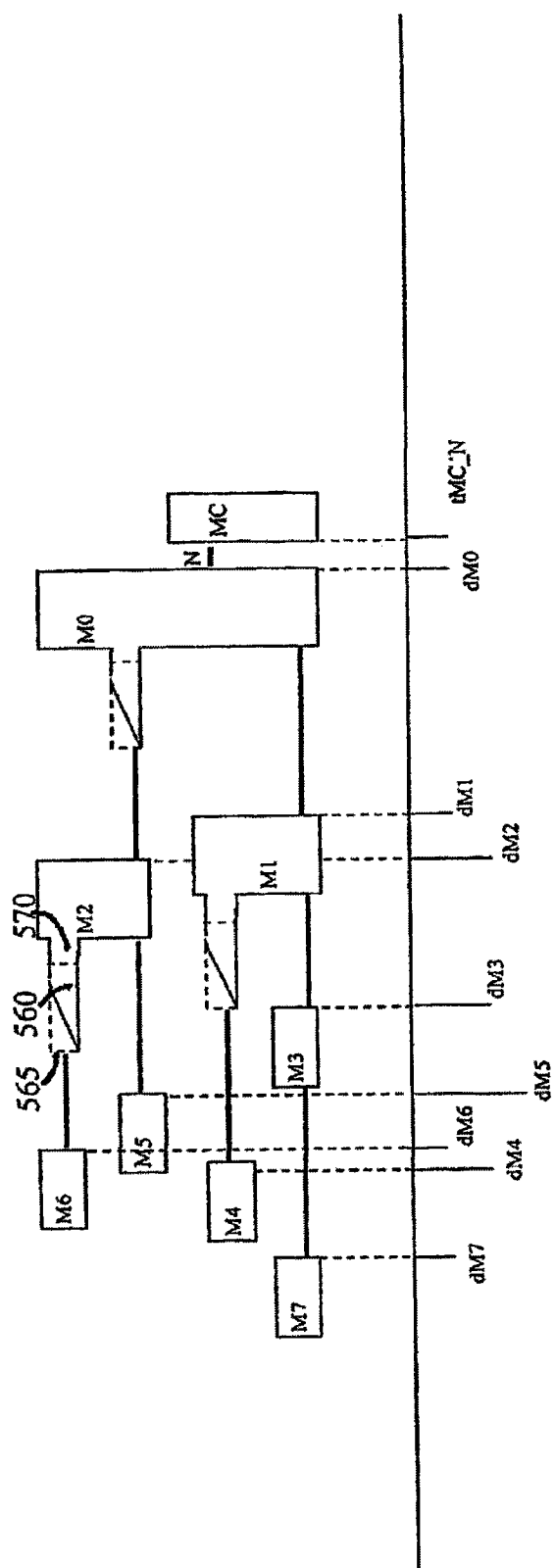


图 51



52

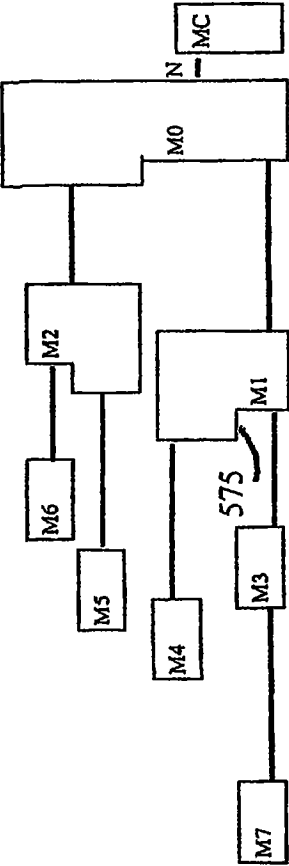


图 53

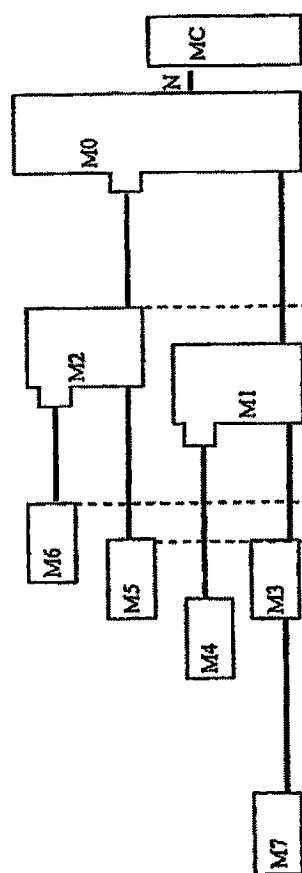


图 54a

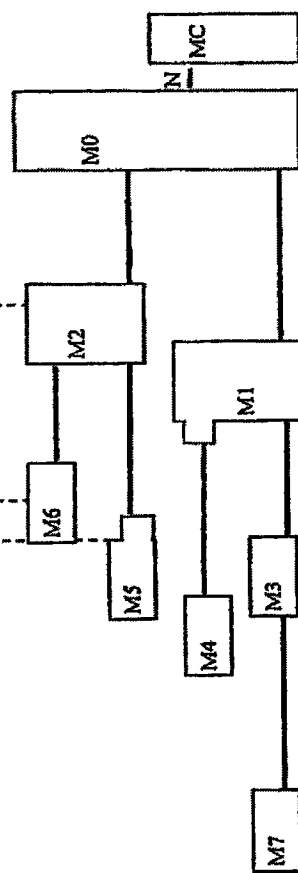


图 54b

图 55a

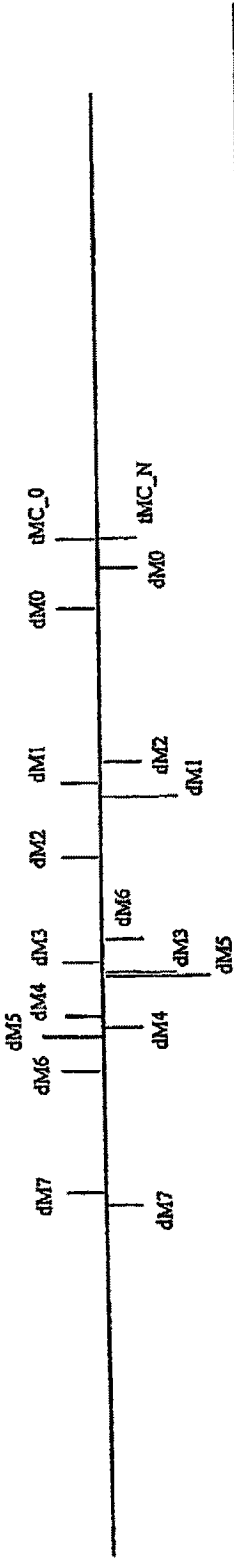
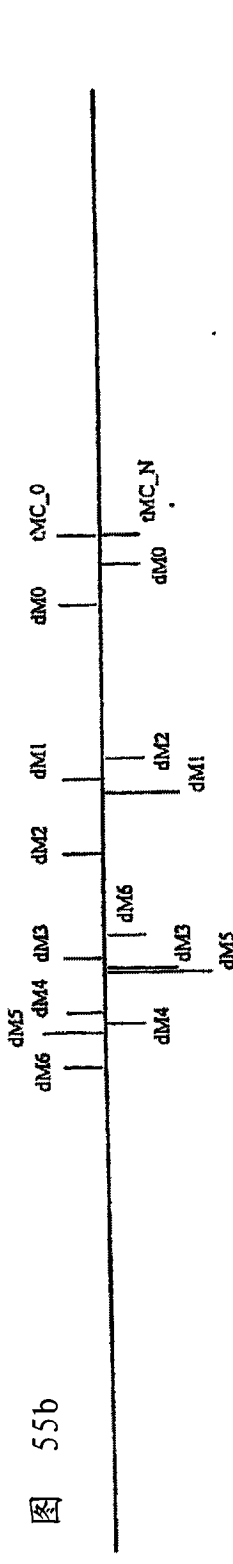


图 55b



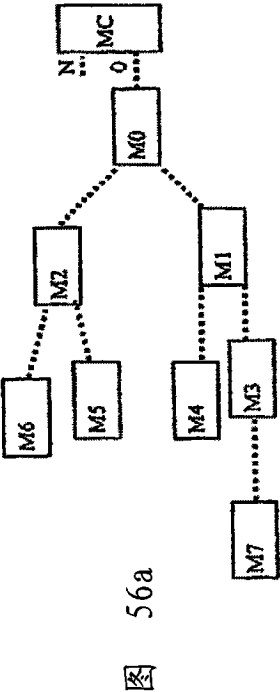
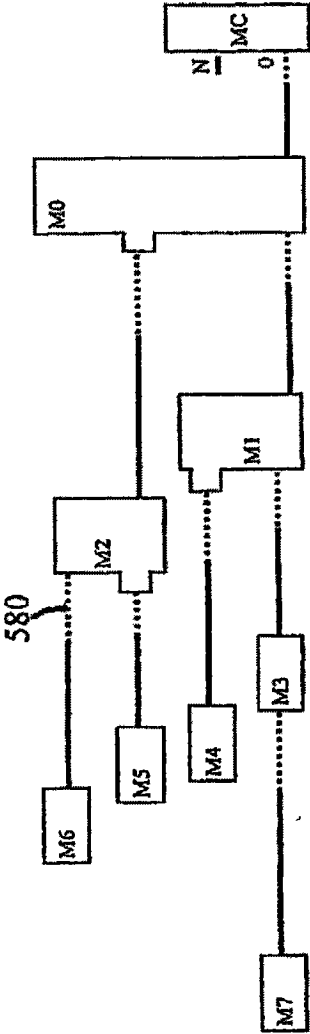


图 56b



sdM_i_N=一个从 MC 到达 M_i 的命令，其从 TX 端口发出的第一个比特，在从 M_i 经 MC 的线路#N 返回至 MC 的 RX I/O 端口的延迟时间

$stM_i_N(t)$ =为响应在 t 时刻从 MC 发出到达 M_i 的命令，一个比特从 M_i 经 MC 的线路#N 返回 MC 的时刻

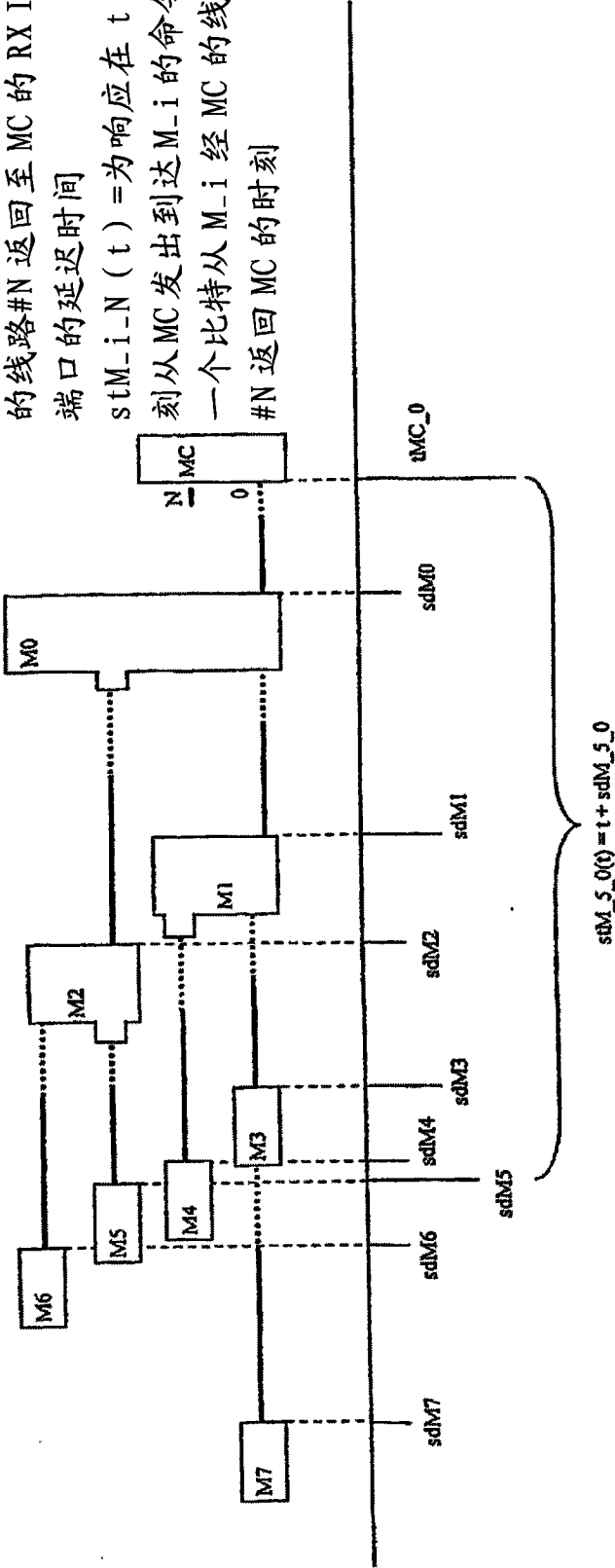
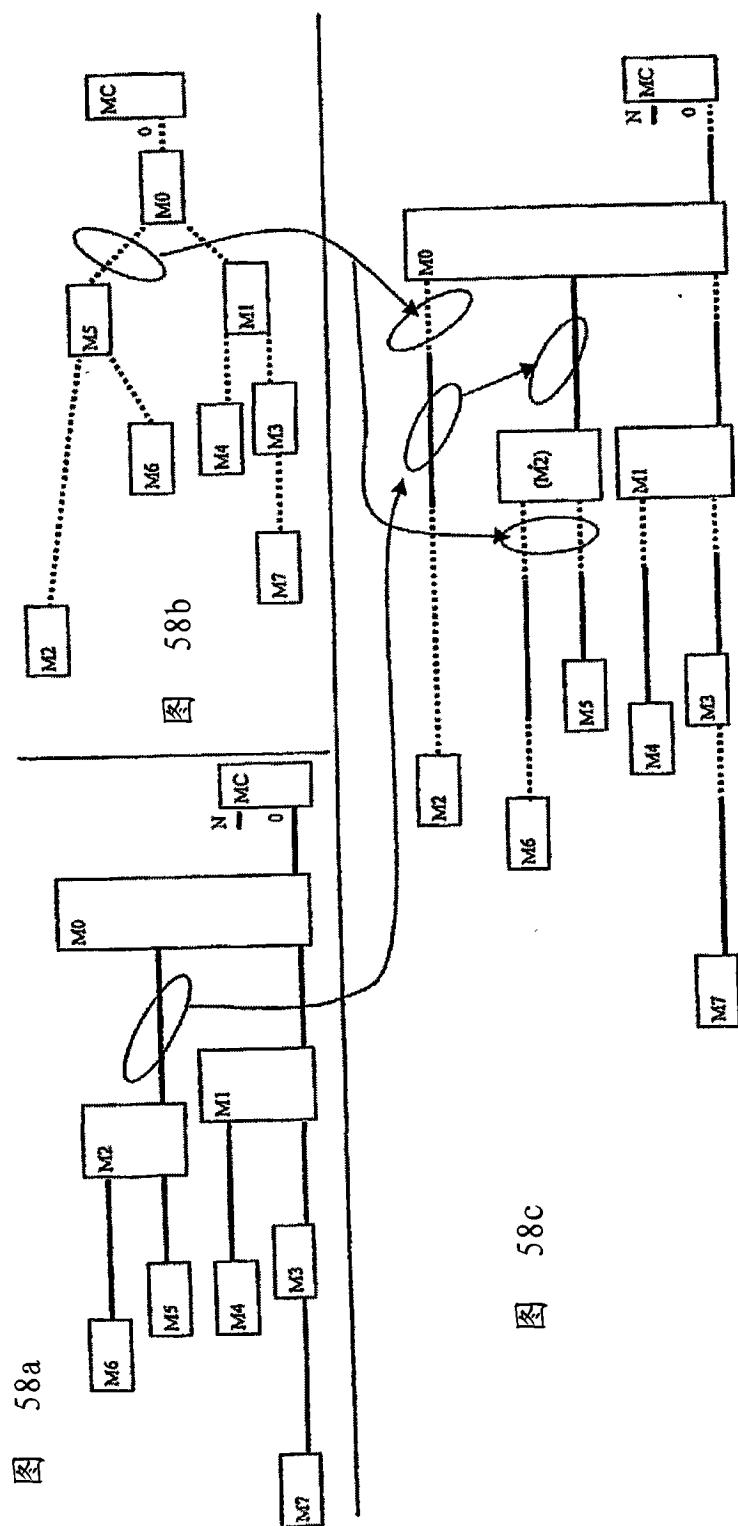


图 57



sdM_i_N=一个从 MC 至
达 M_i 的命令，其从
TX 端口发出再从 M_i
发出的第一个比特，在
经 MC 的线路#N 返回 MC
的 RX I/O 端口的延迟
时间
 $stM_i_N(t)$ =为响应
在 t 时刻从 MC 发出到
达 M_i 的命令，一个比
特从 M_i 经 MC 的线路
#N 返回 MC 的时刻

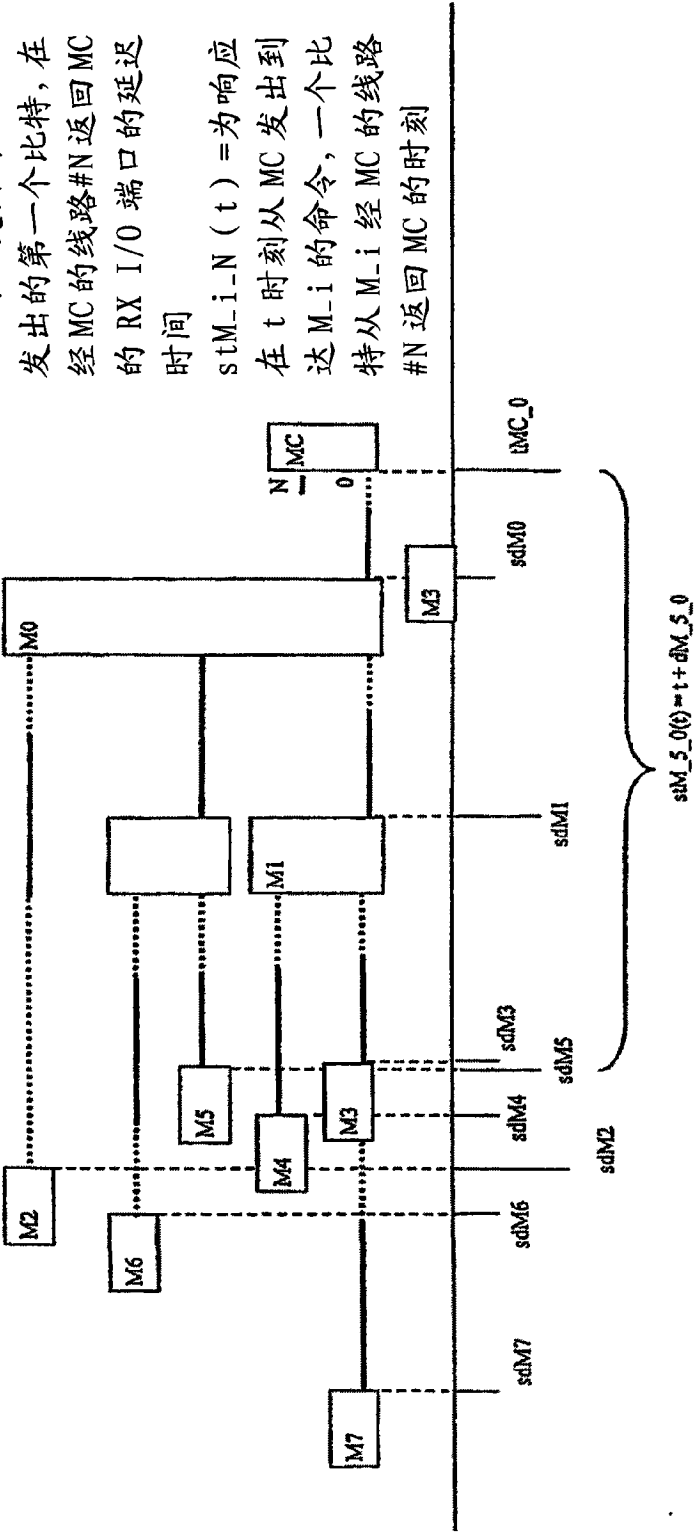


图 59

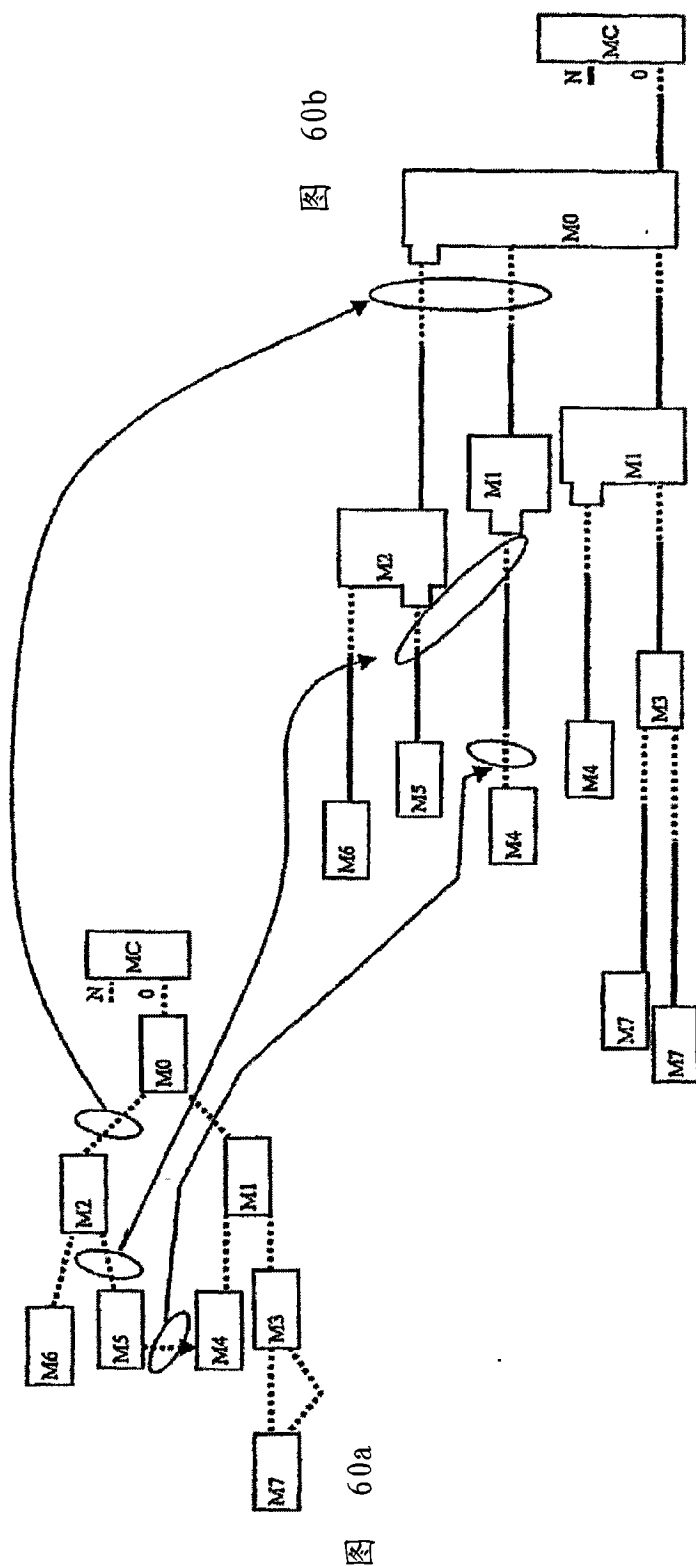


图 61a

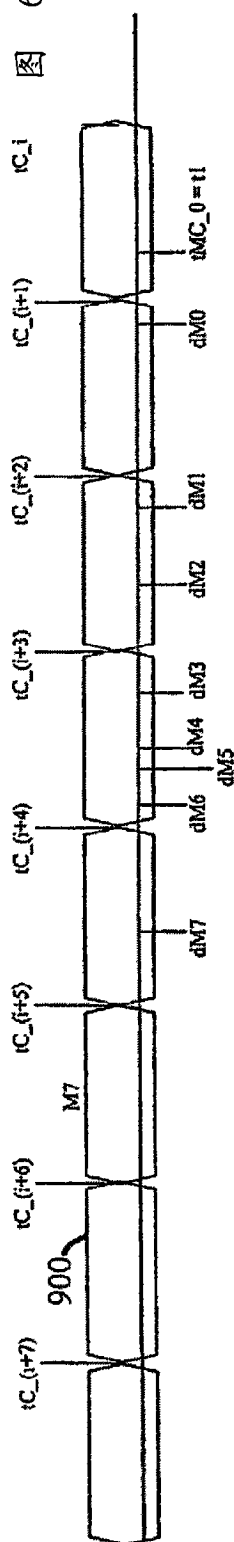


图 61b

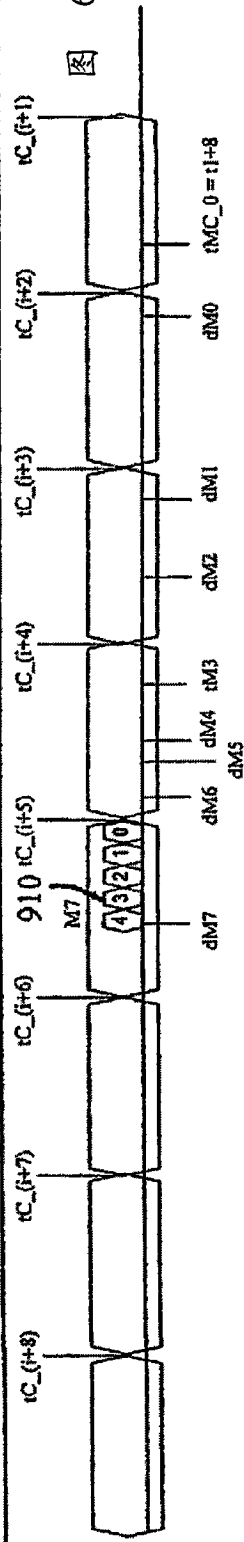
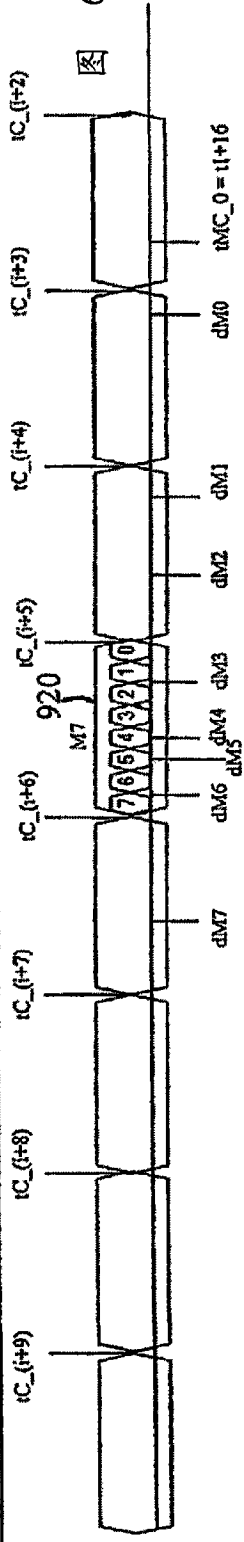
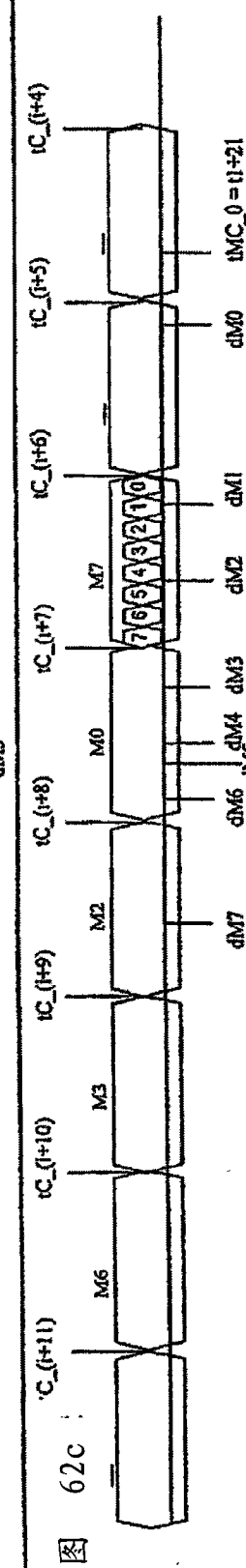
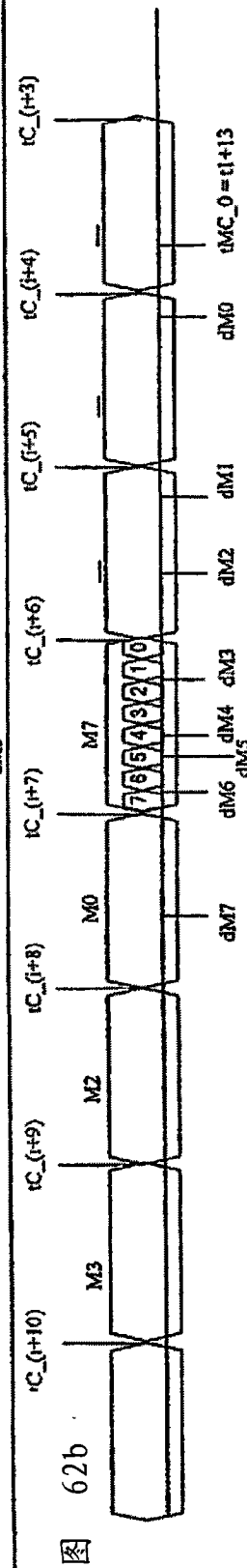
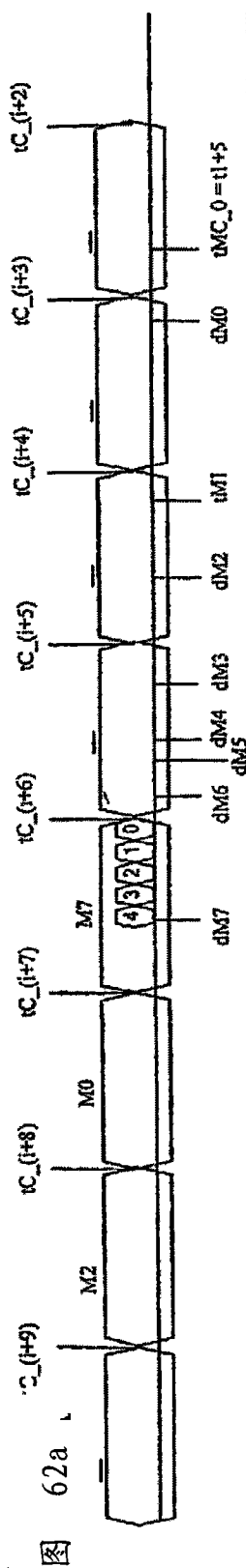
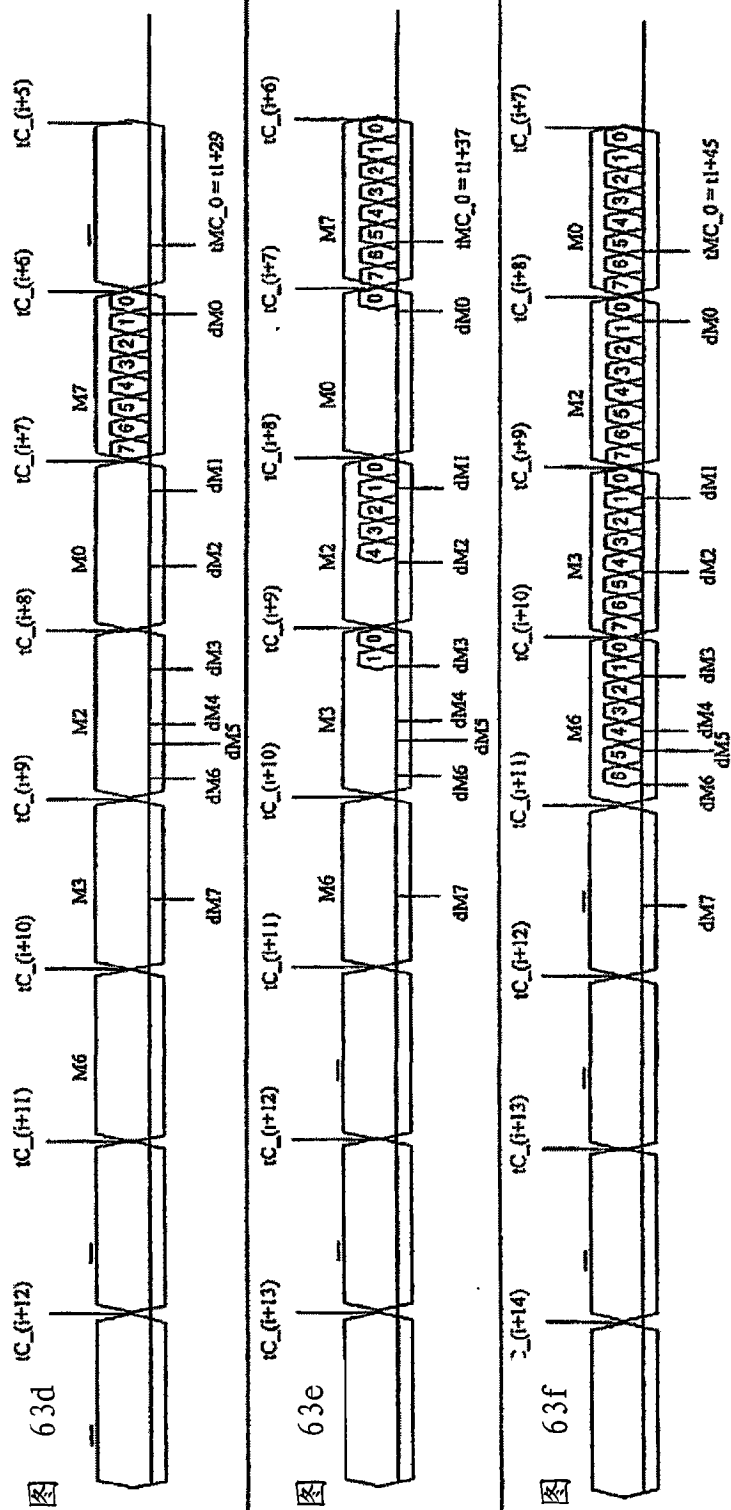
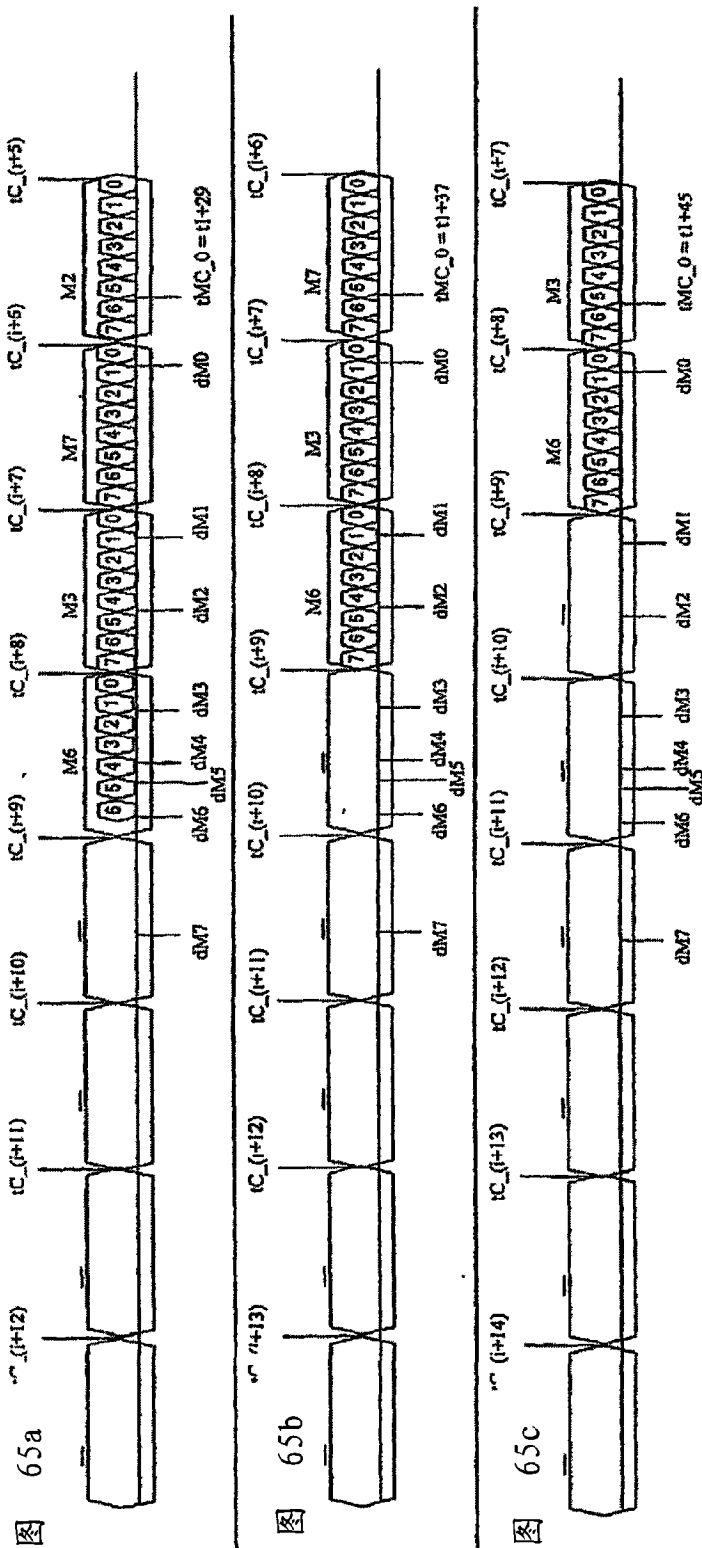


图 61c









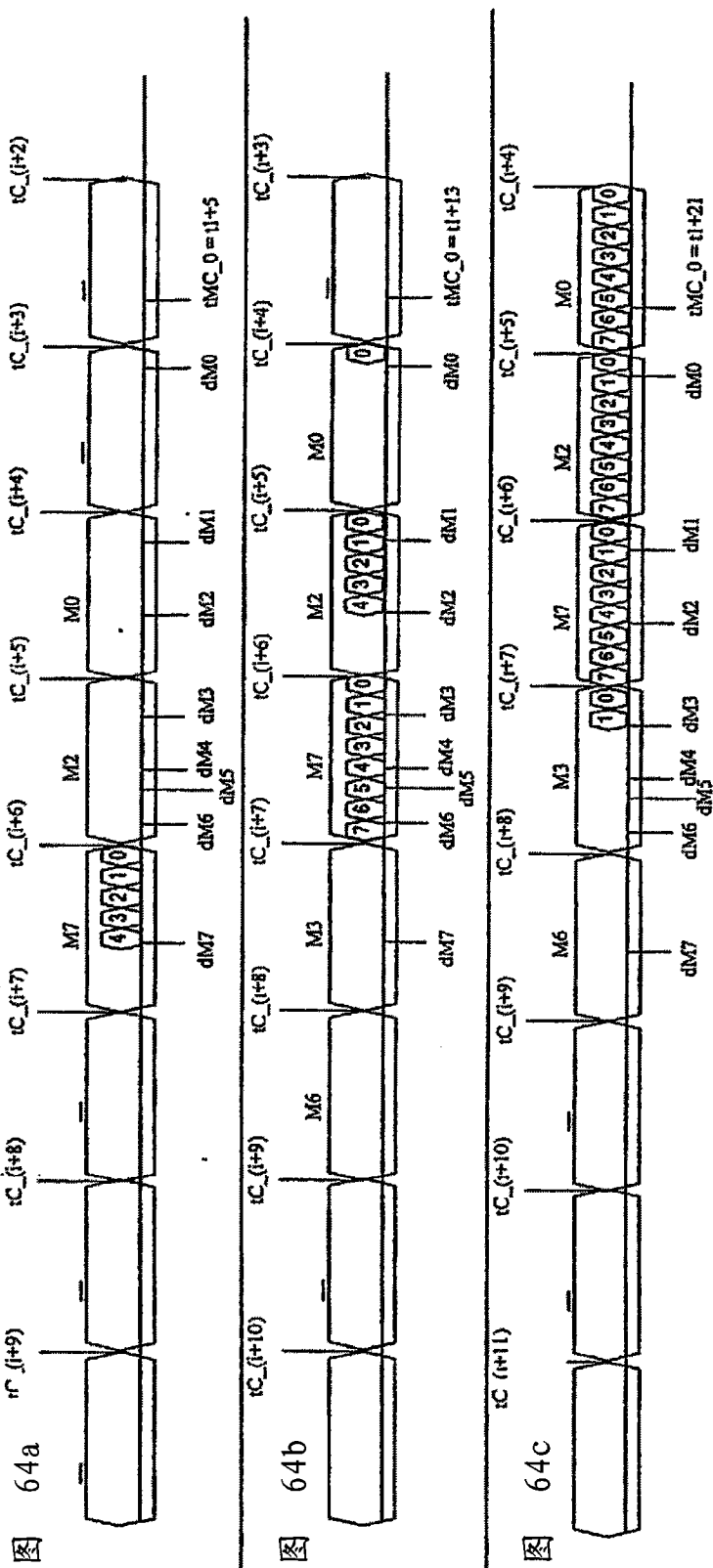


图 66a

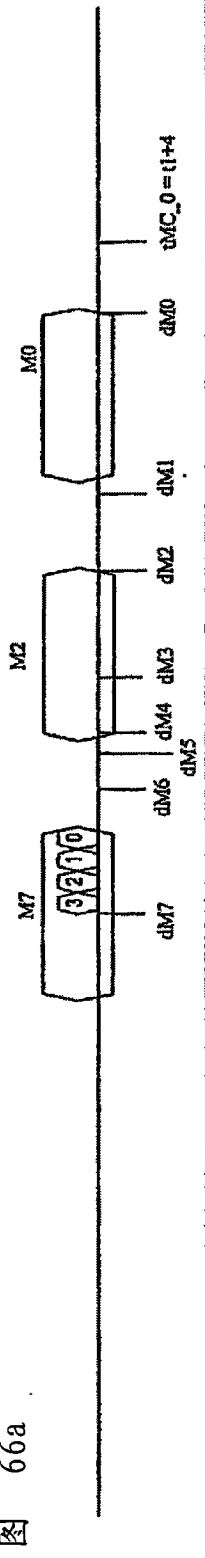


图 66b

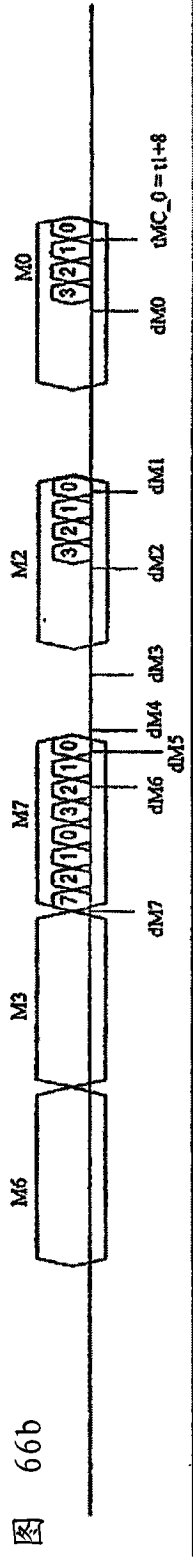


图 66c

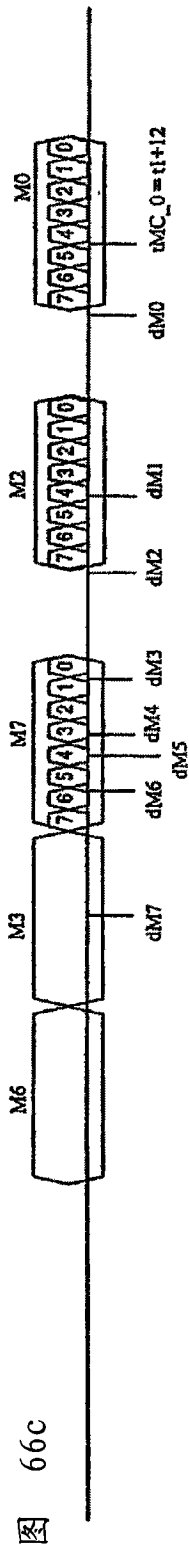


图 67a

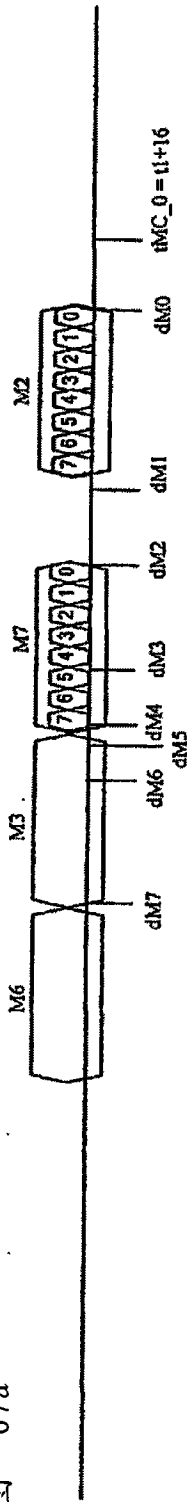


图 67b

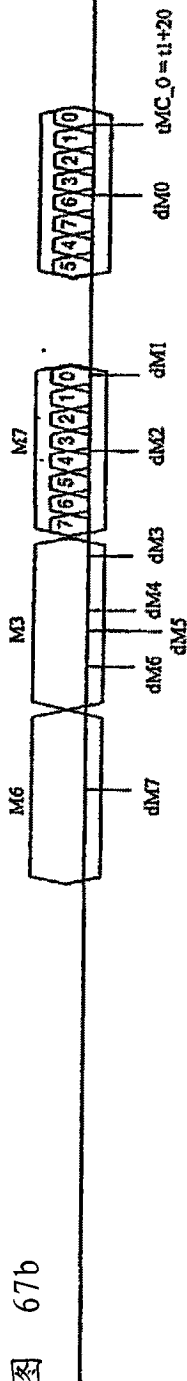
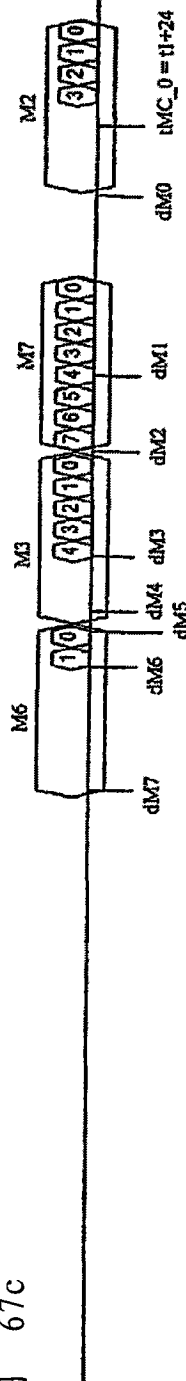
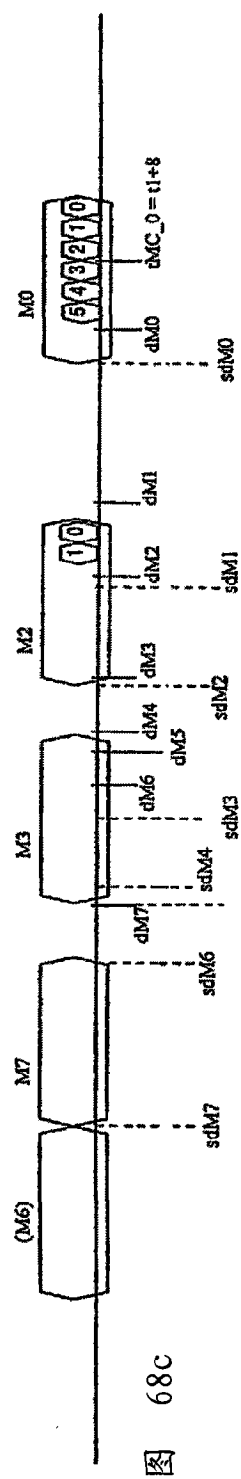
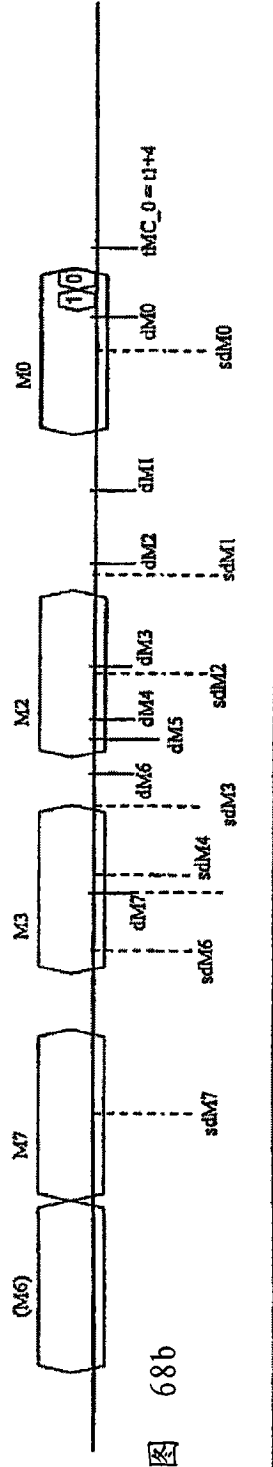
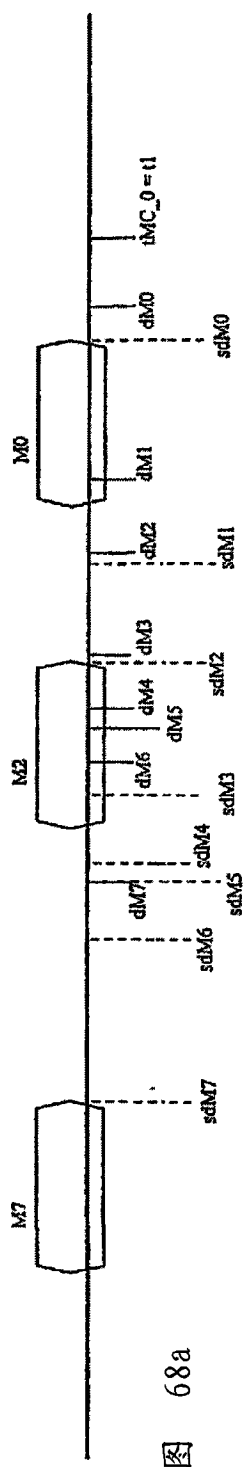
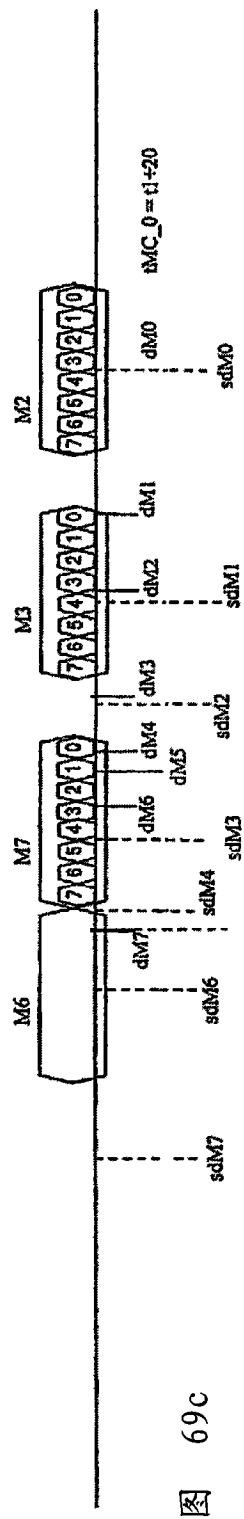
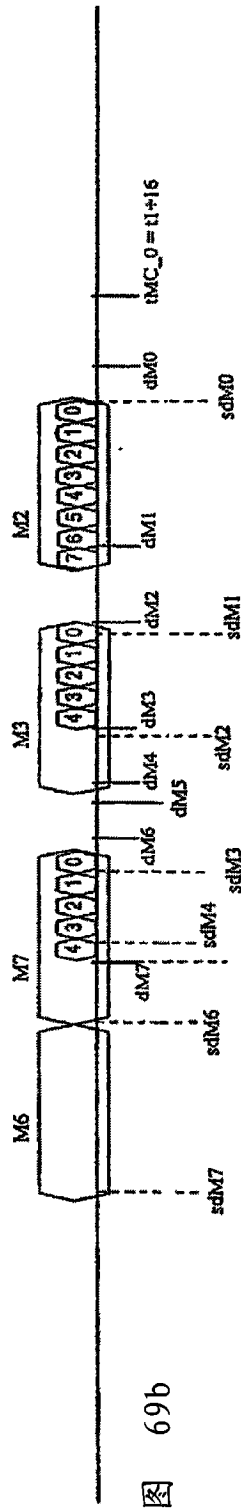
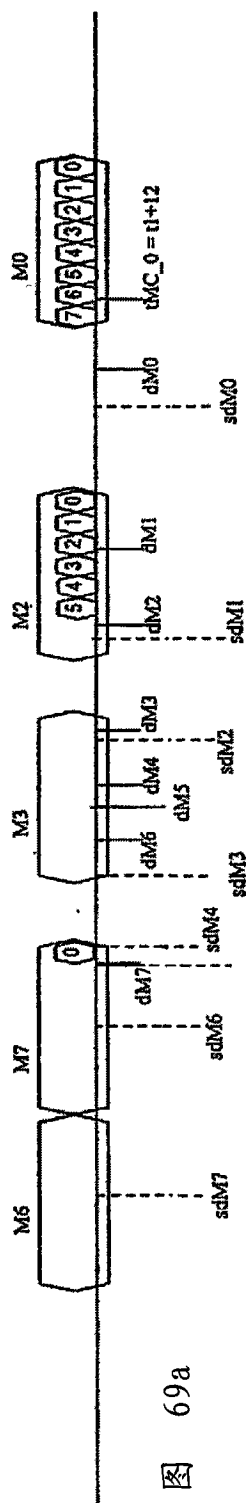
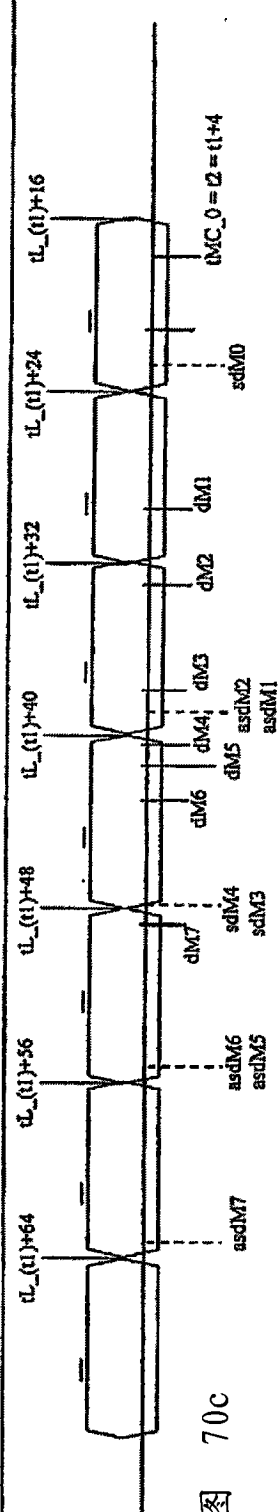
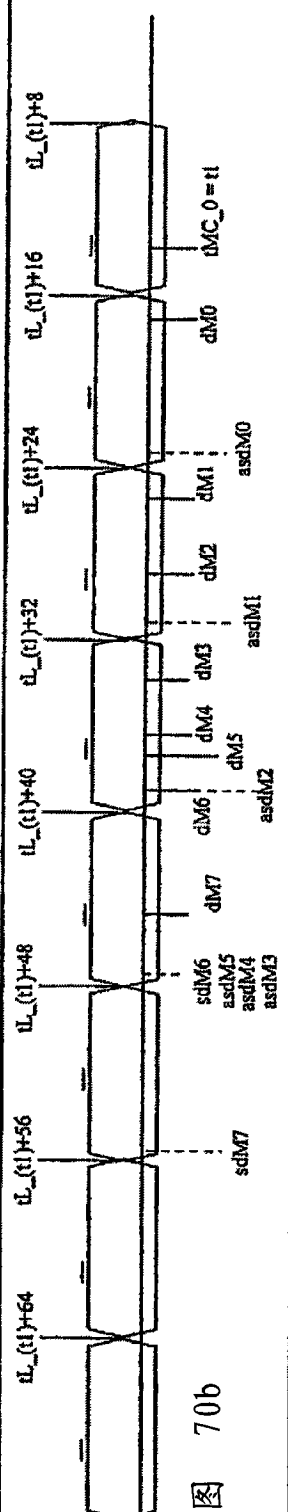
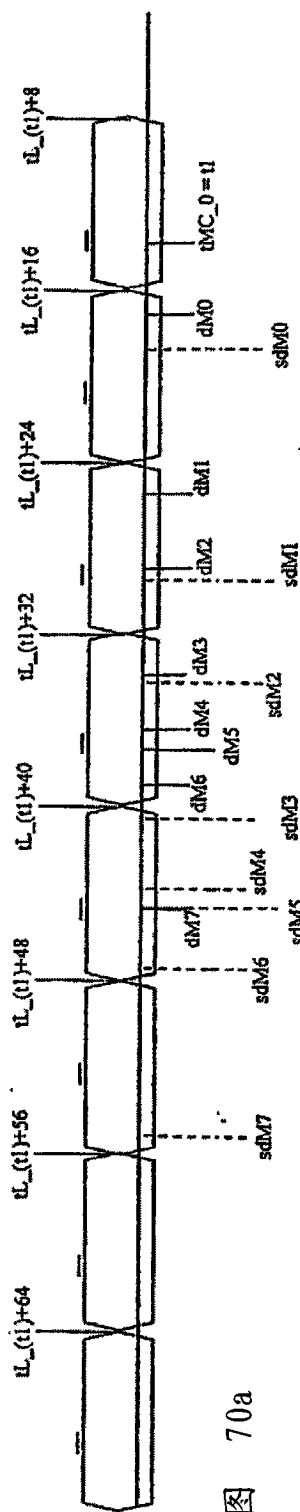


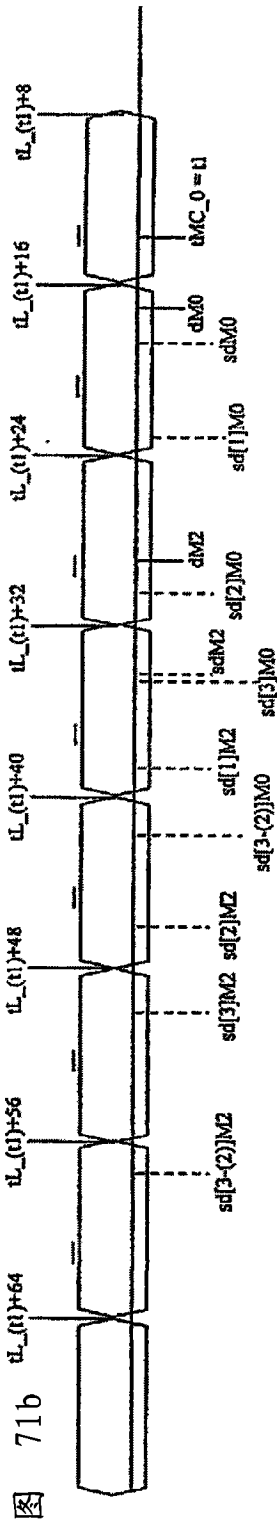
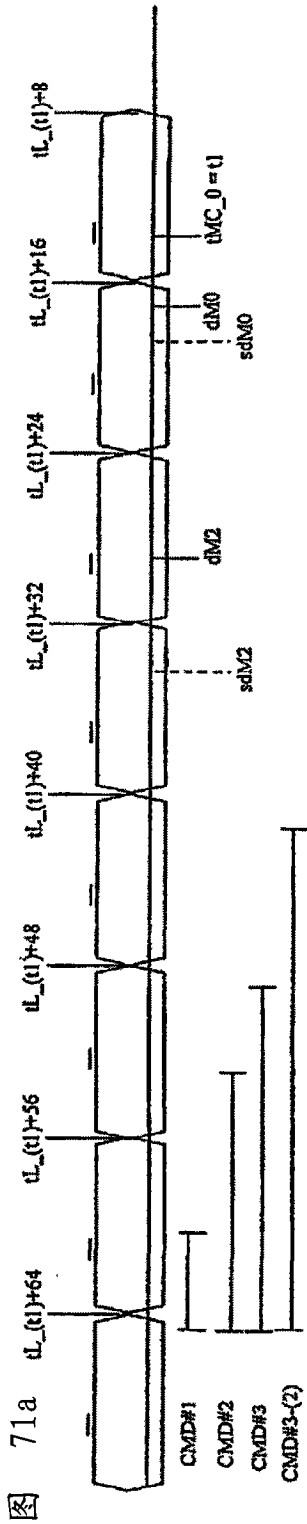
图 67c











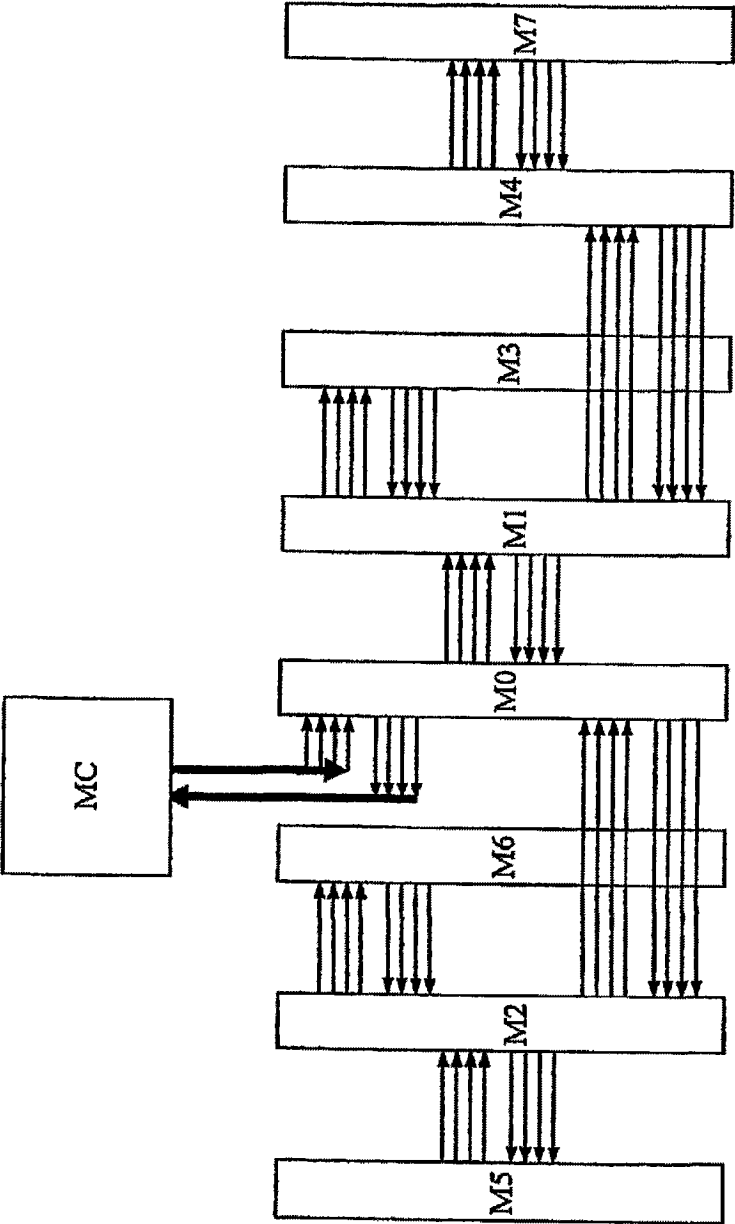


图 72

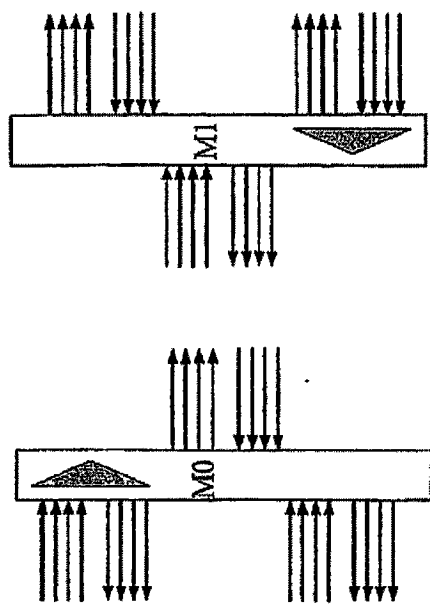


图 73b

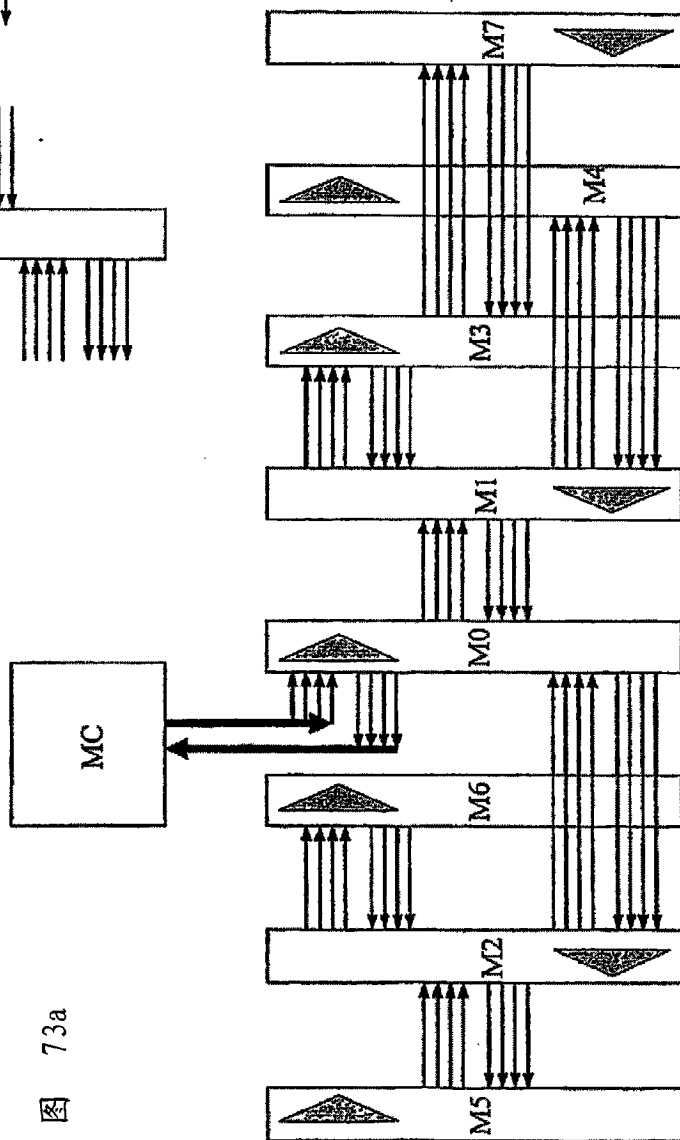


图 73a

图 74a

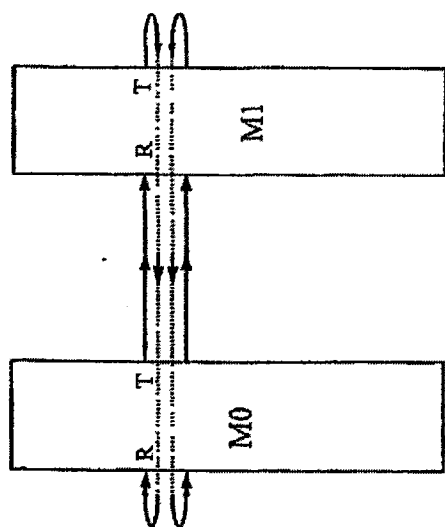


图 74b

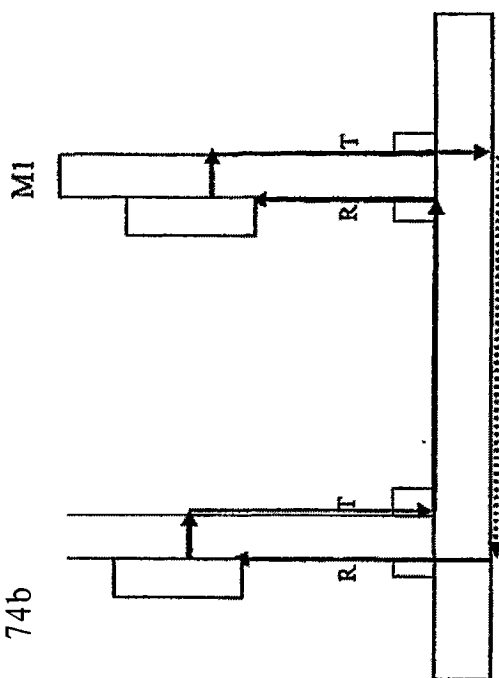


图 74c

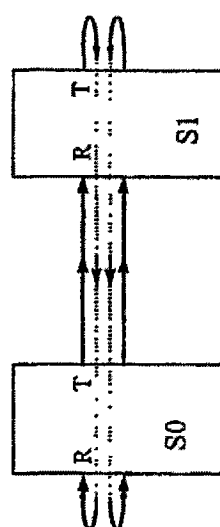


图 74d



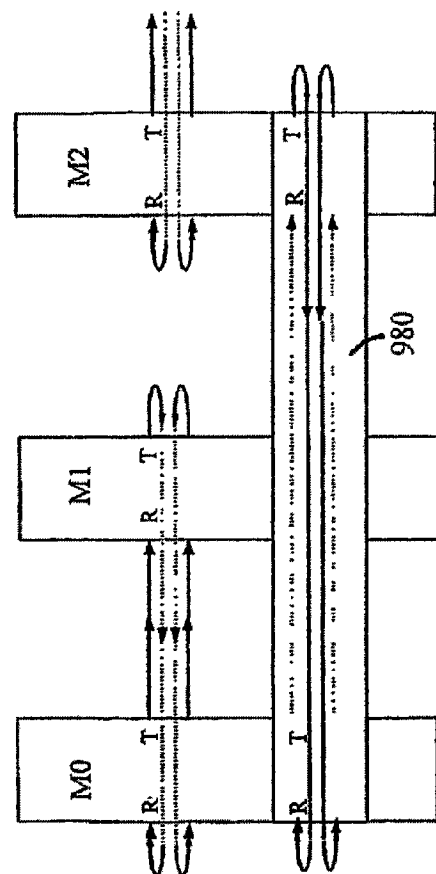


图 75a

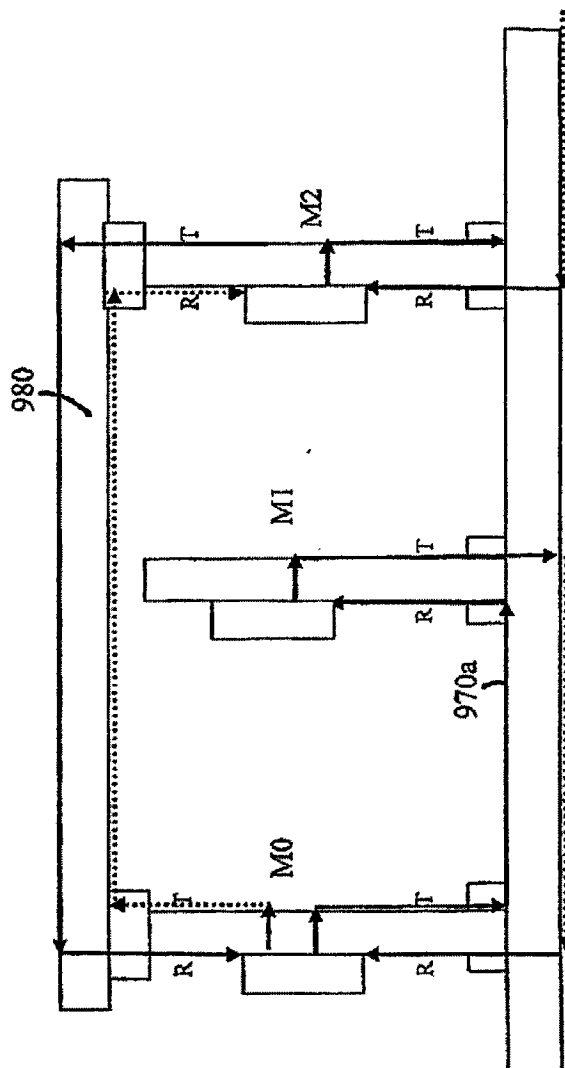


图 75b

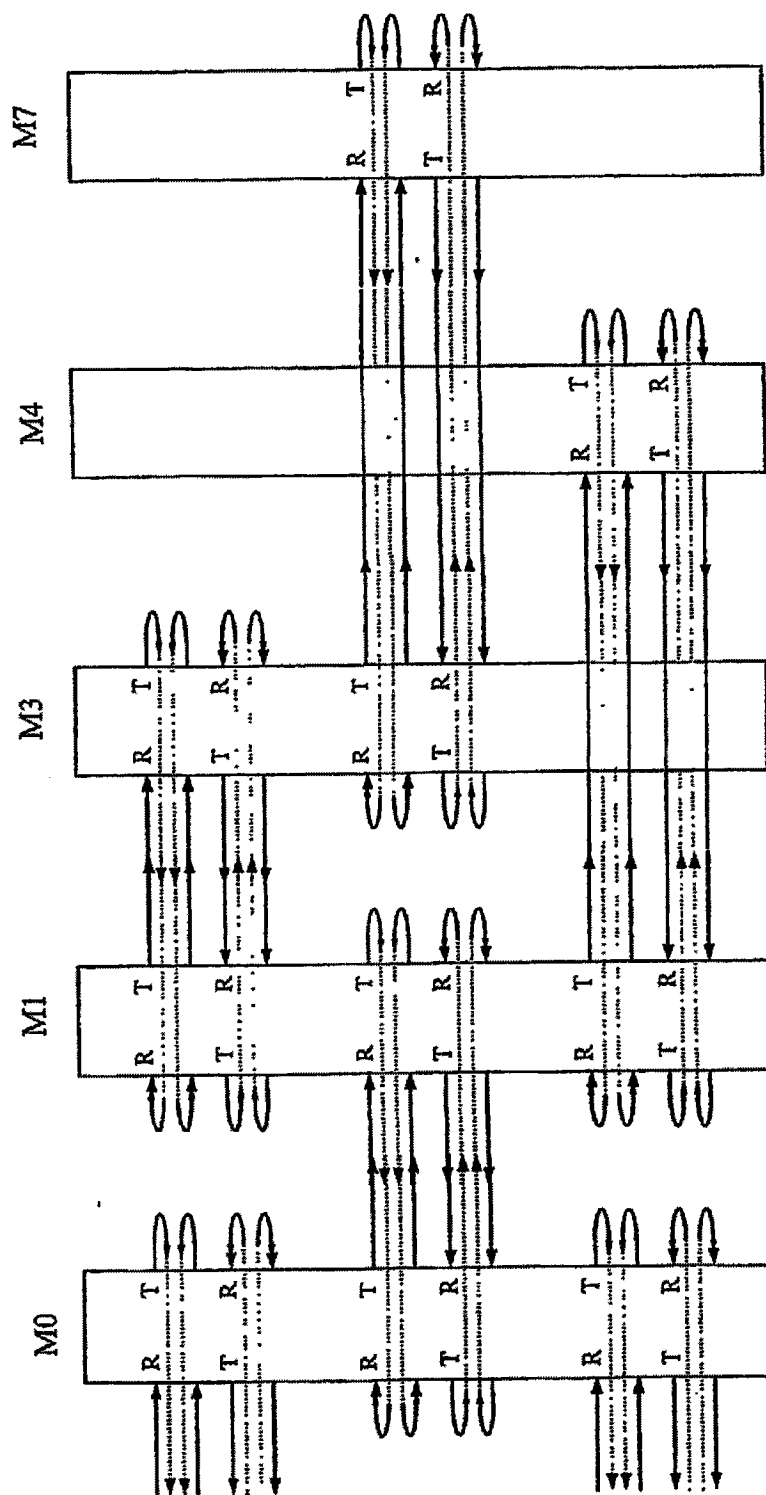
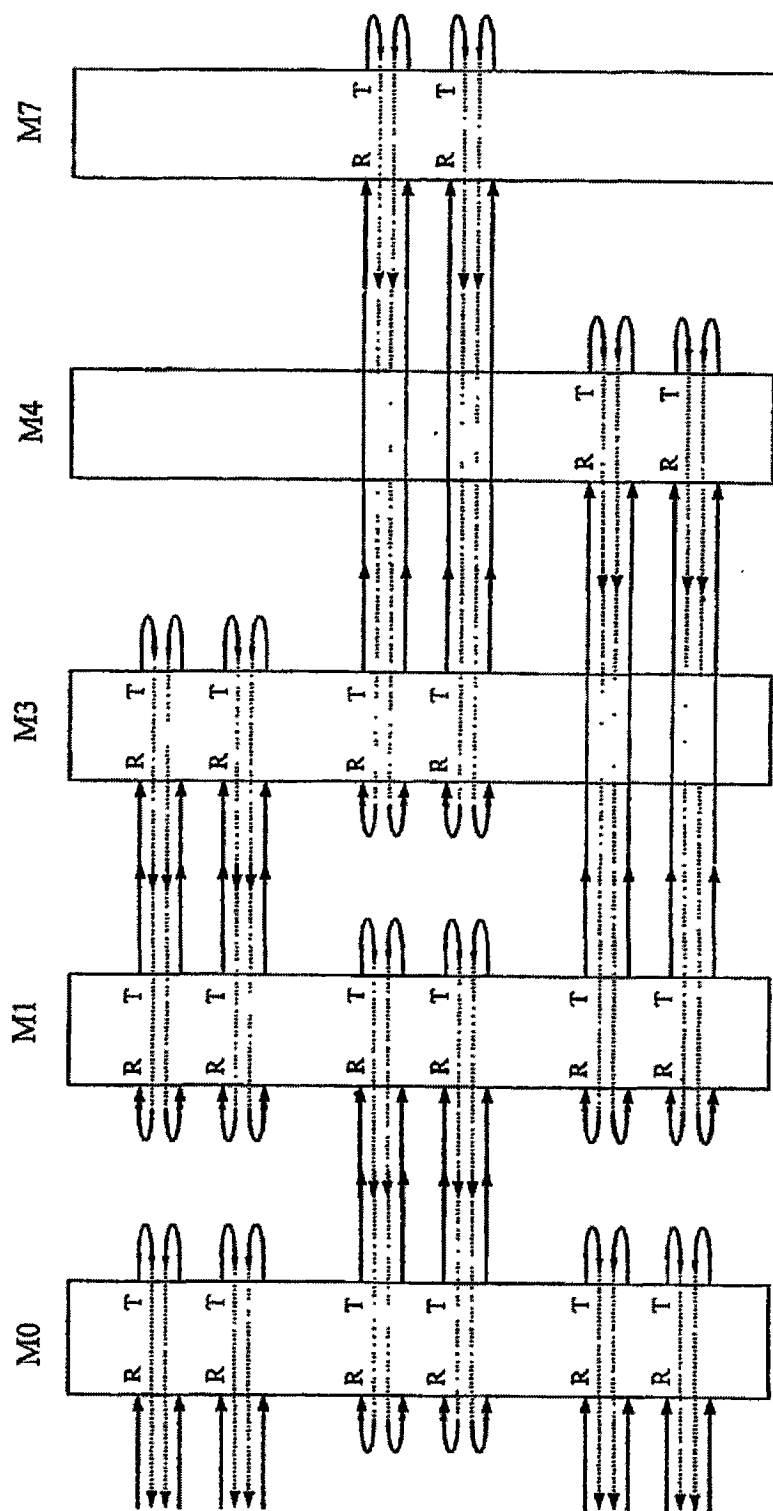


图 76



77

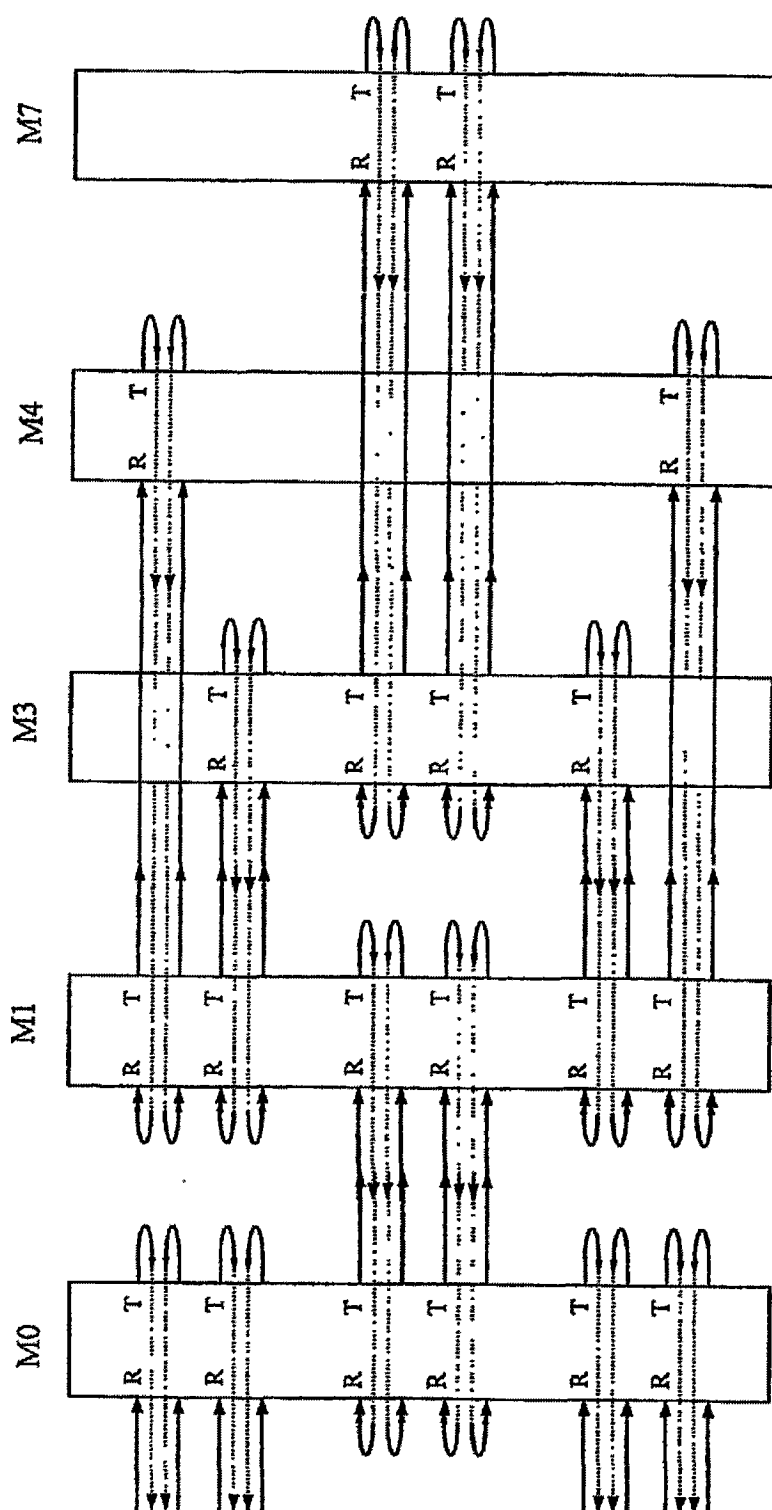


图 78

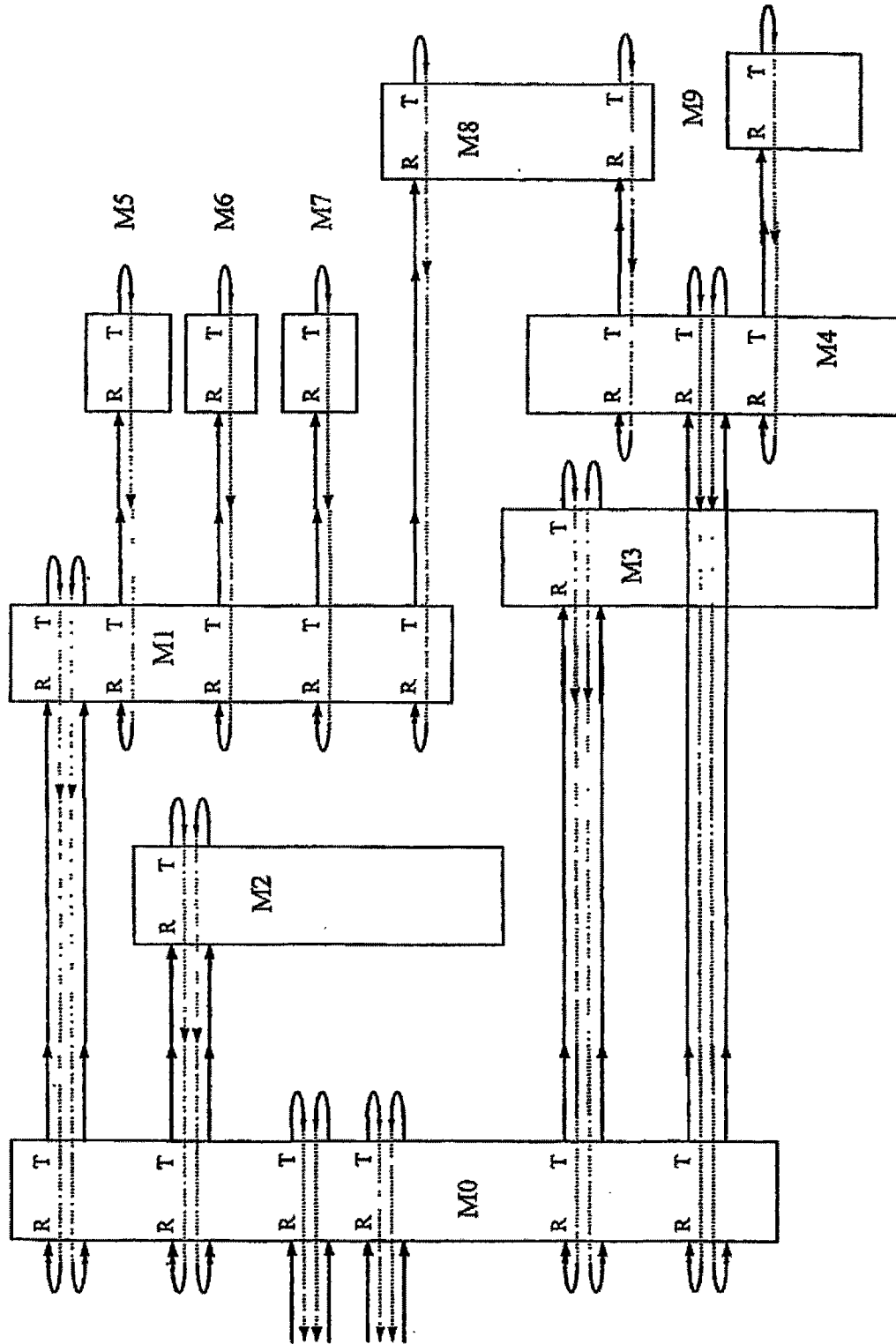


图 79

图 80a

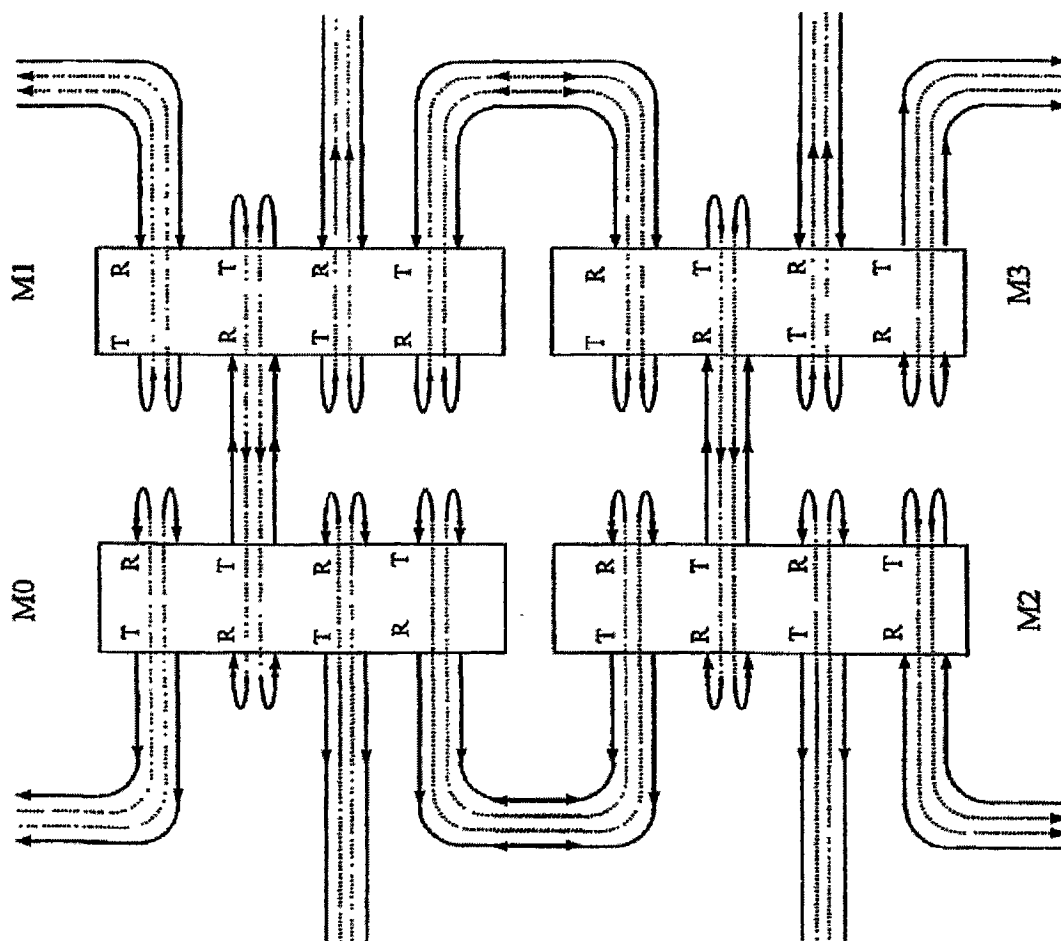


图 80b

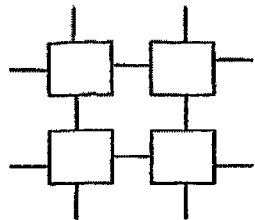
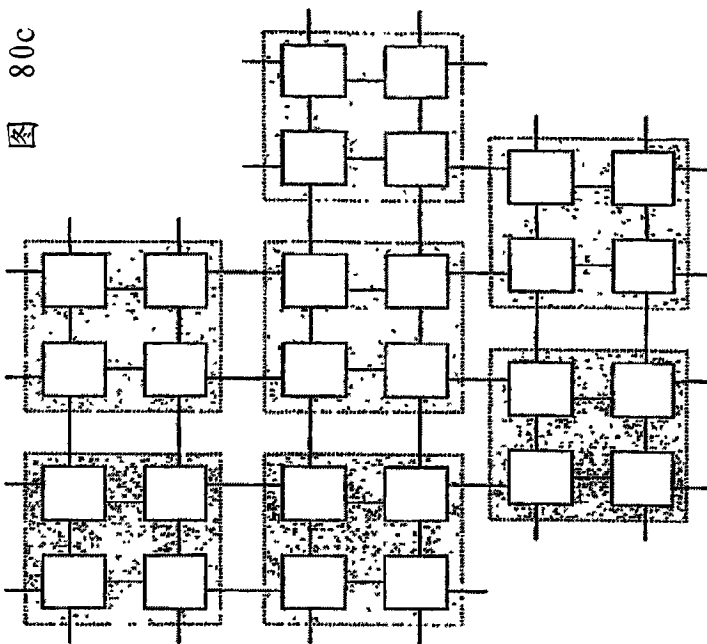


图 80c



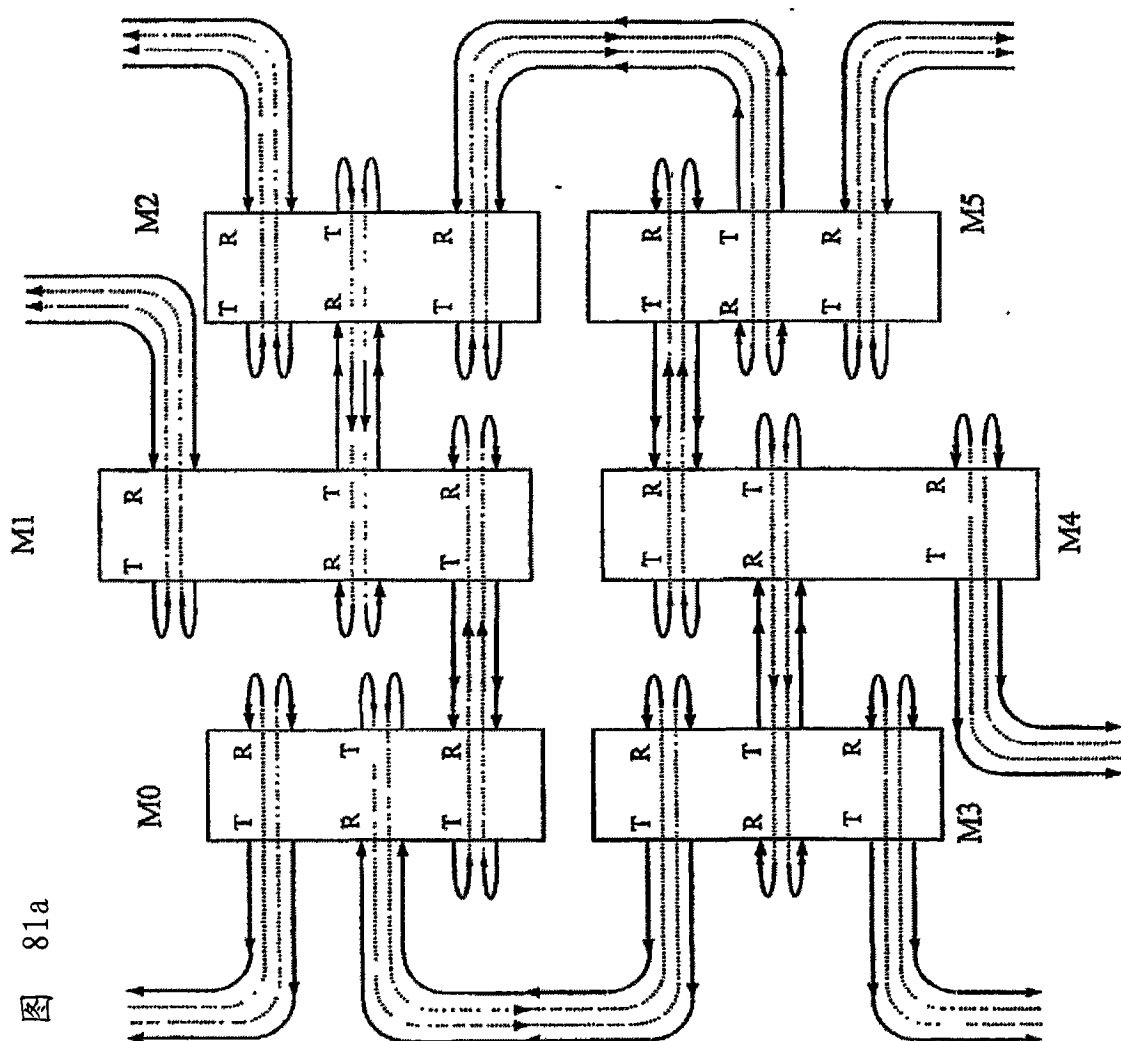
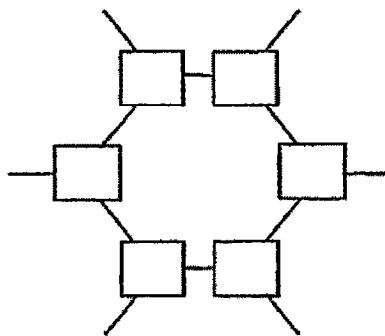


图 81b



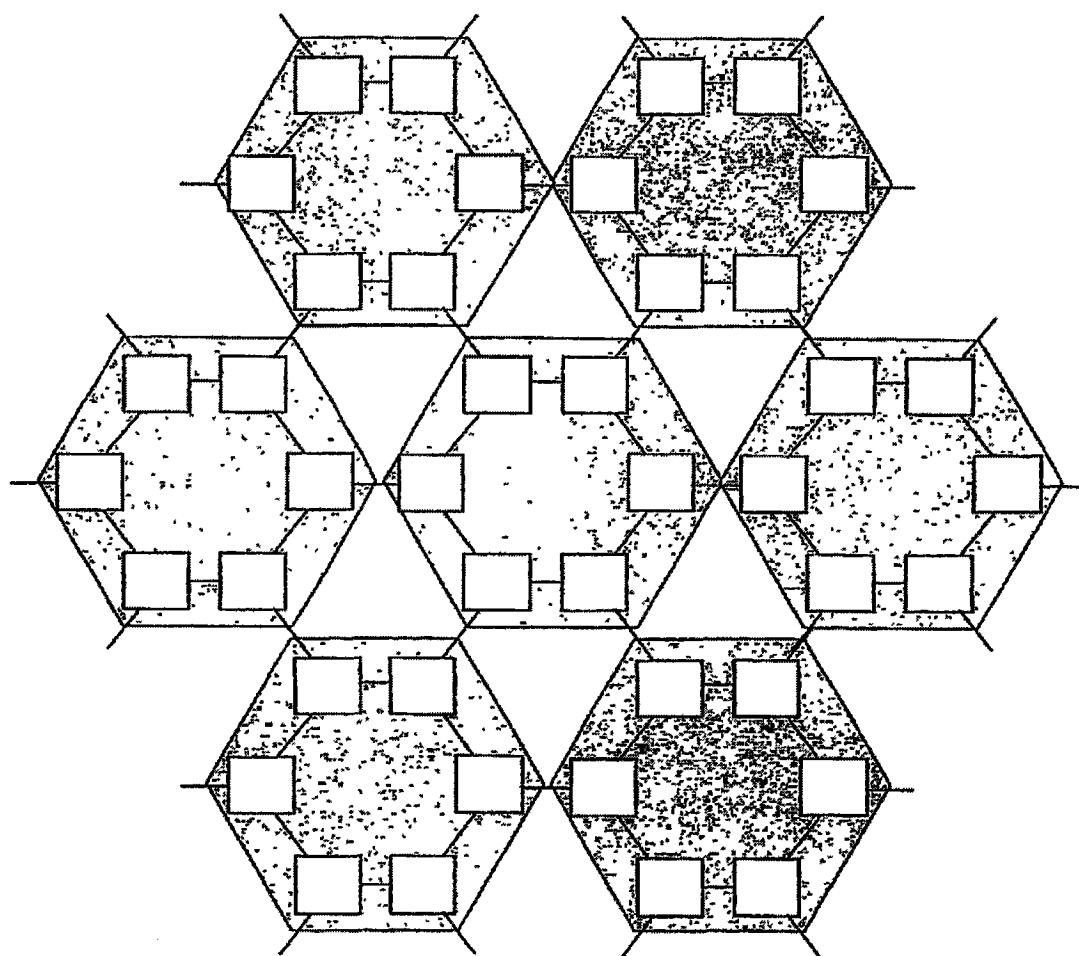


图 82

图 83a

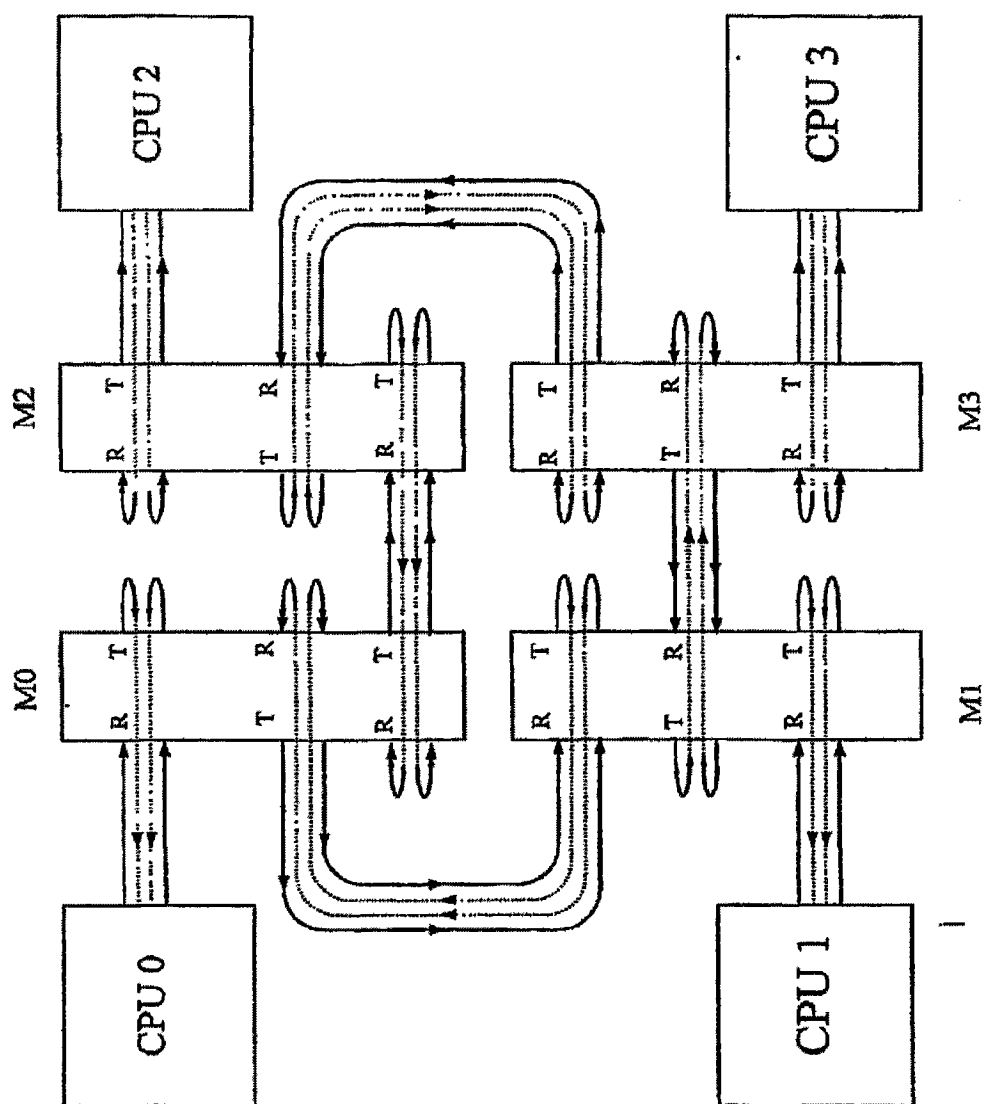


图 83b

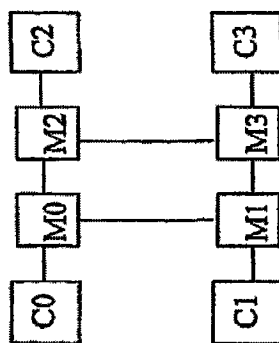


图 84a

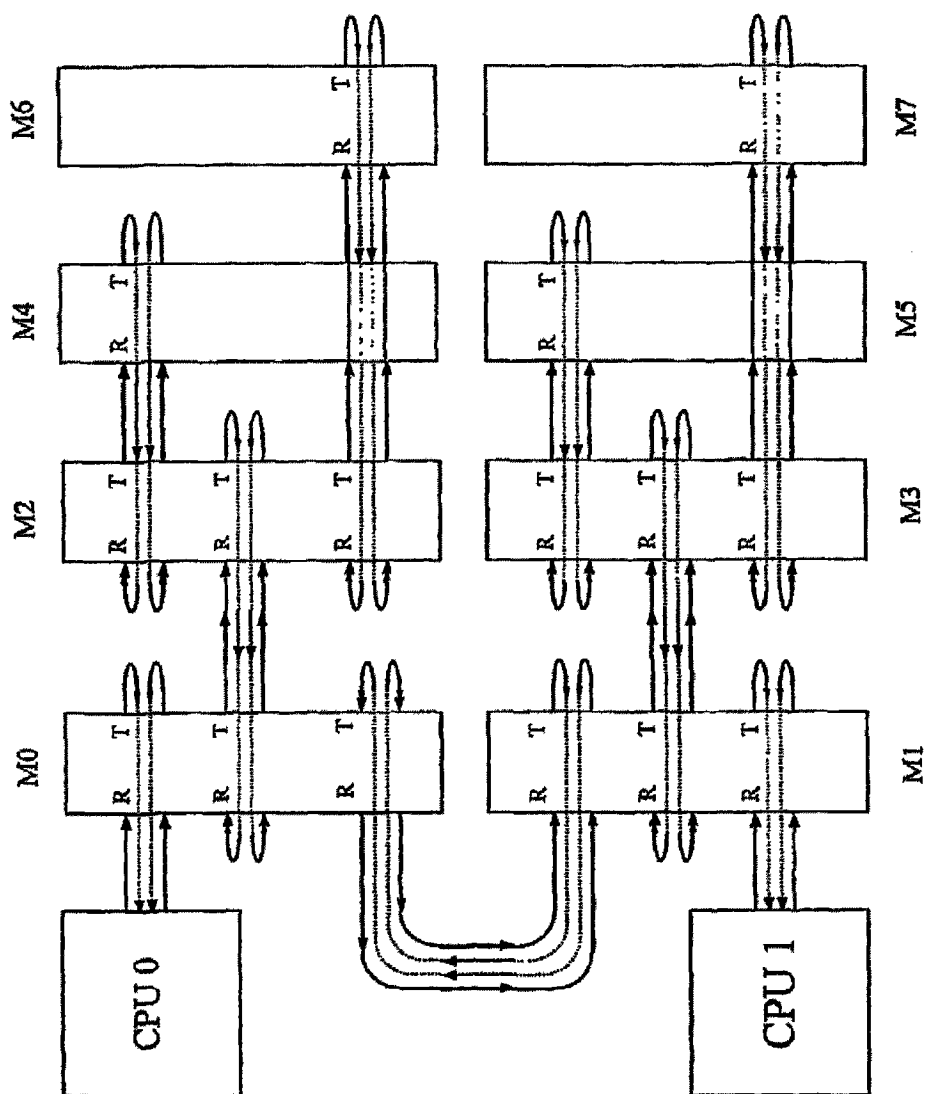


图 84b

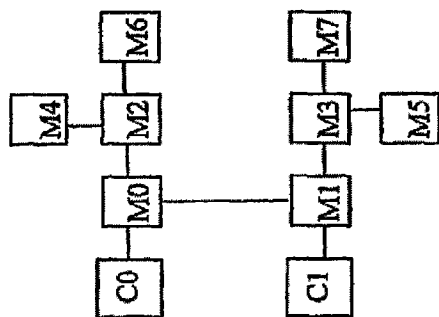


图 85b

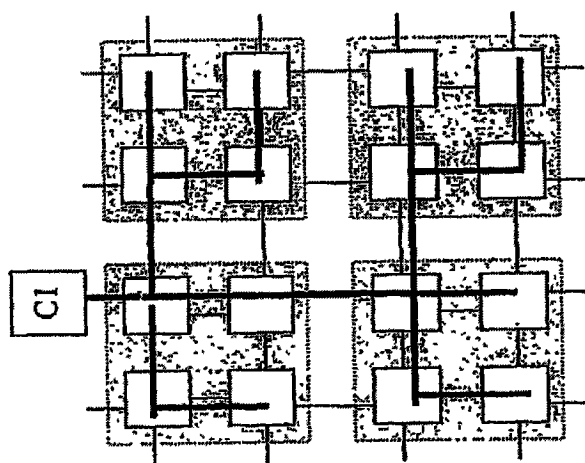
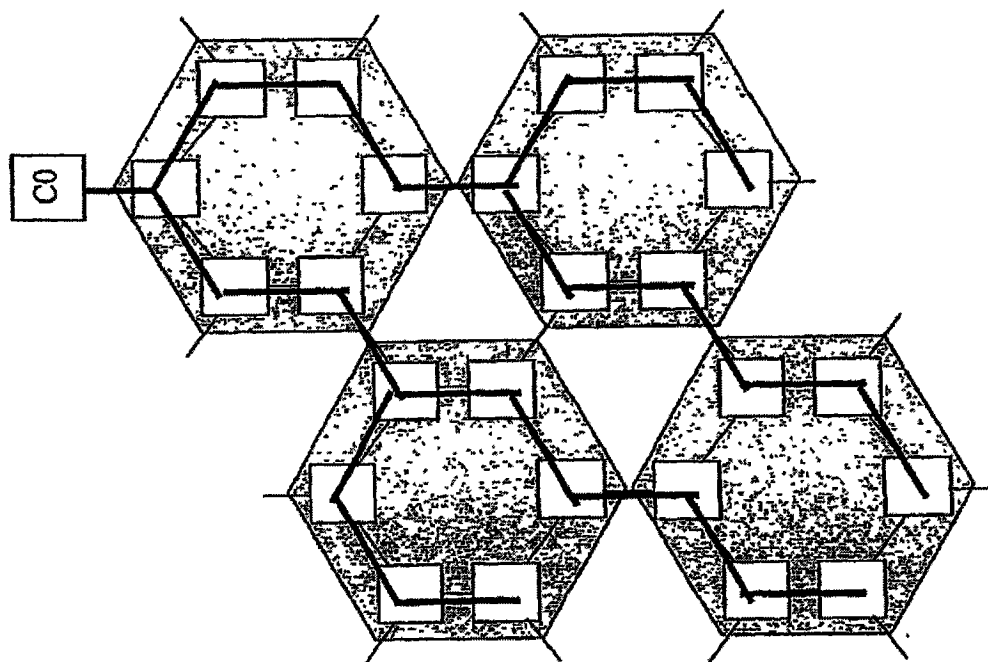
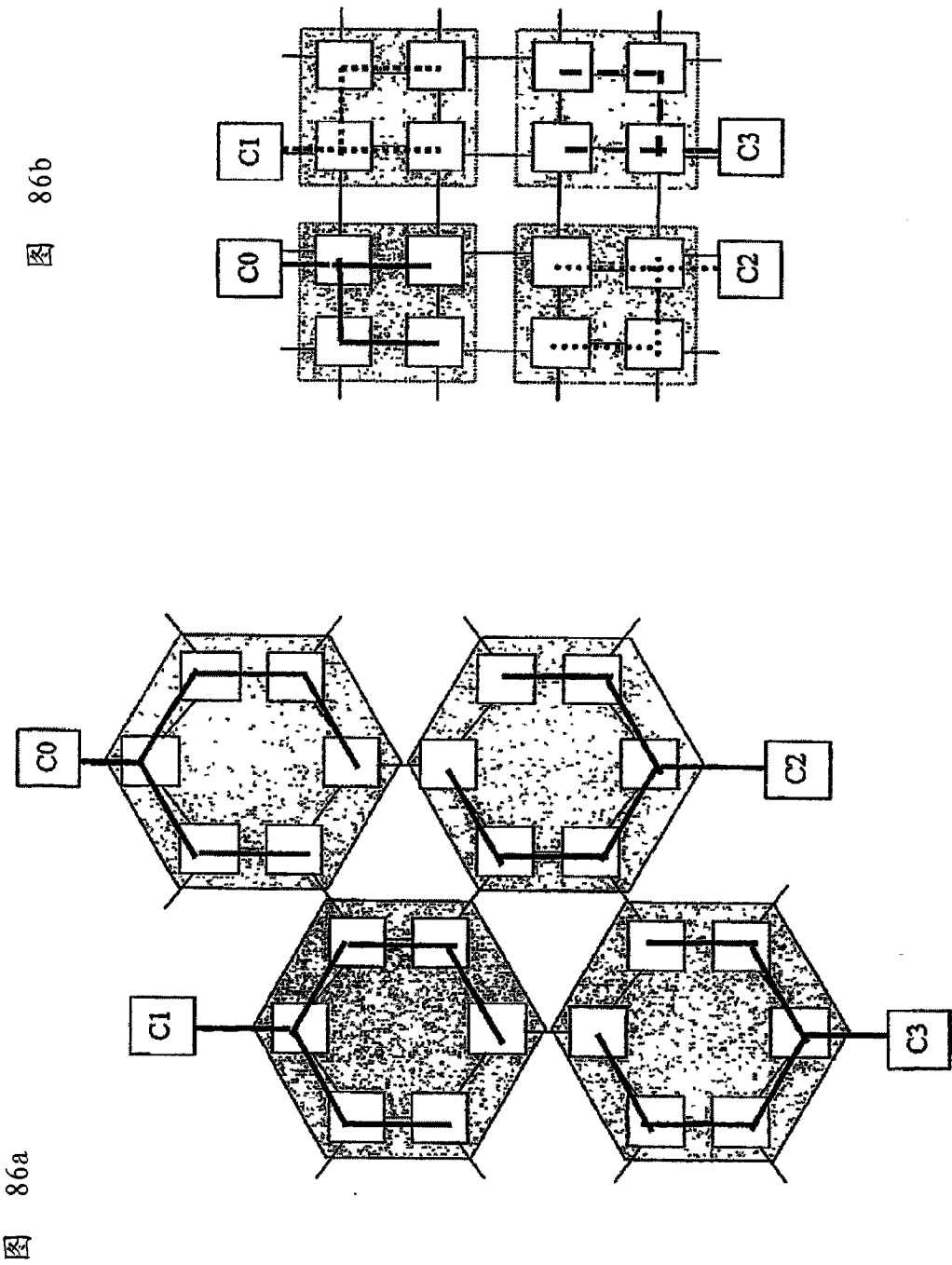


图 85a





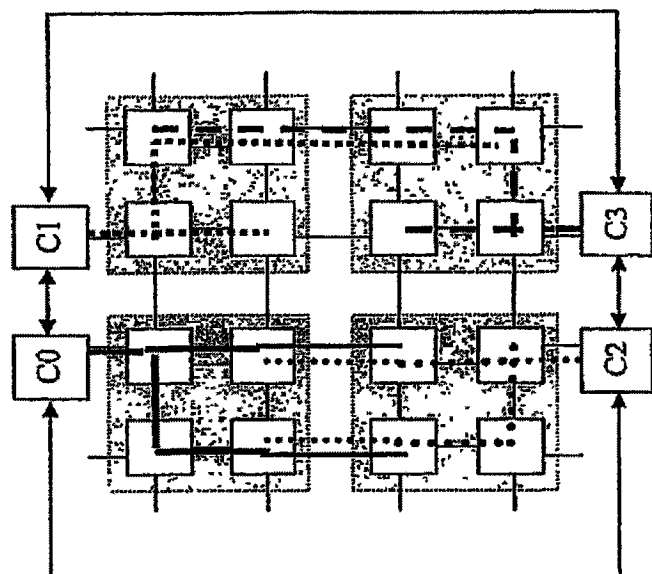


图 87b

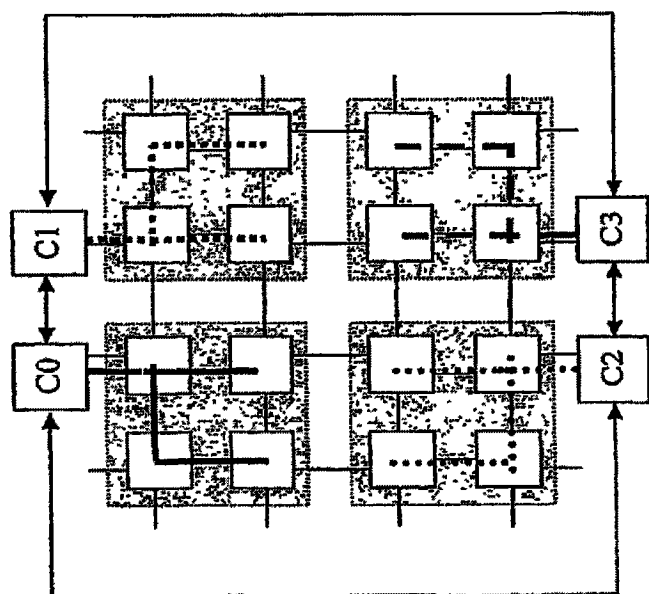
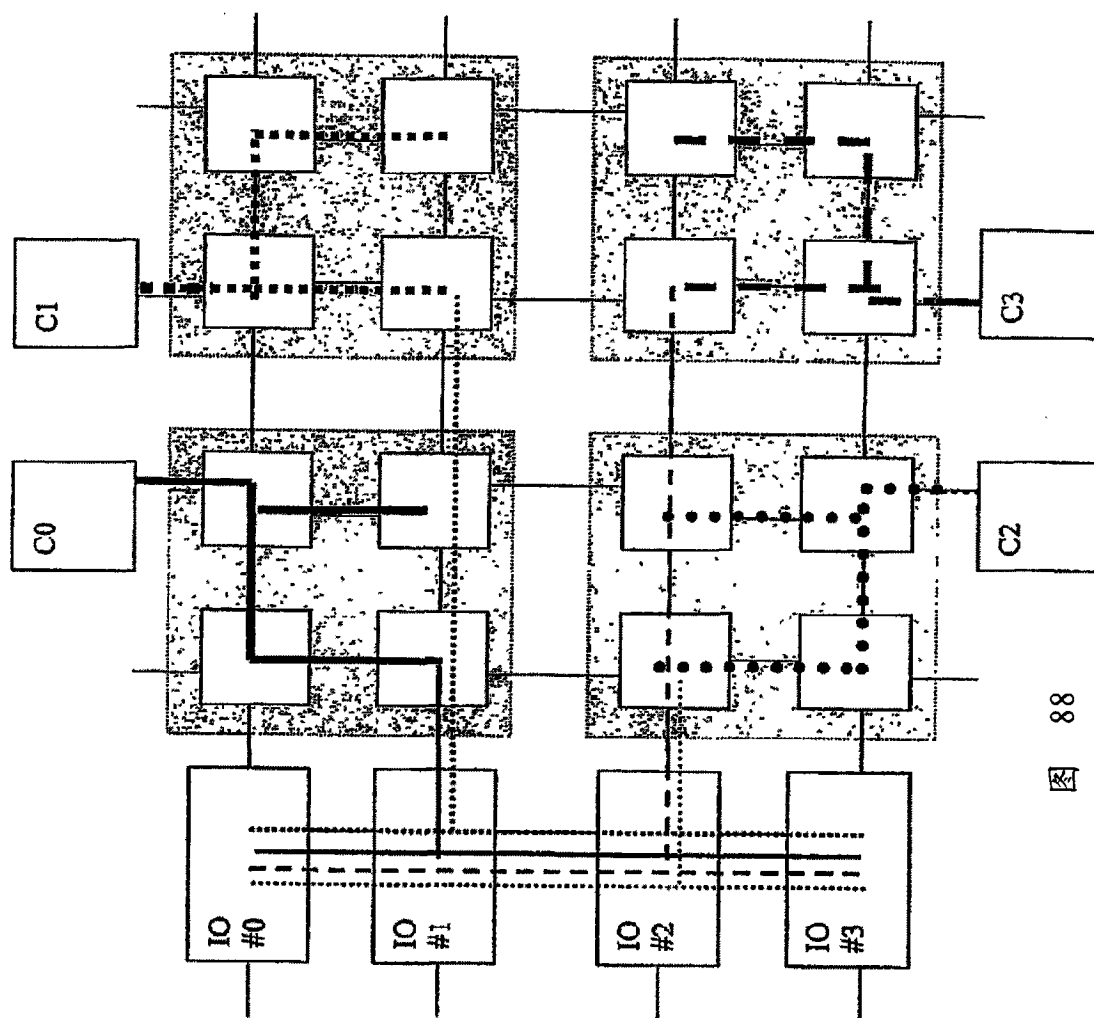


图 87a



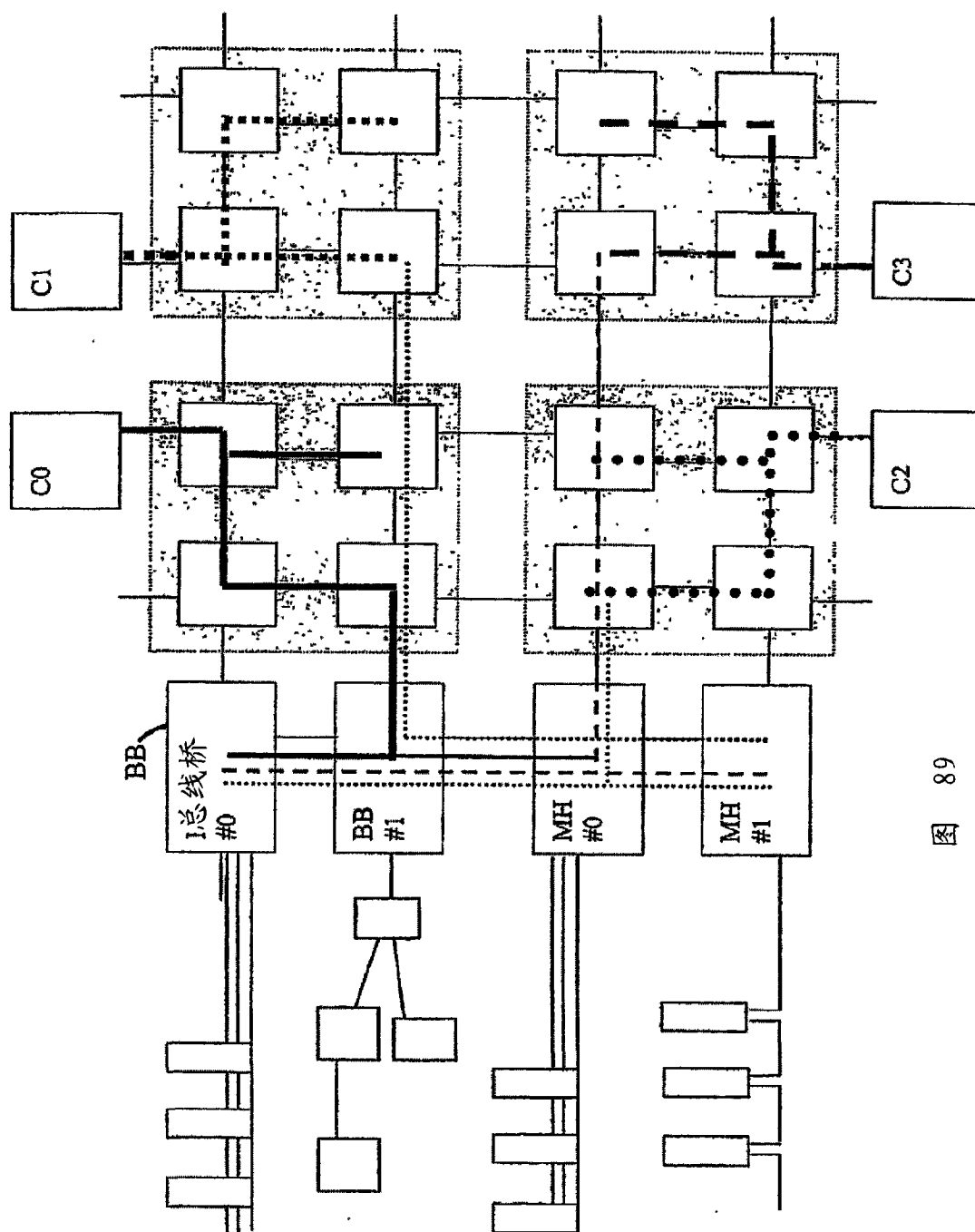


图 89

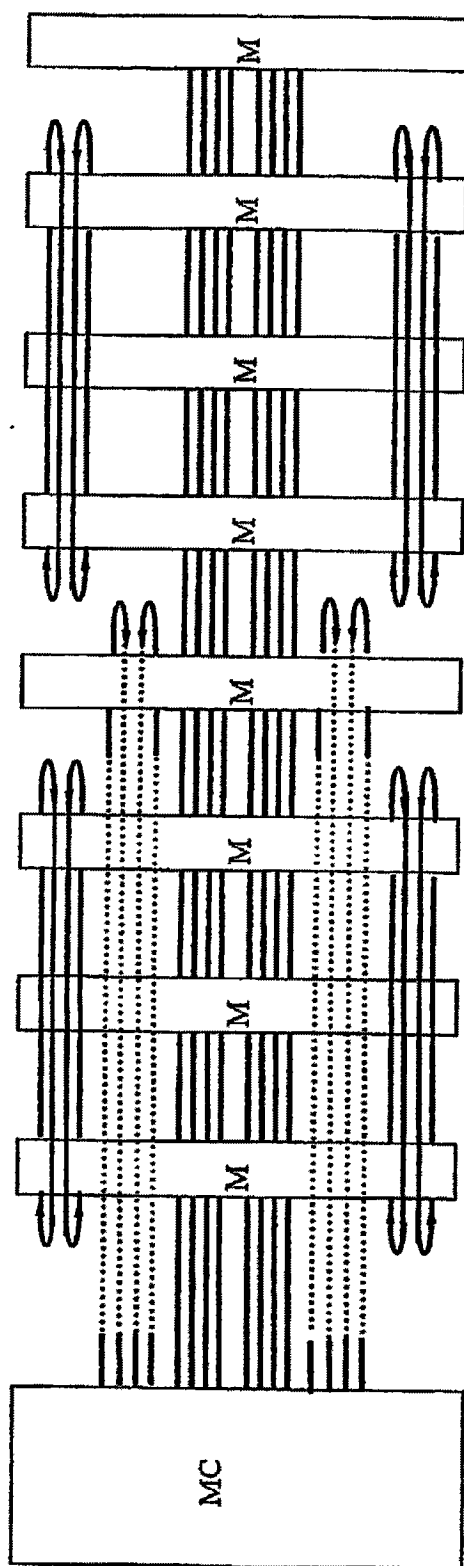


图 90

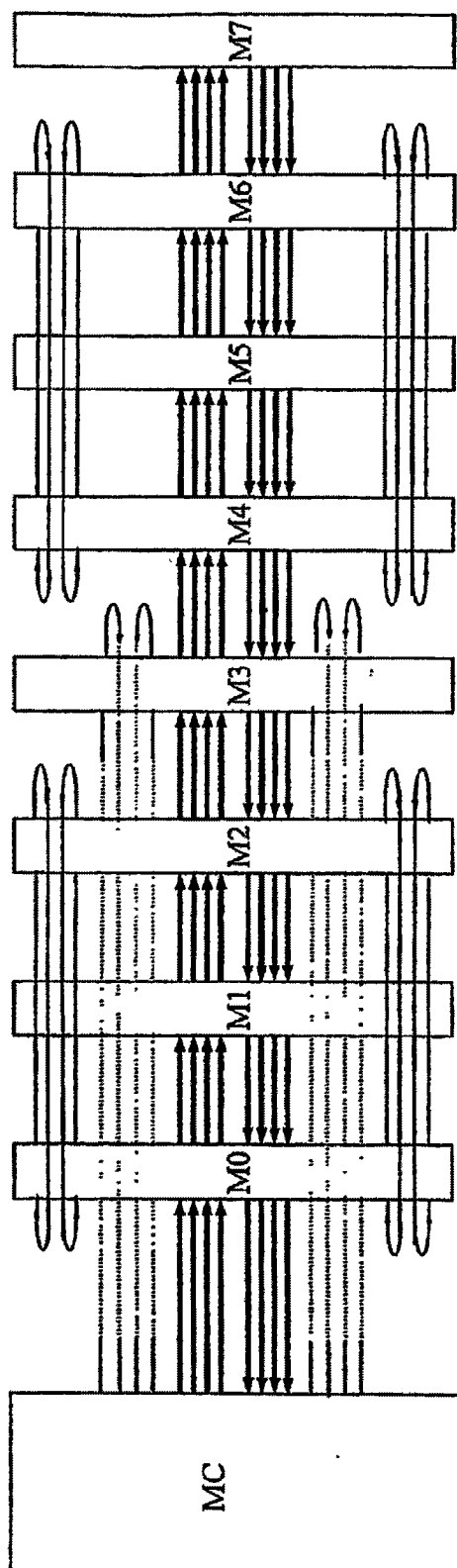
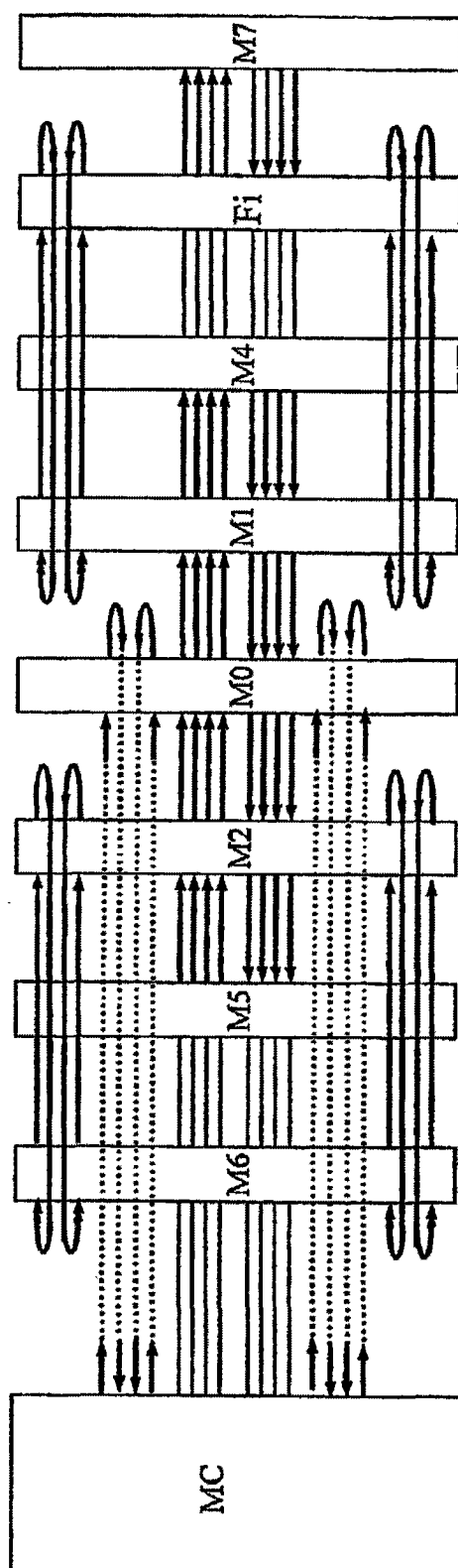


图 91



92

图 93a

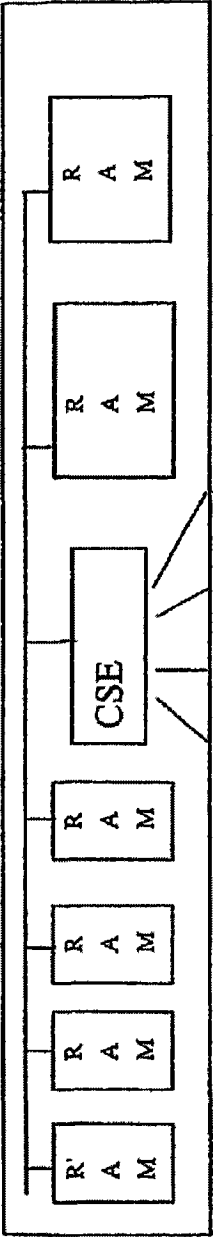
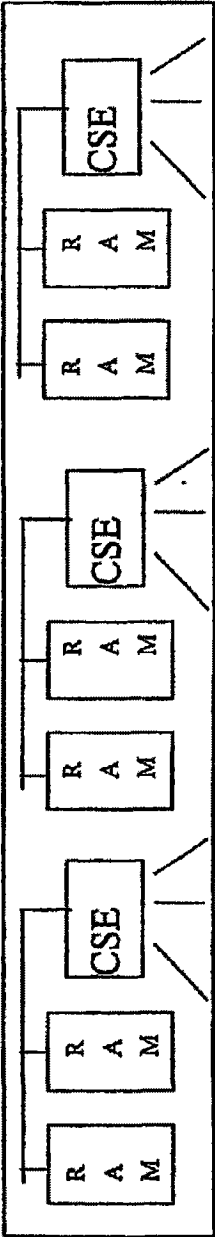


图 93b



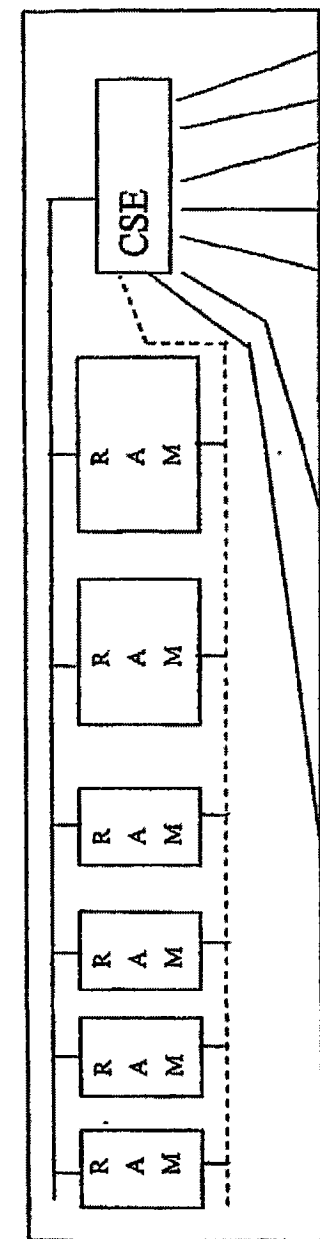


图 94a

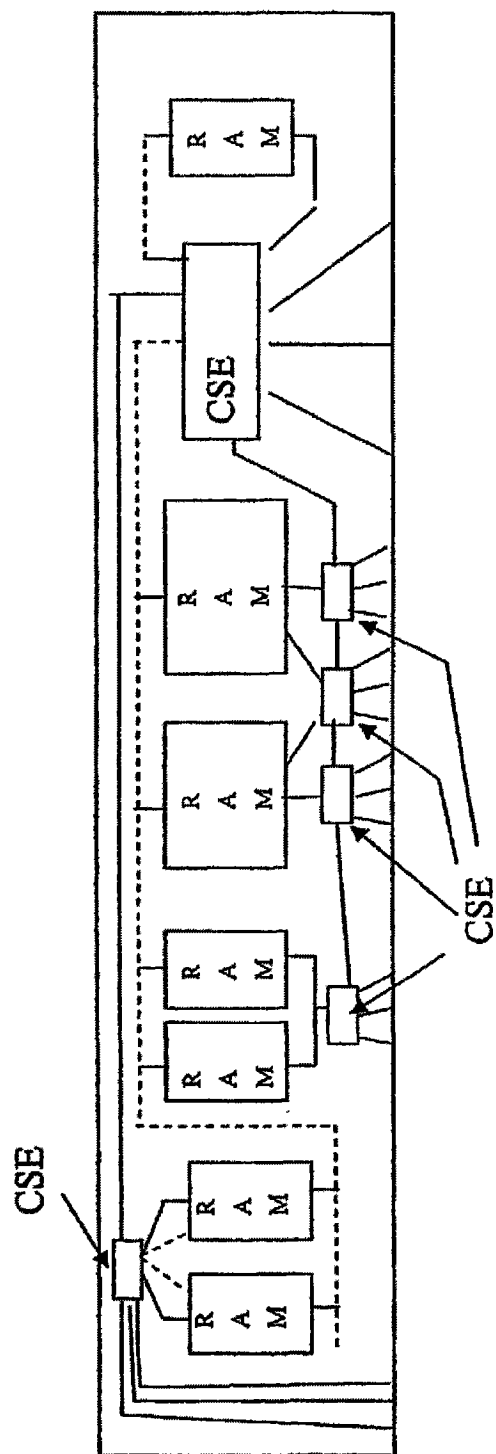


图 94b

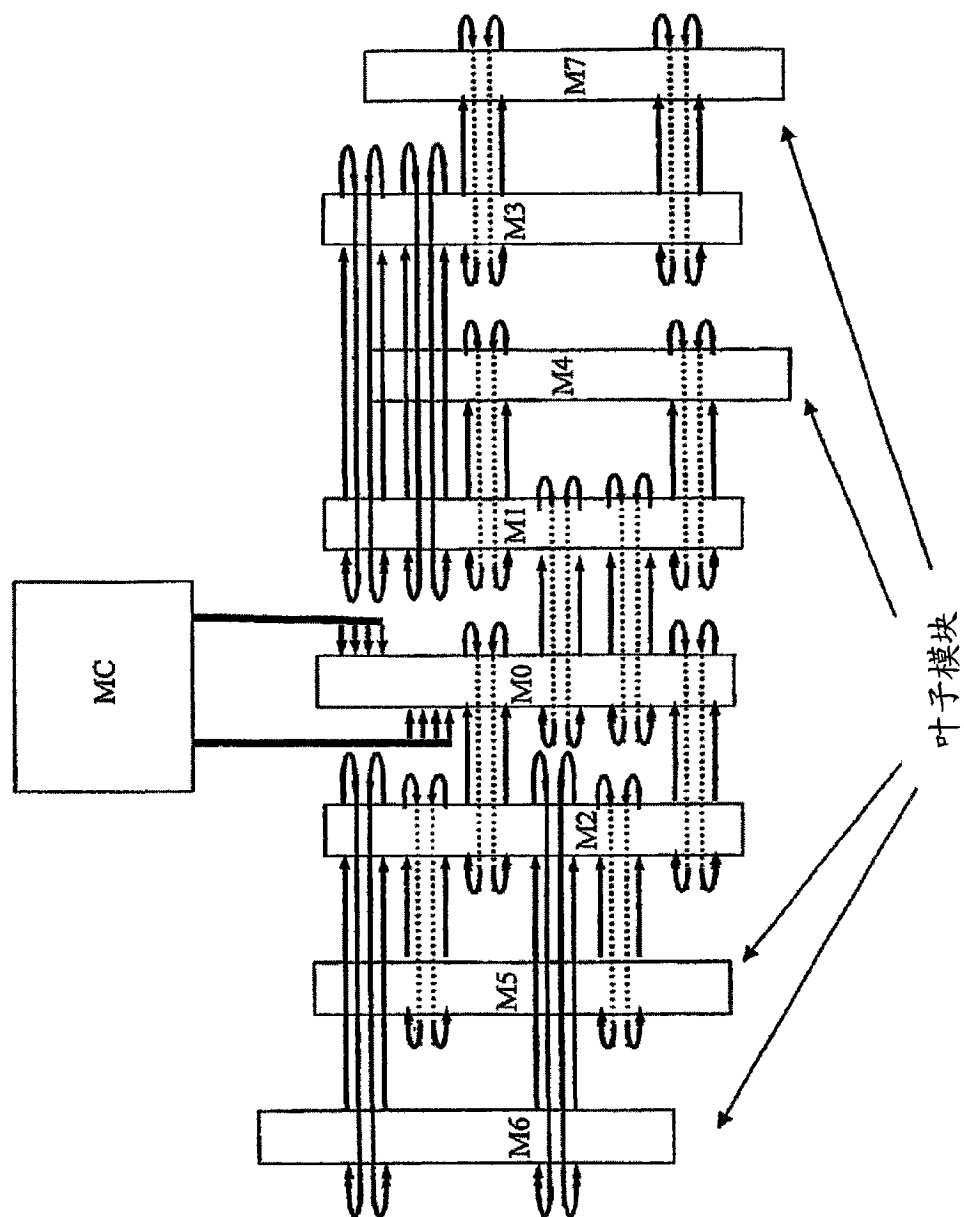


图 95

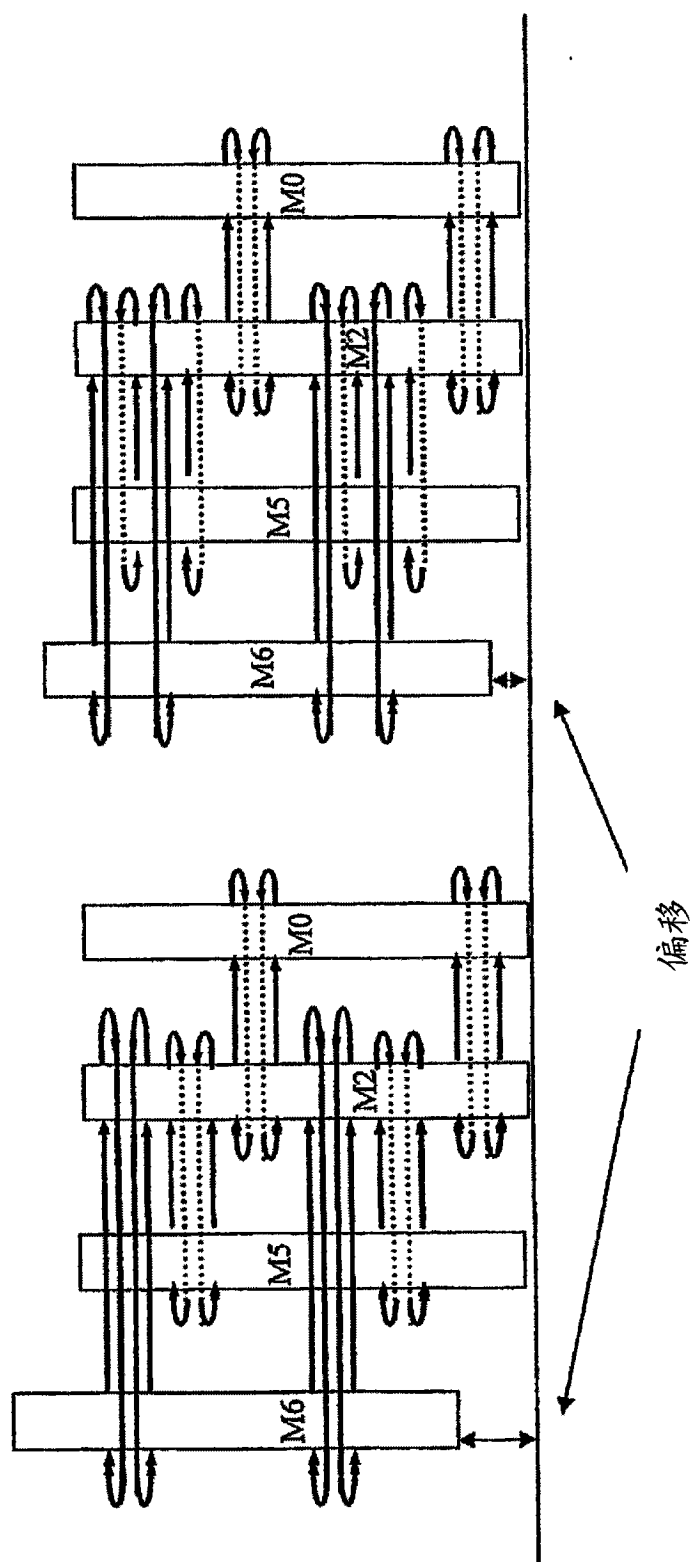


图 96

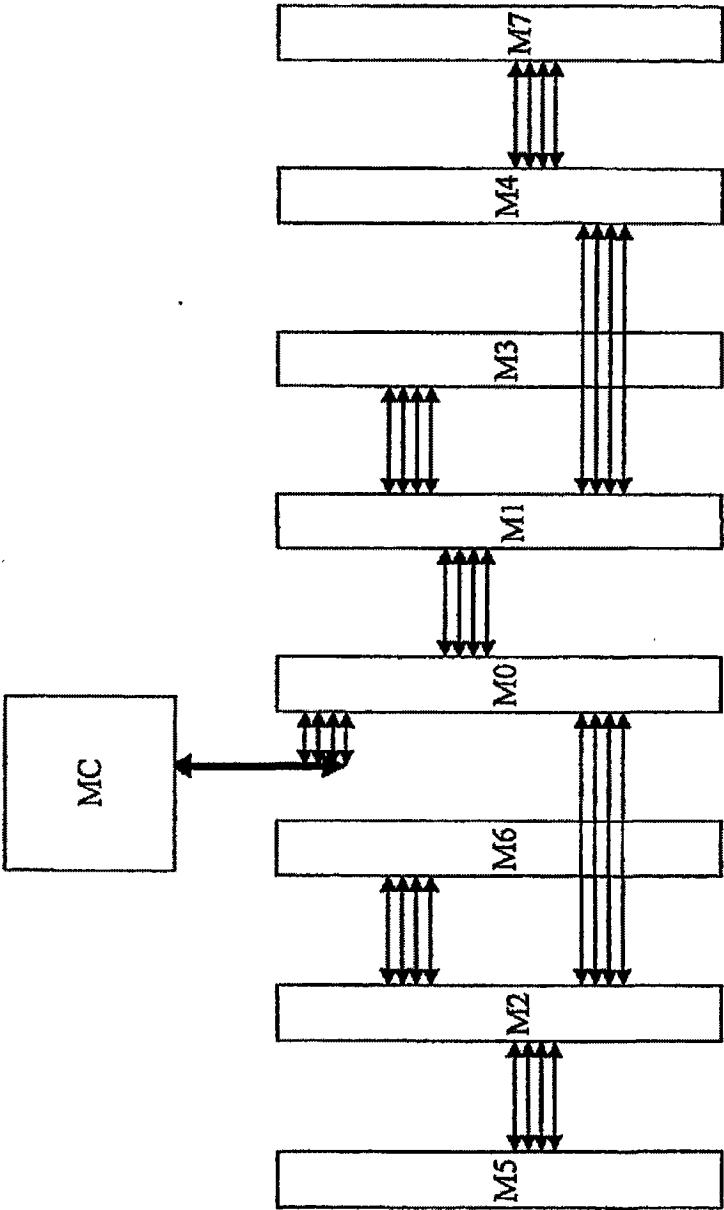


图 97

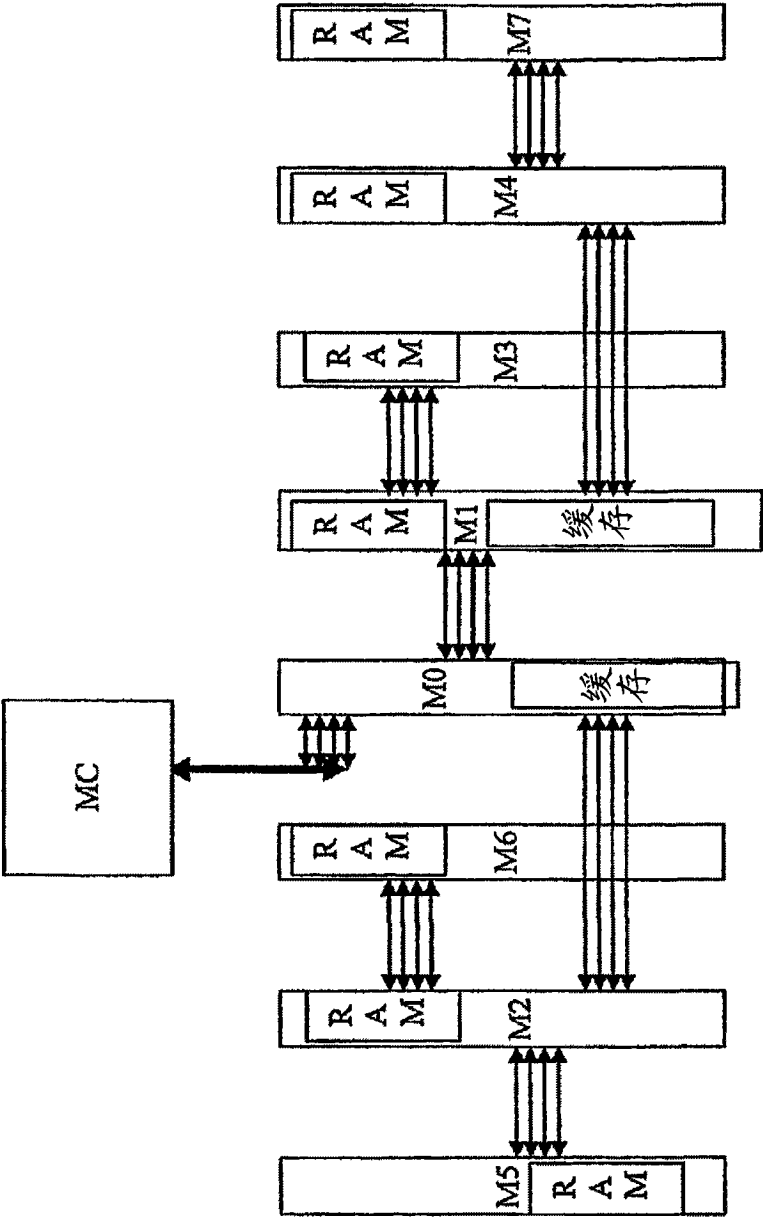


图 98

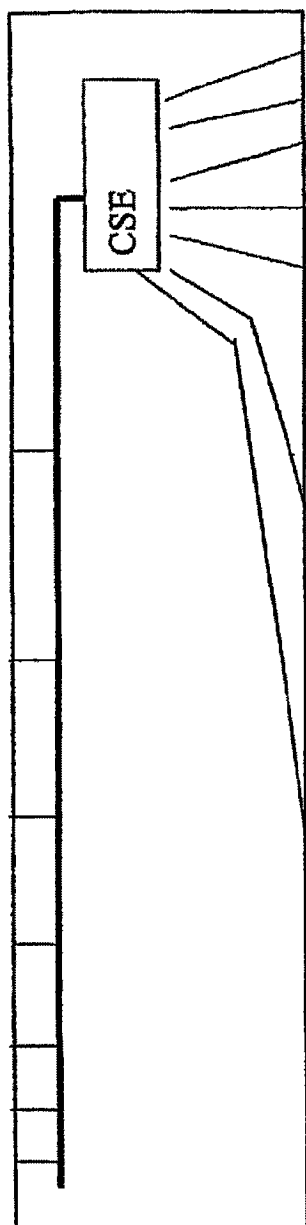


图 99a

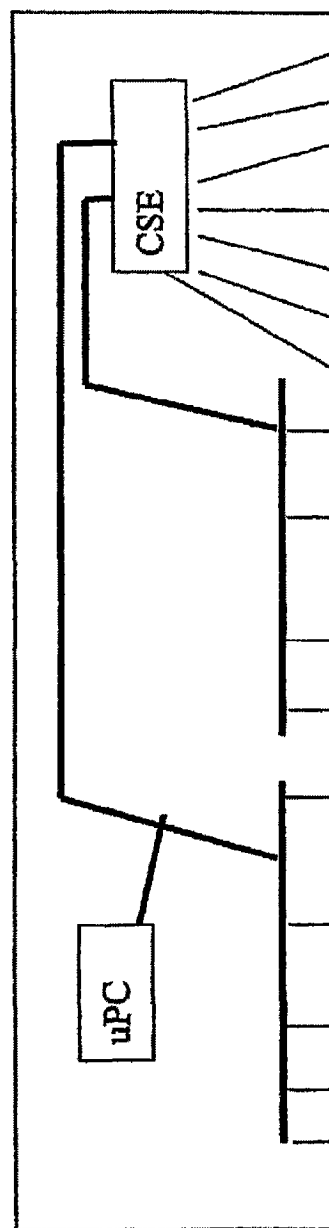


图 99b

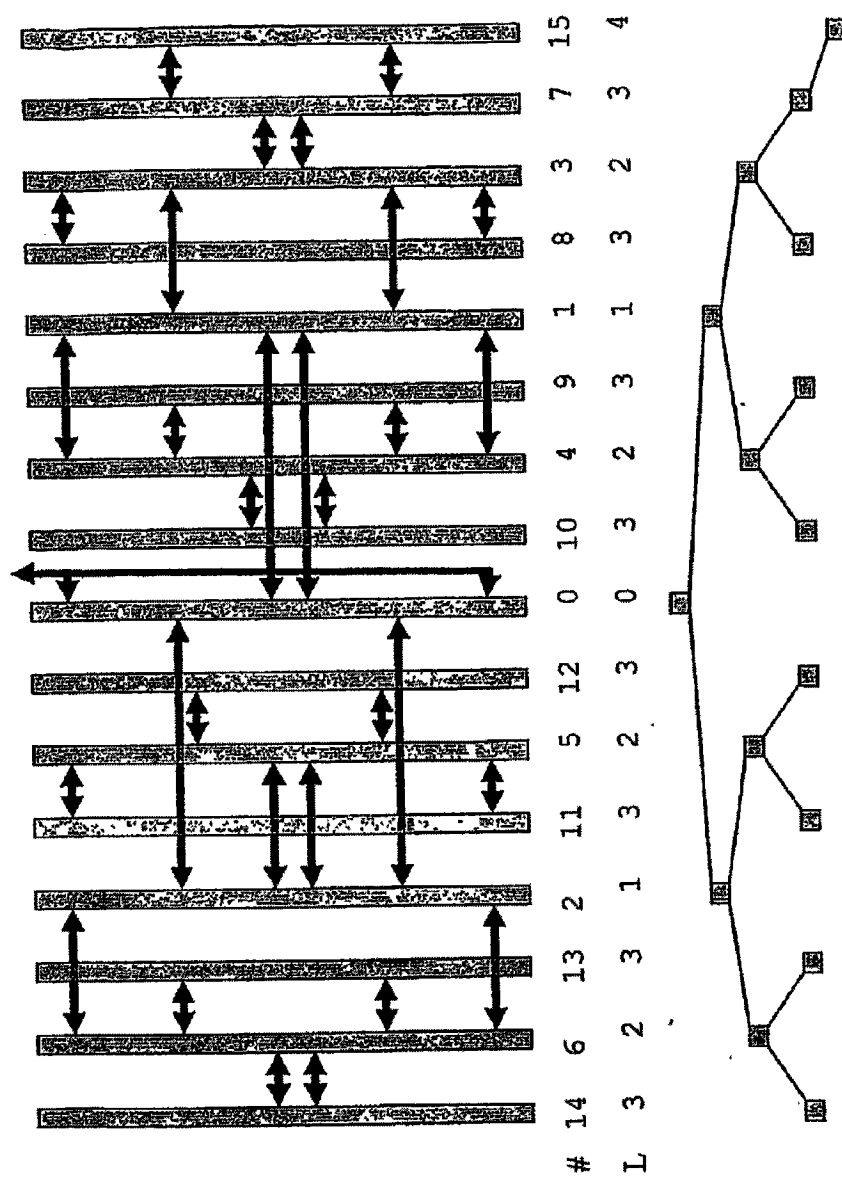


图 100

图 101a

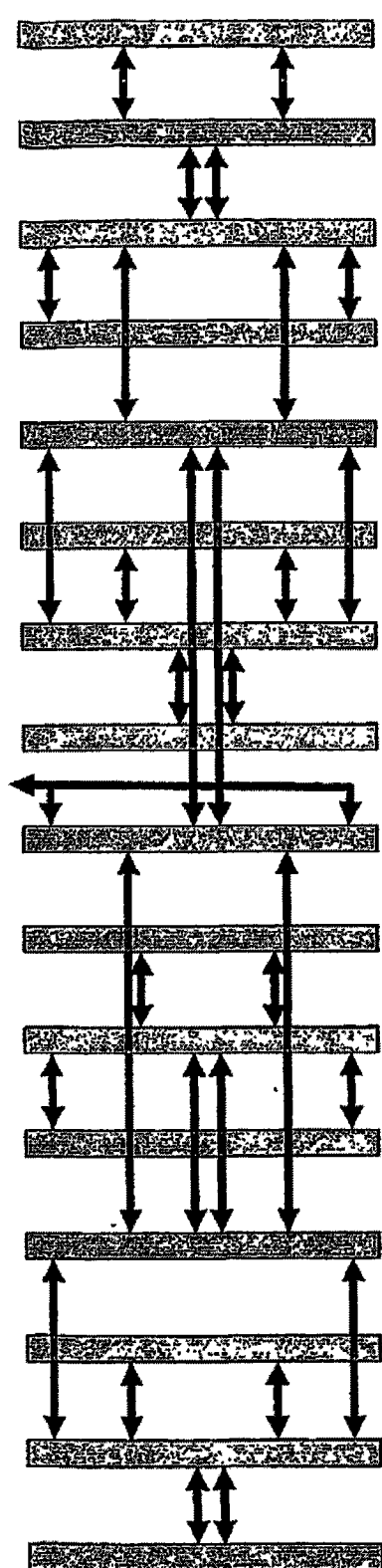
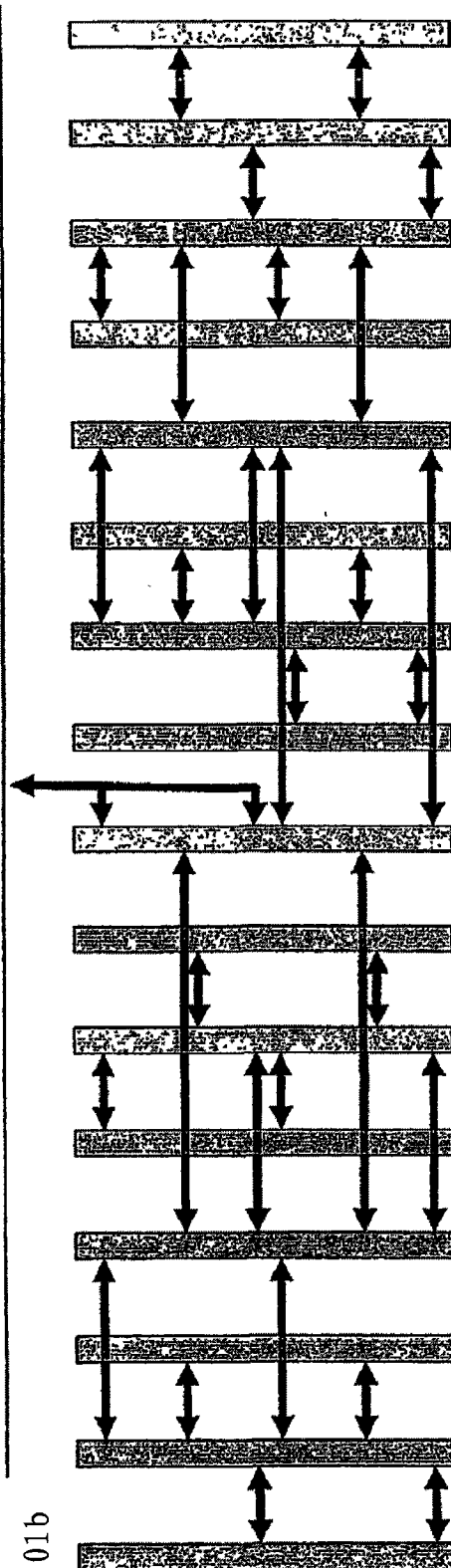


图 101b



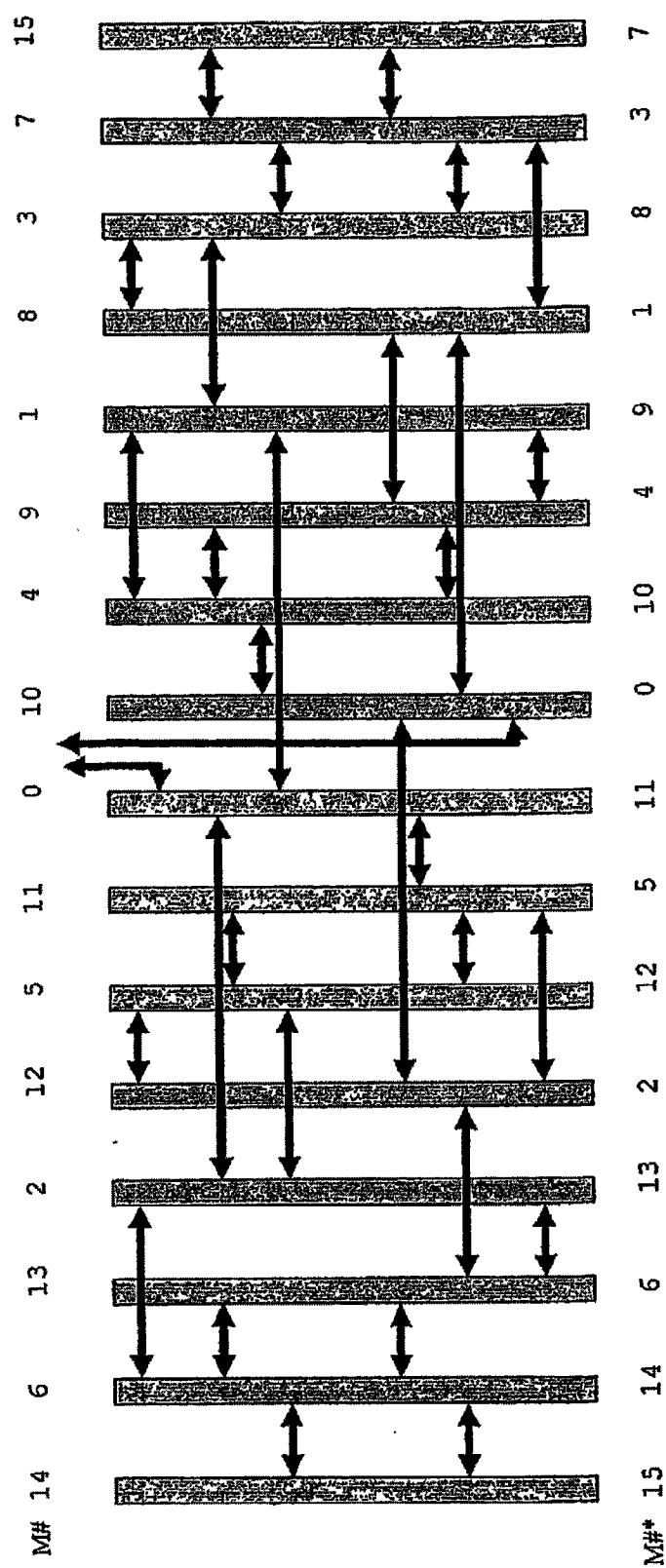
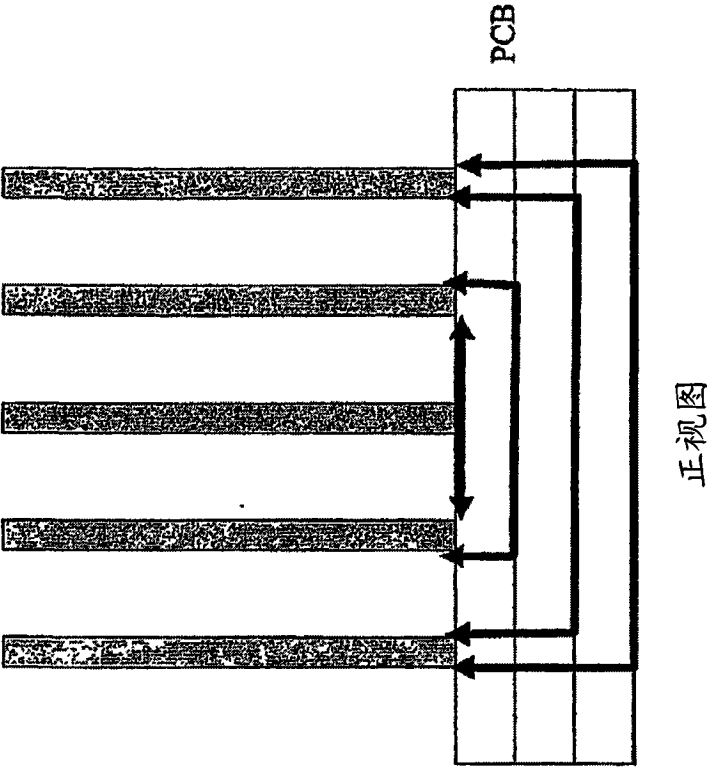
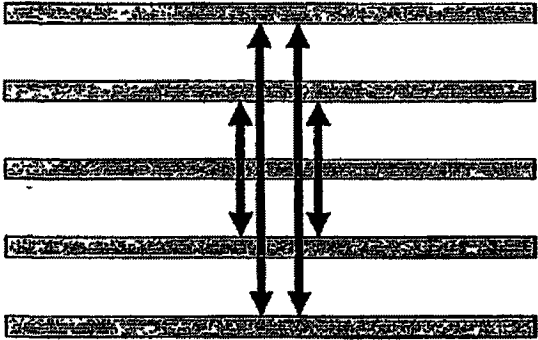


图 102



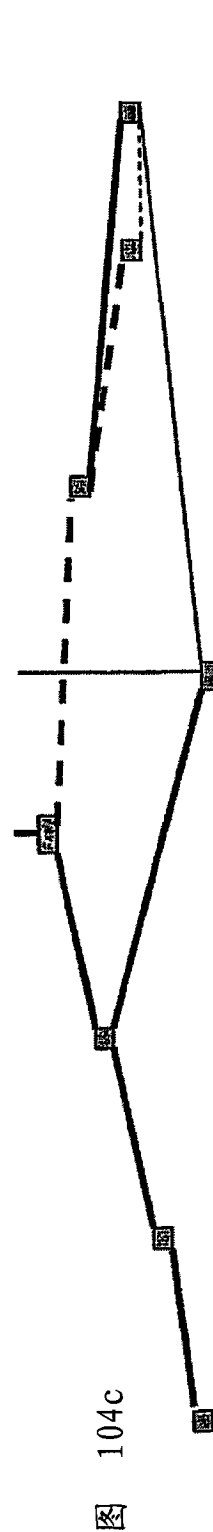
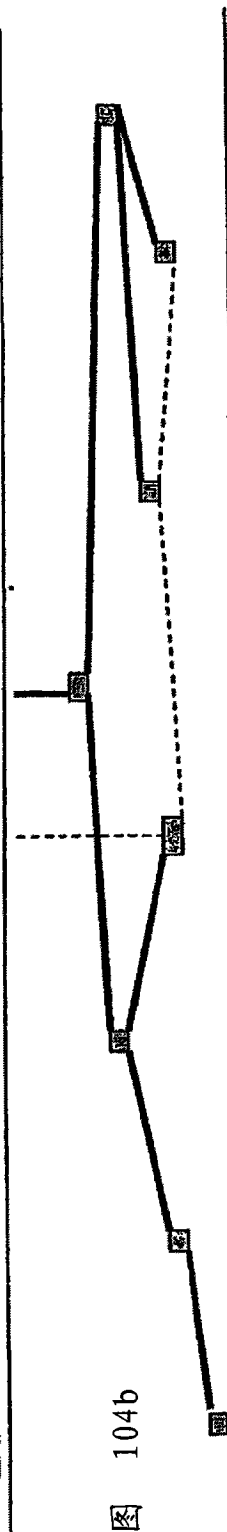
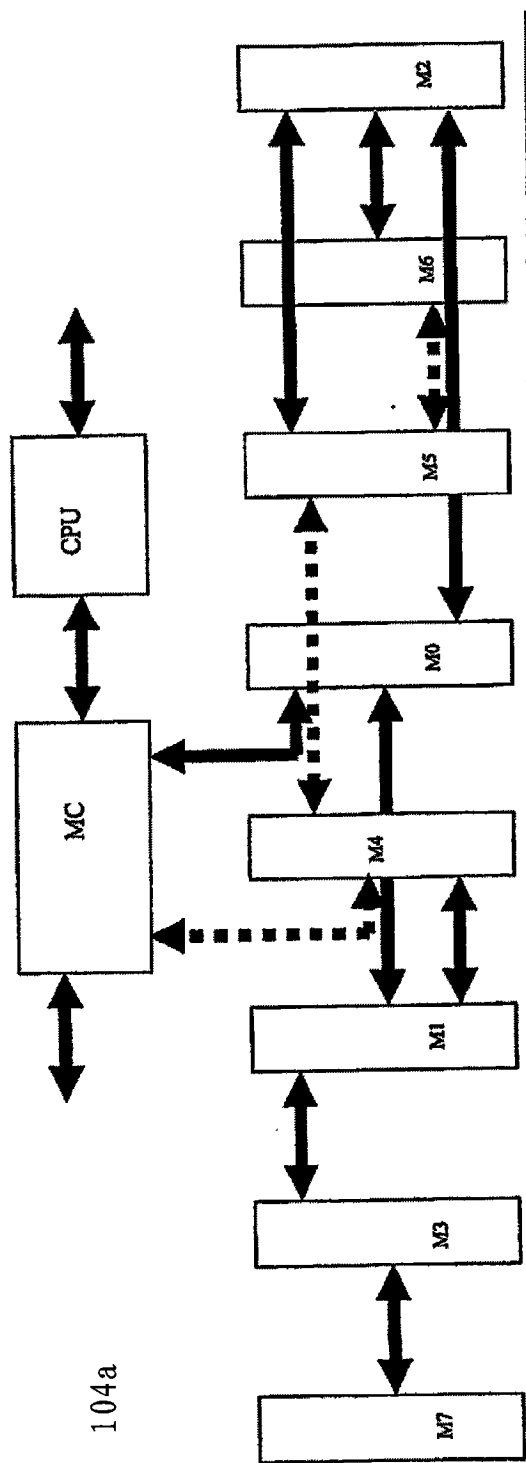
俯视图

图 103a



正视图

图 103b



根 1 根 2

图 105a

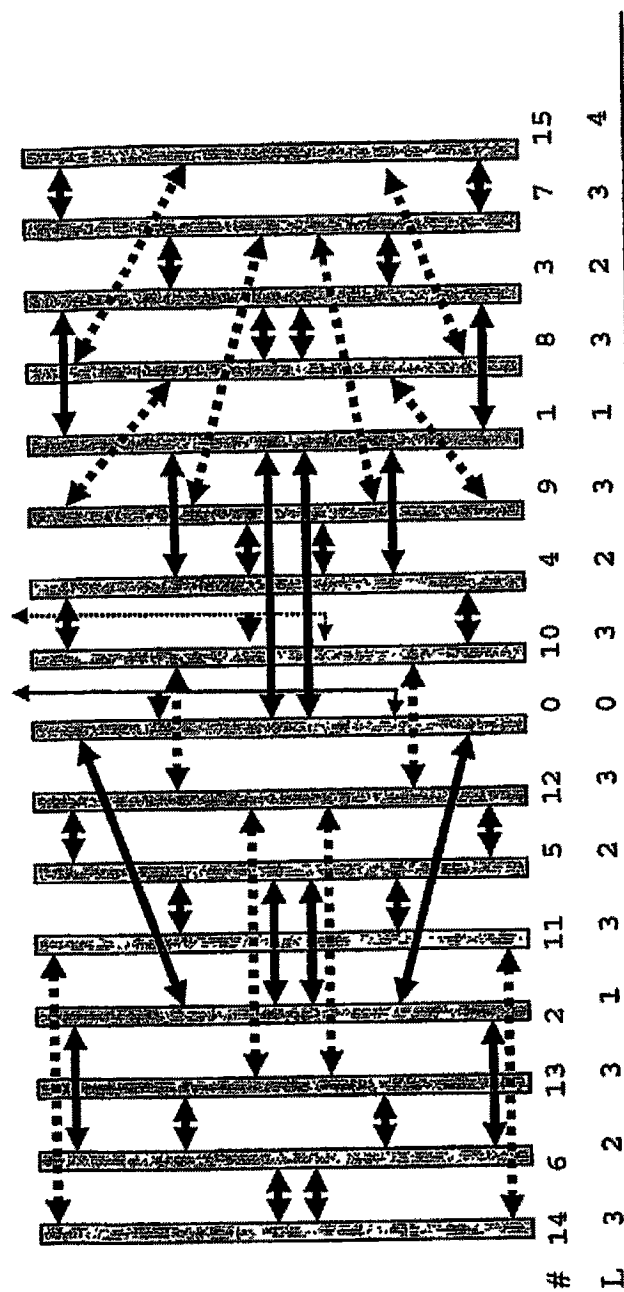
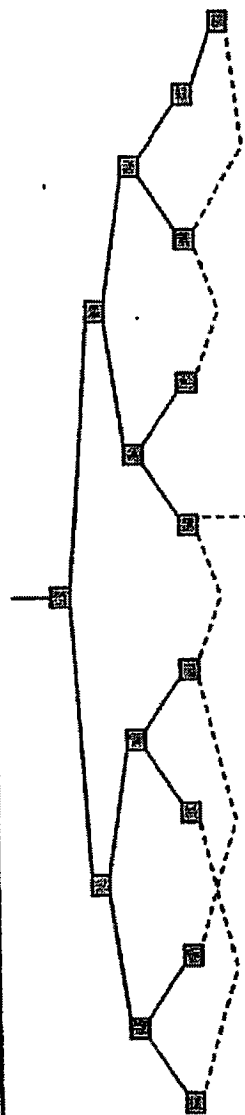
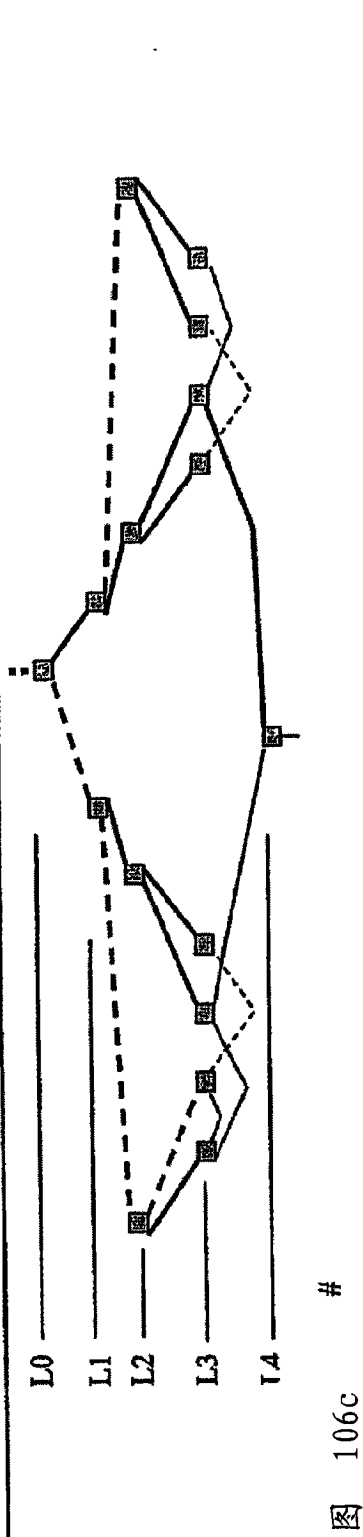
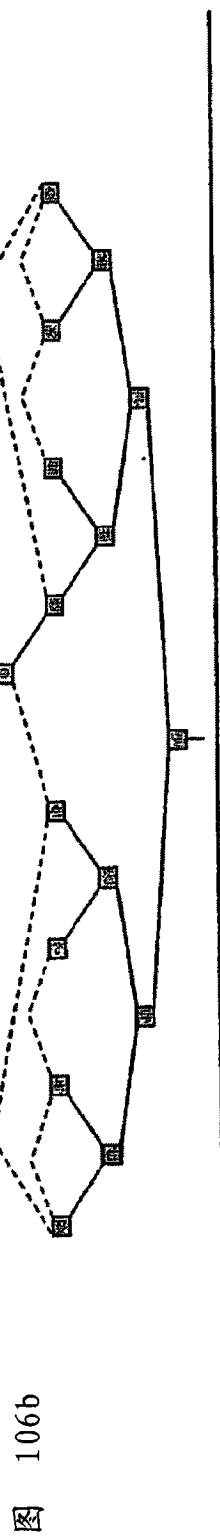
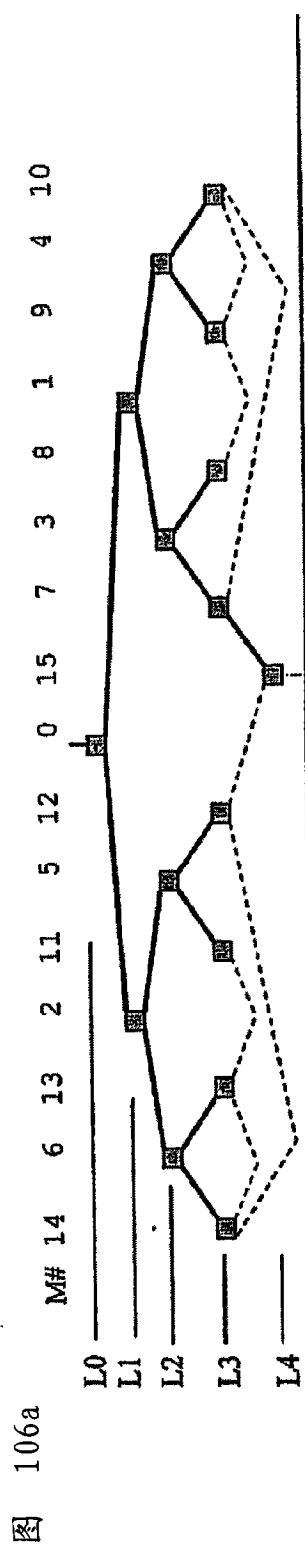


图 105b





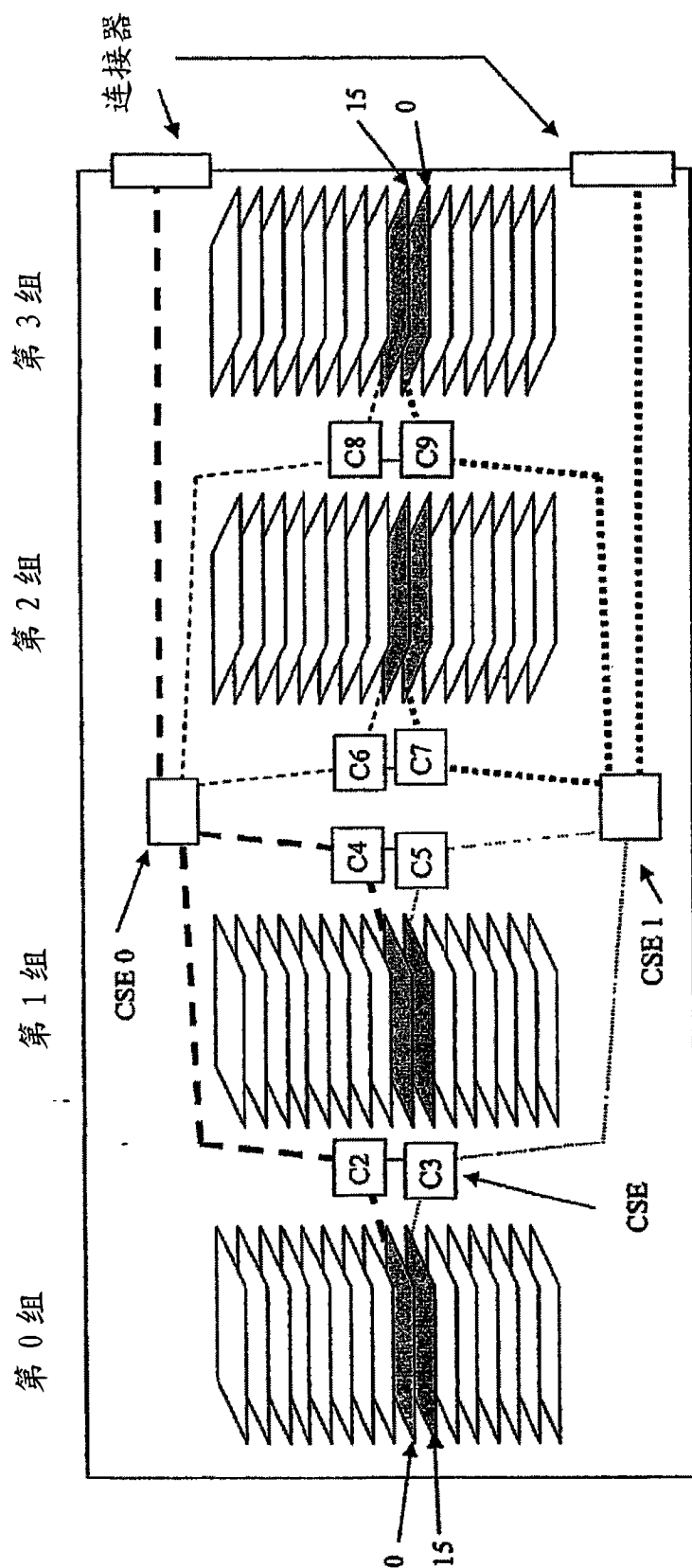


图 107

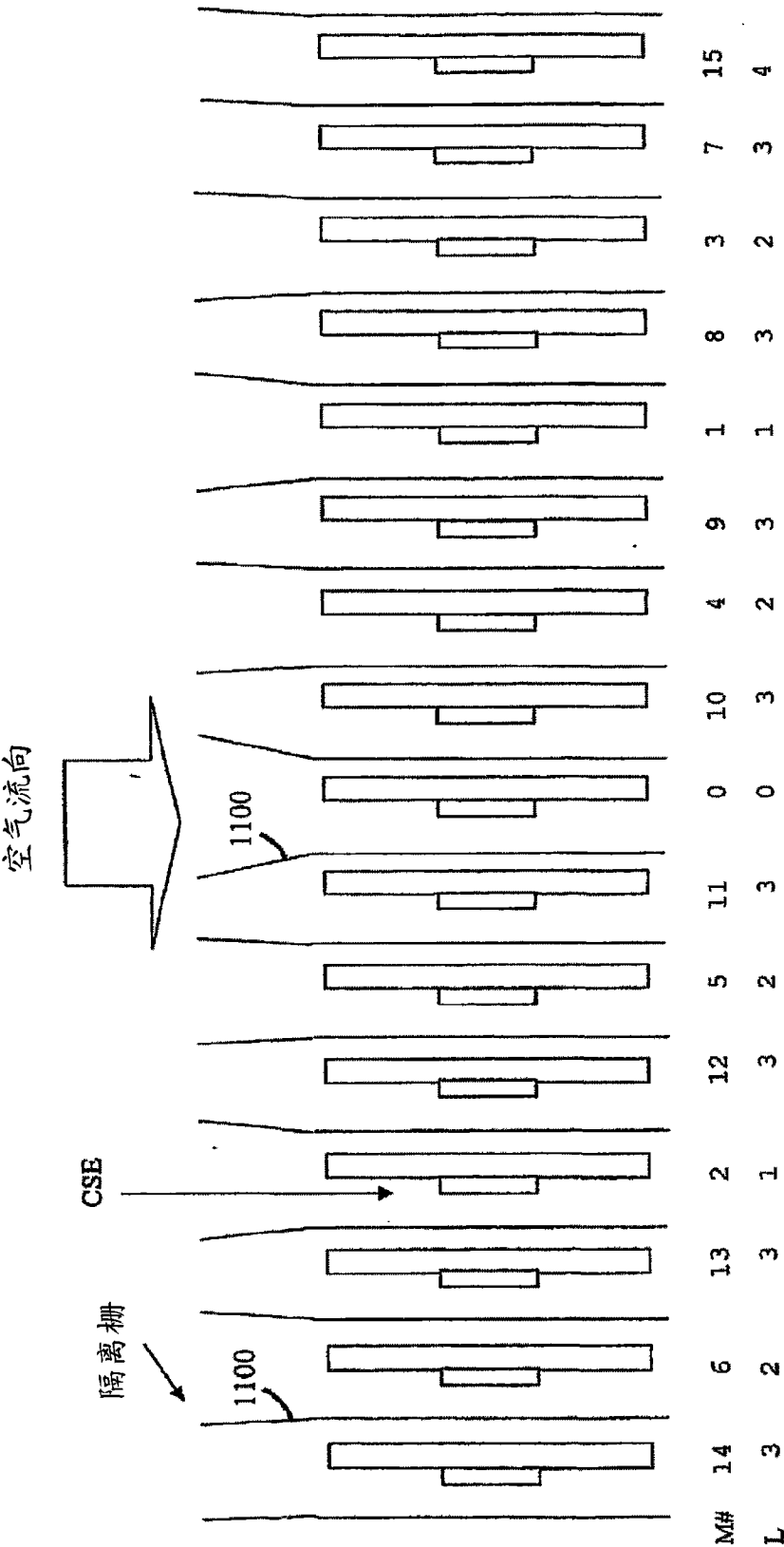


图 108

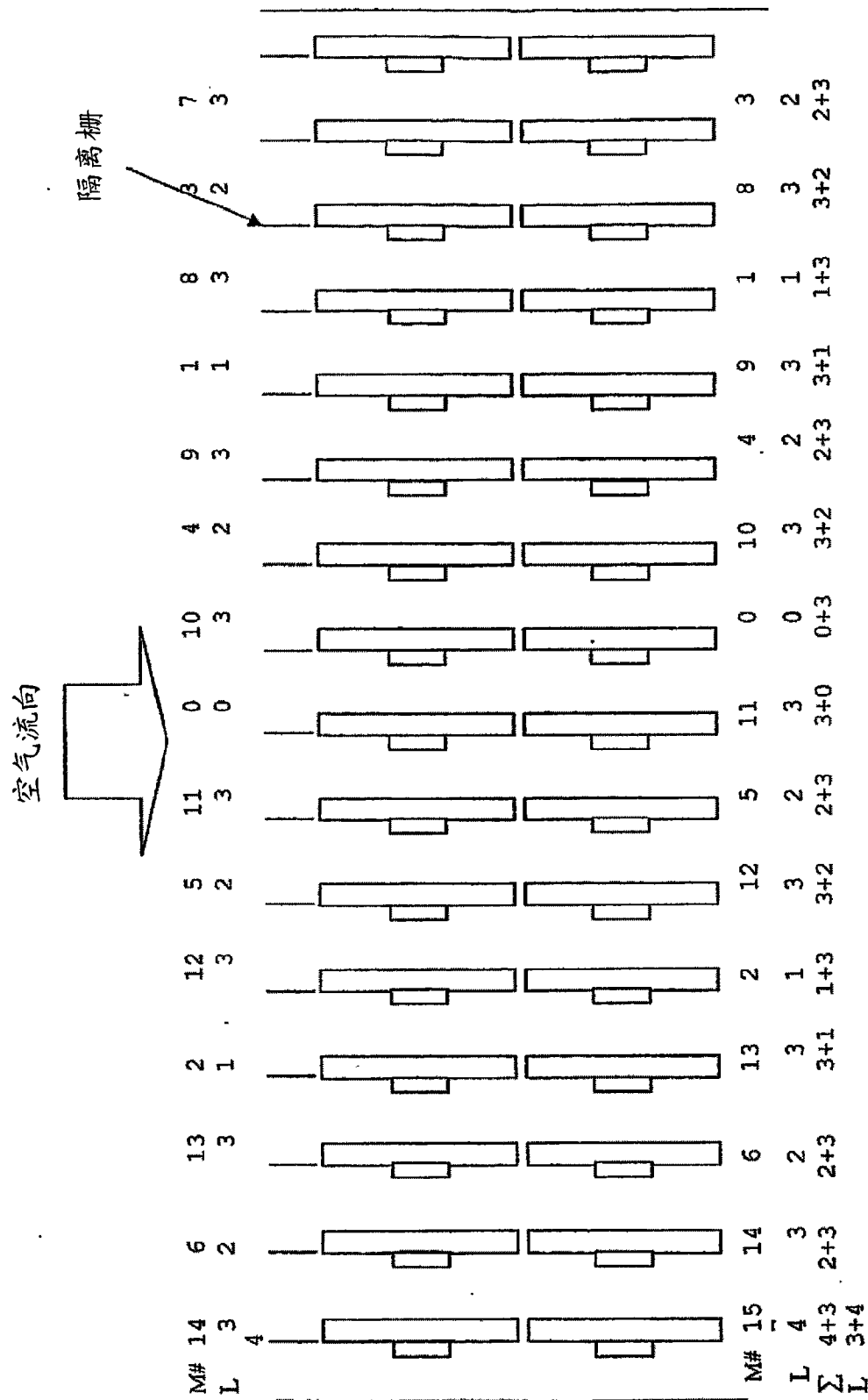


图 109

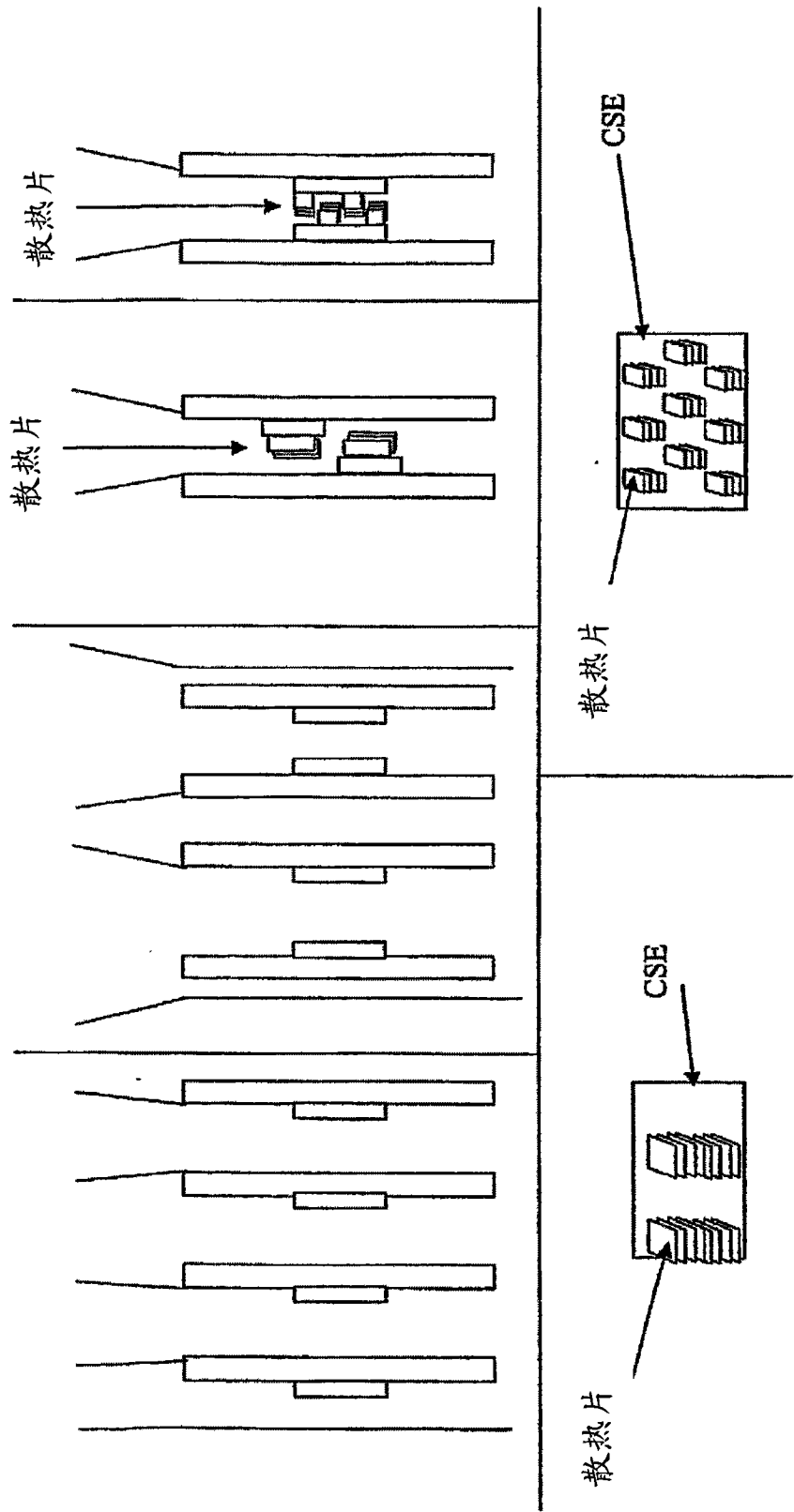


图 110

图 111a

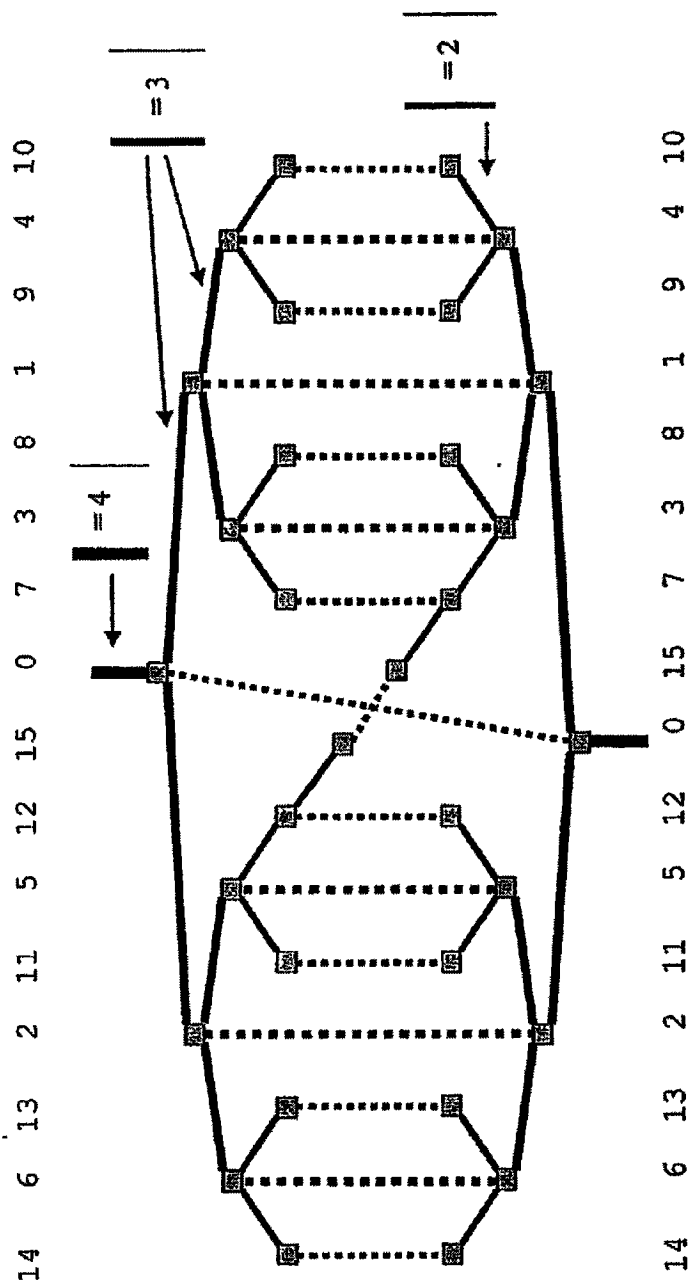


图 111c

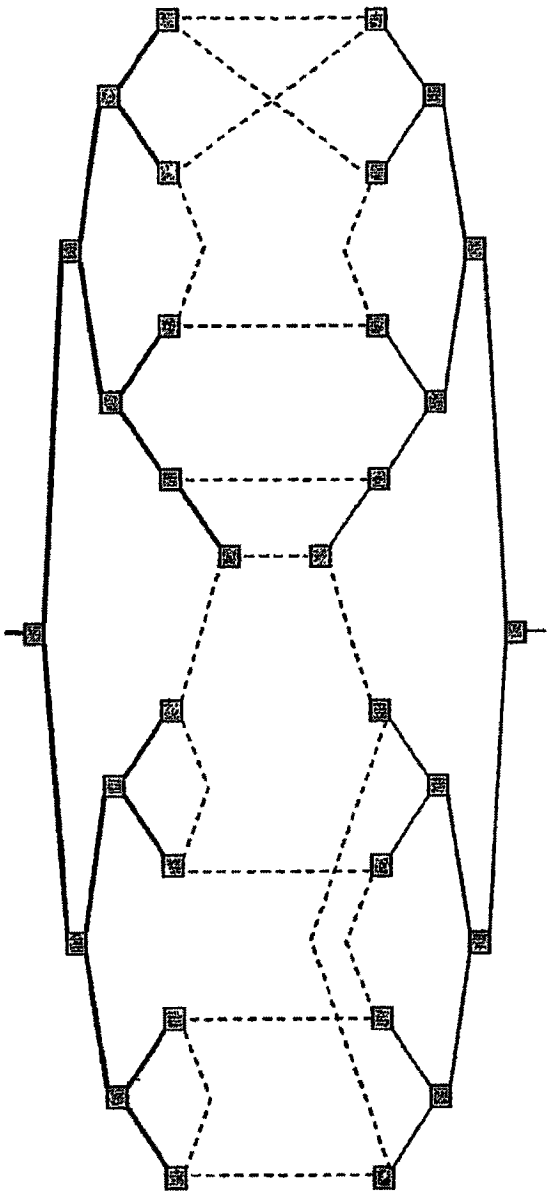
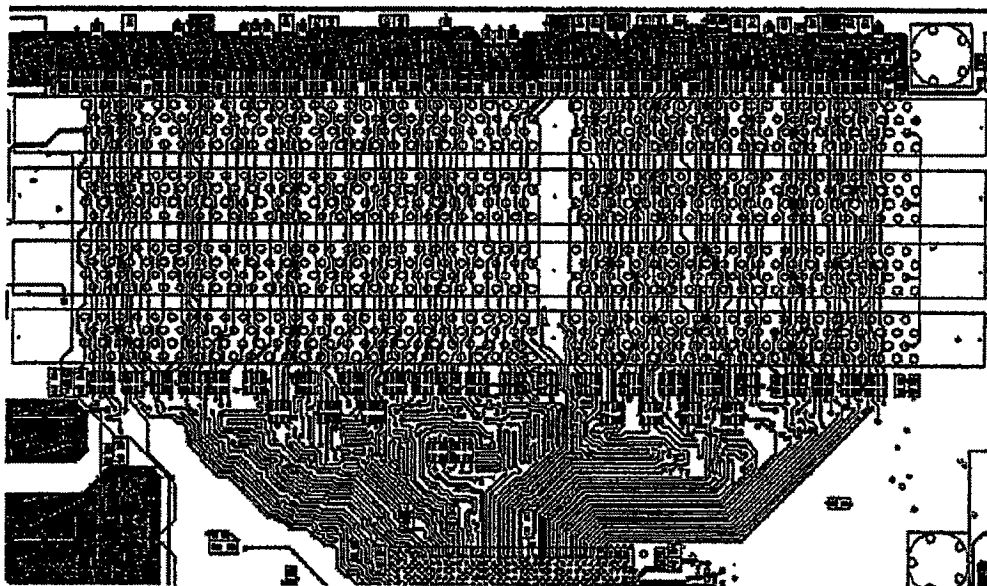


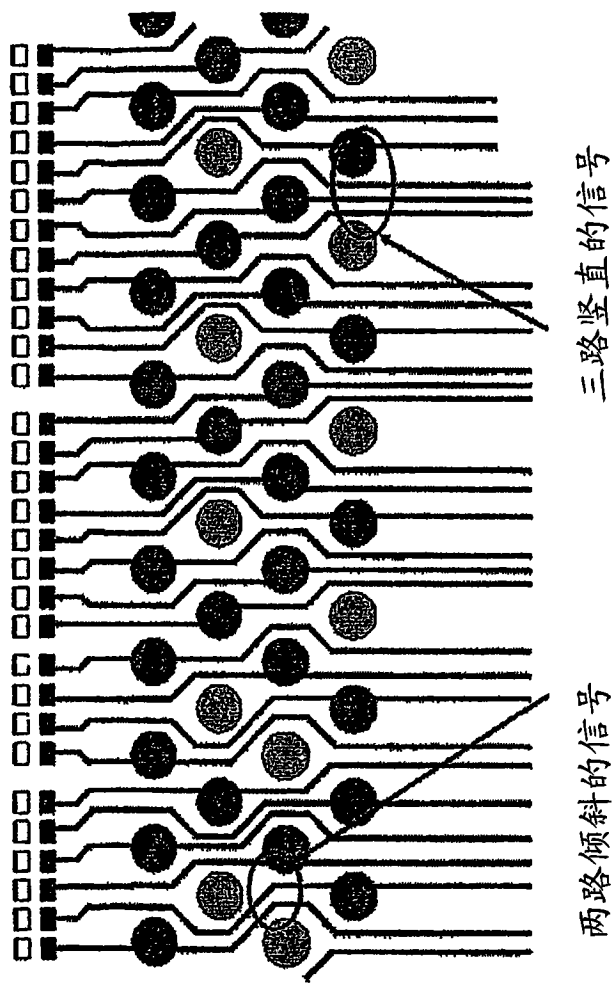
图 111b

DDR 存储器线路的第一层



现有技术

图 112a



现有技术

图 112b

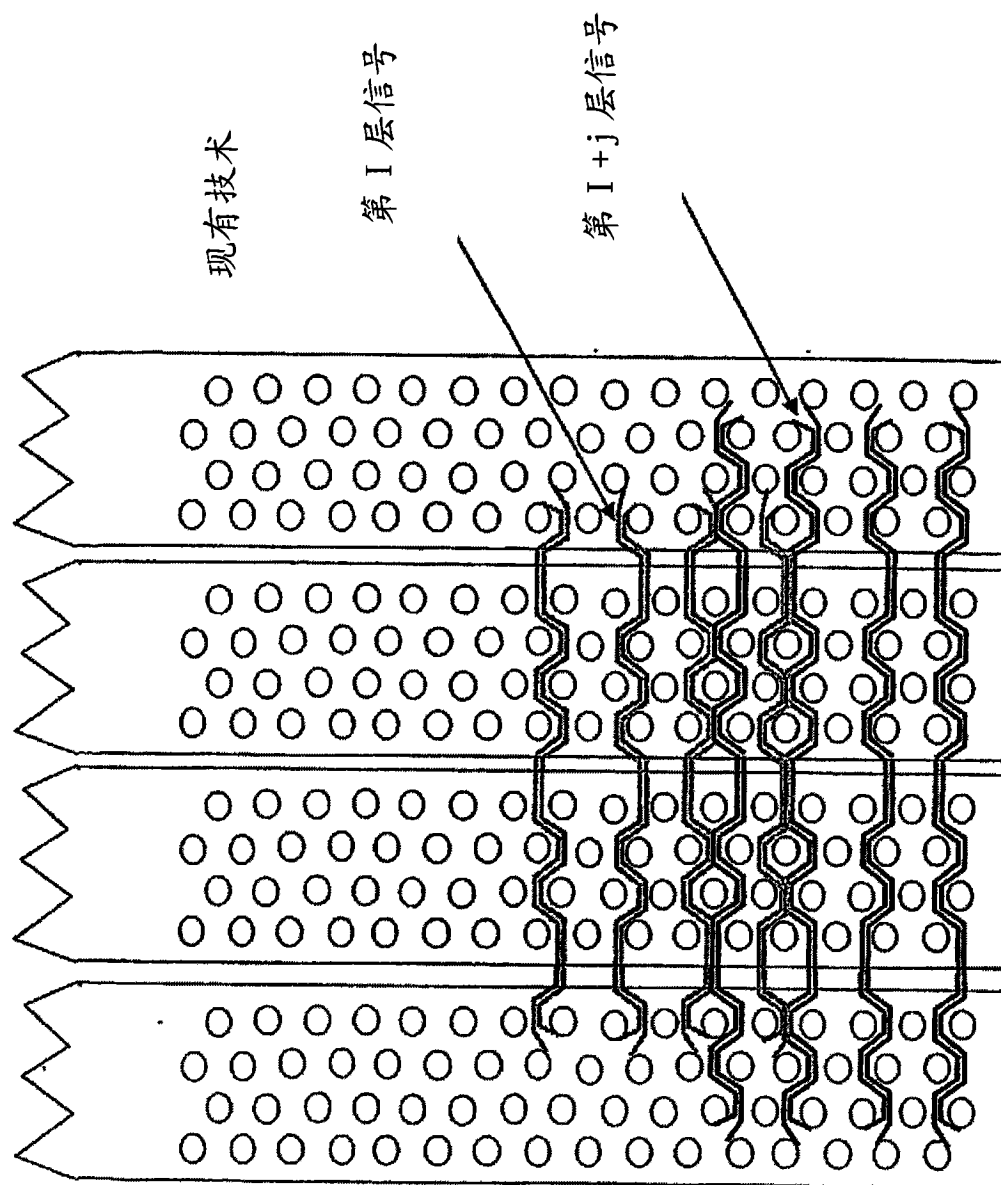


图 112c

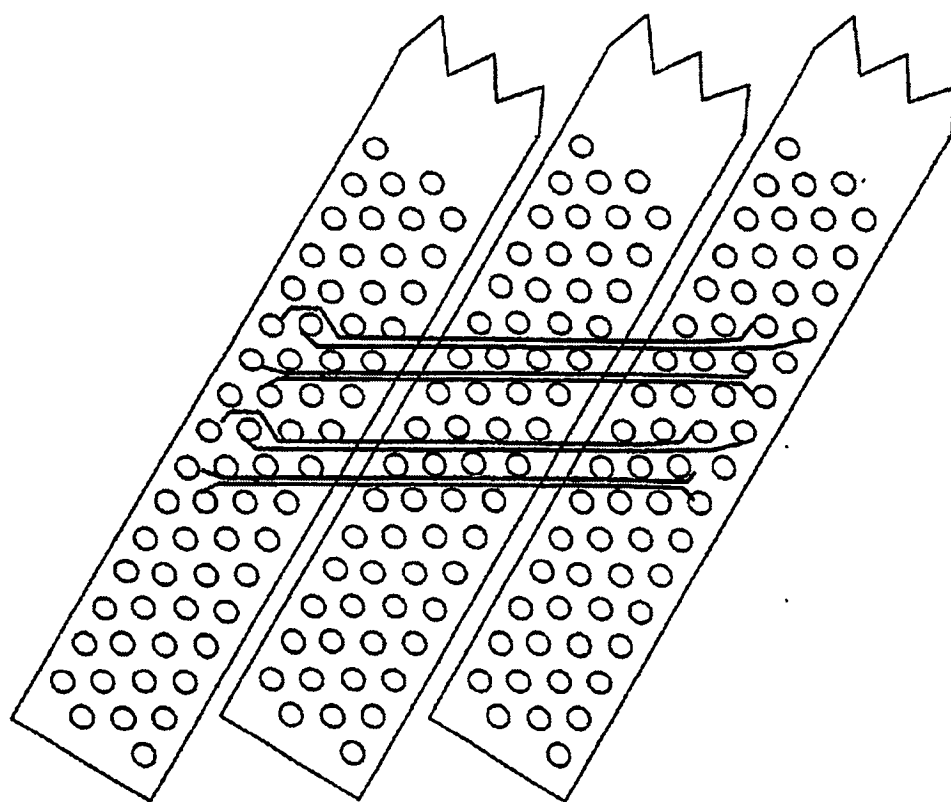


图 113

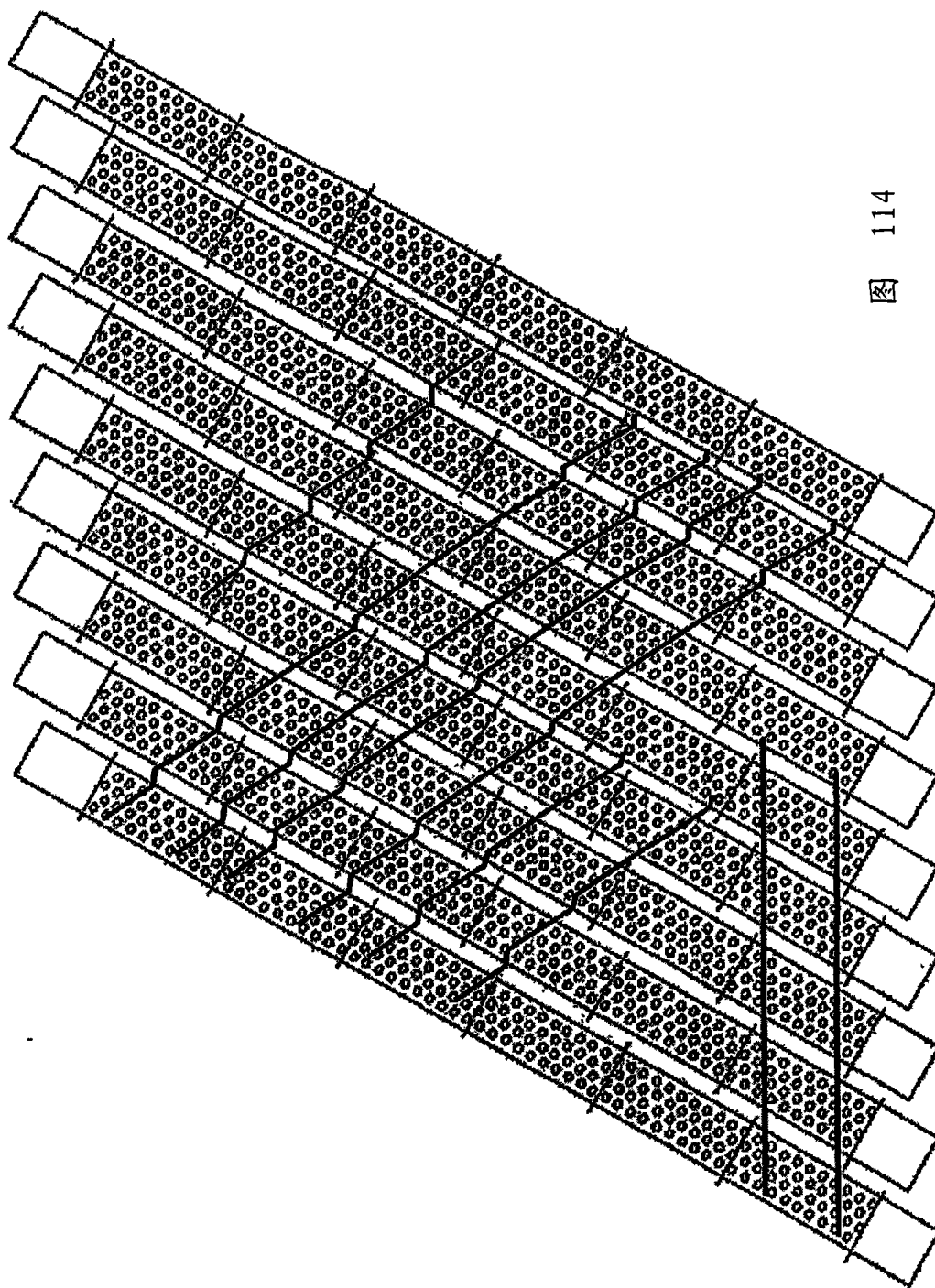


图 114

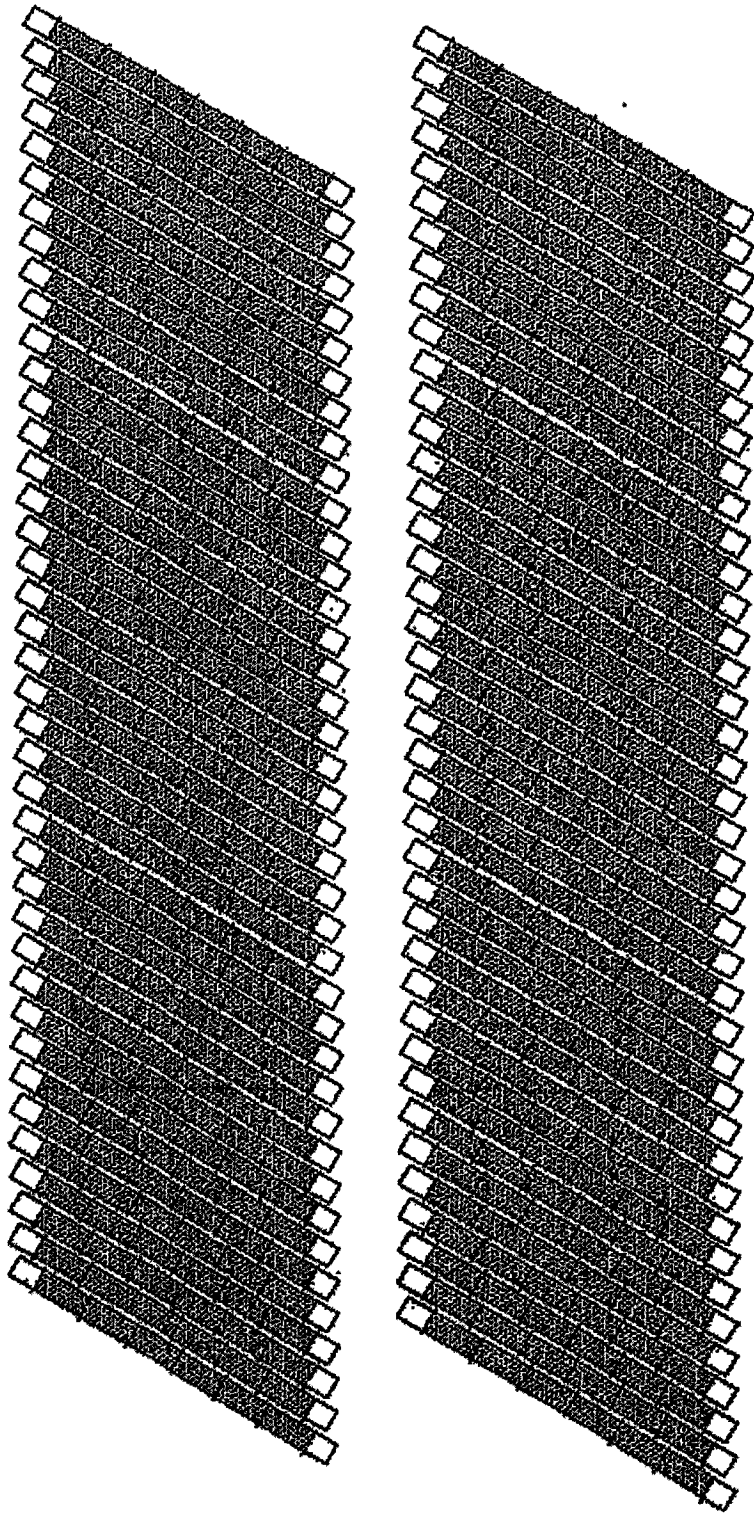


图 115

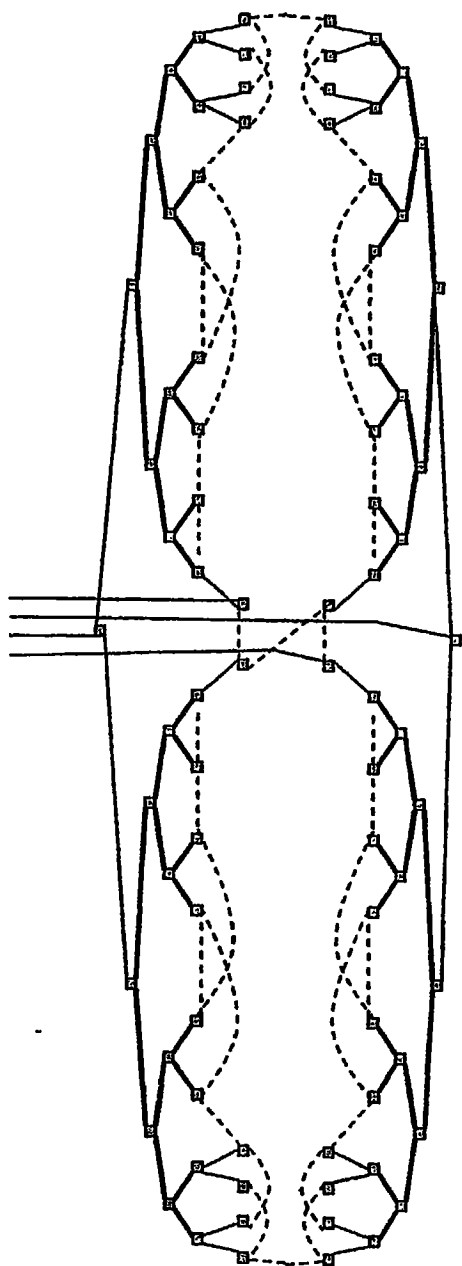


图 116

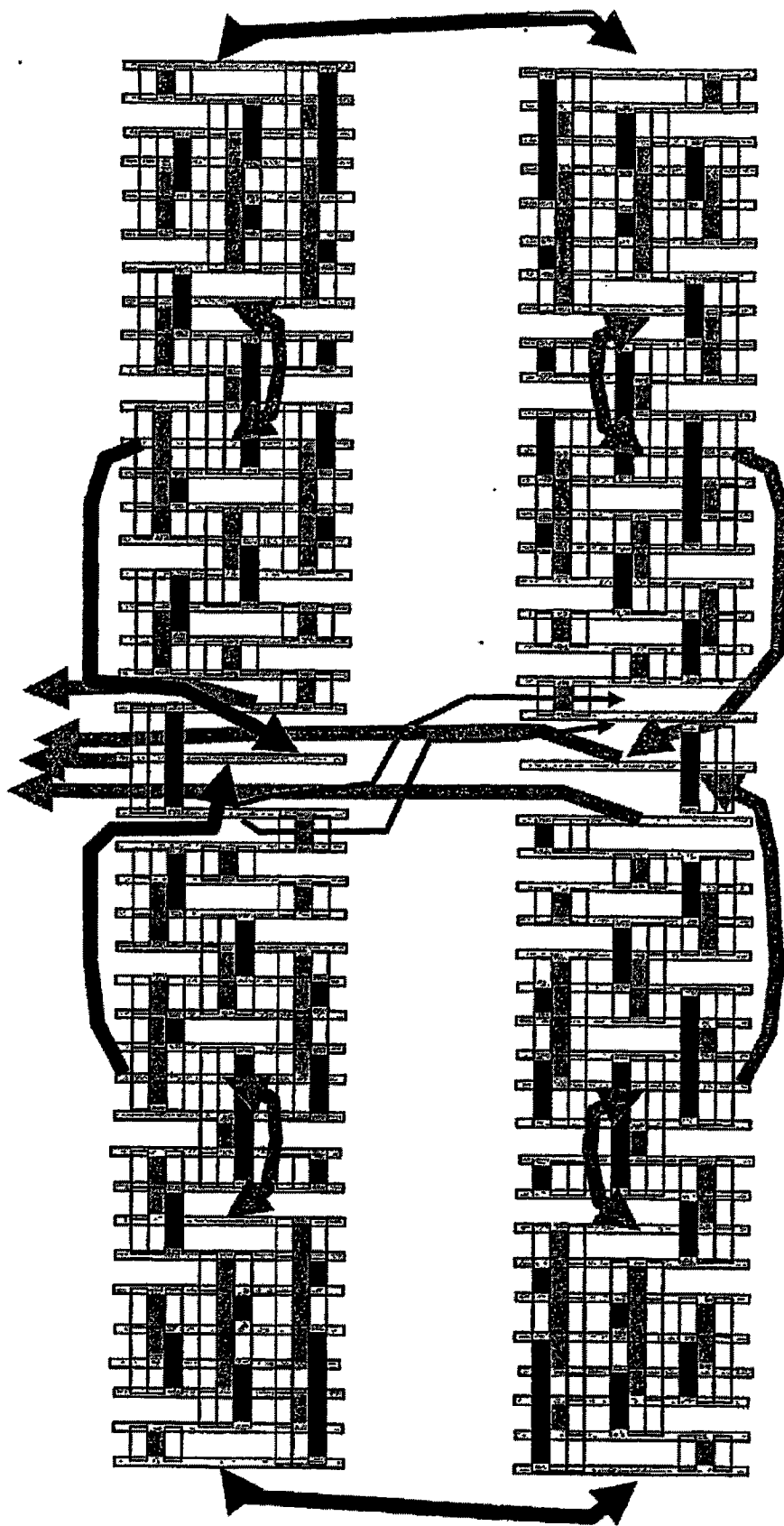


图 117

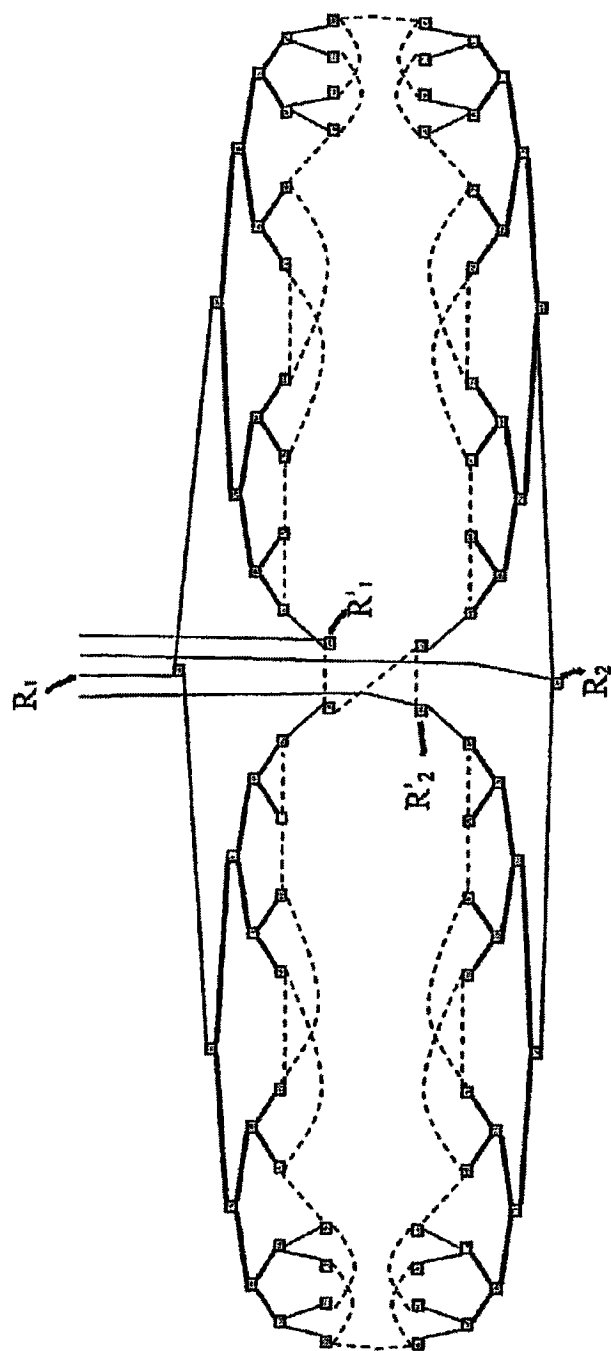
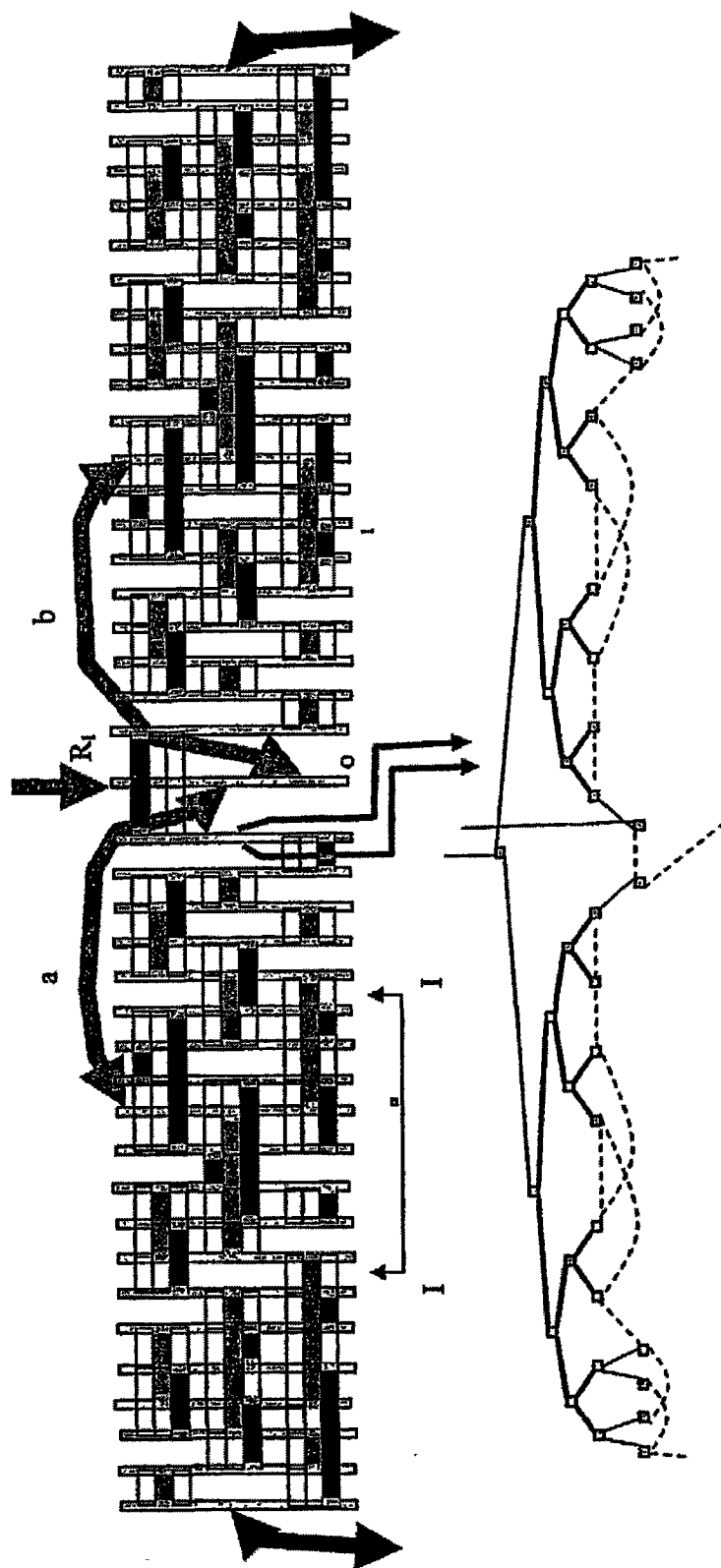
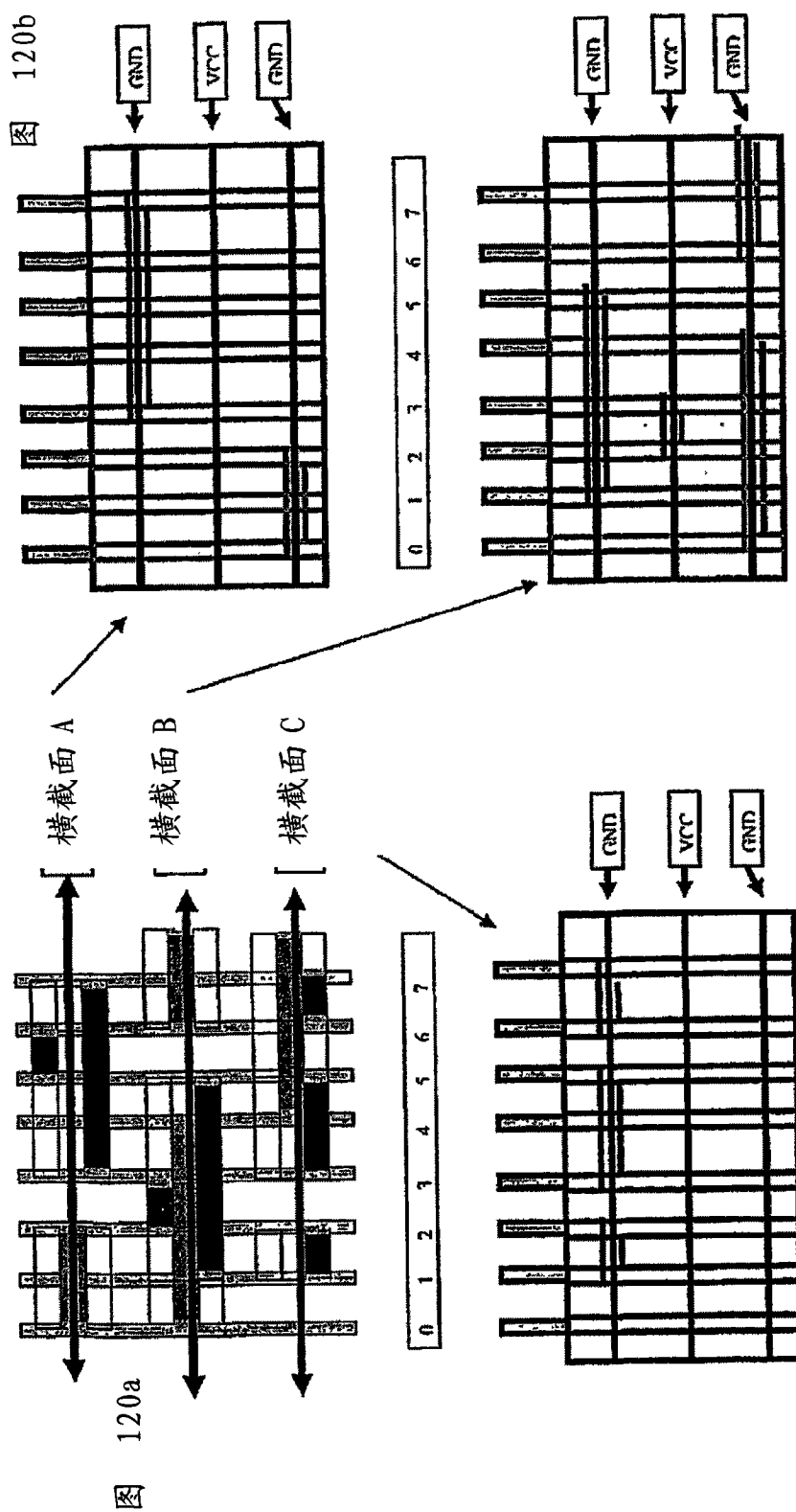


图 118



119



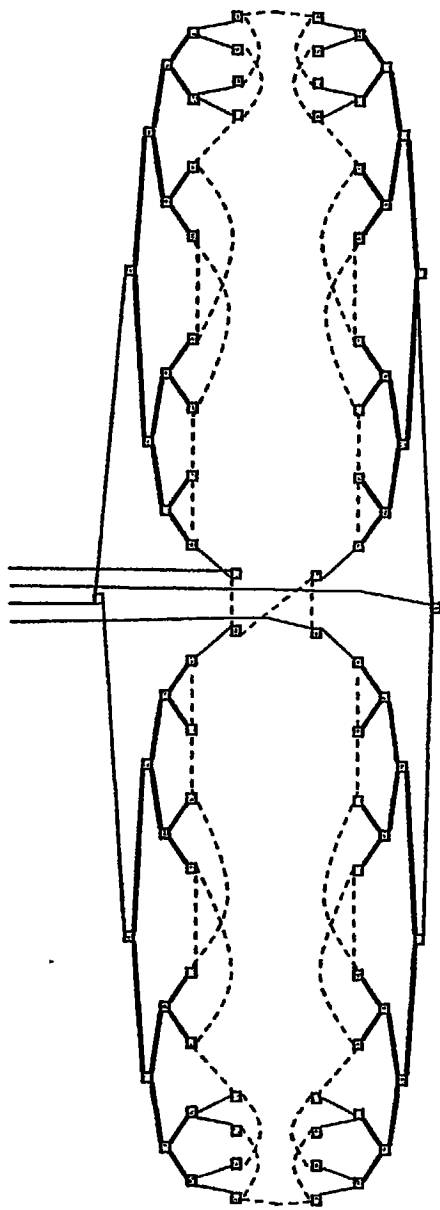


图 121

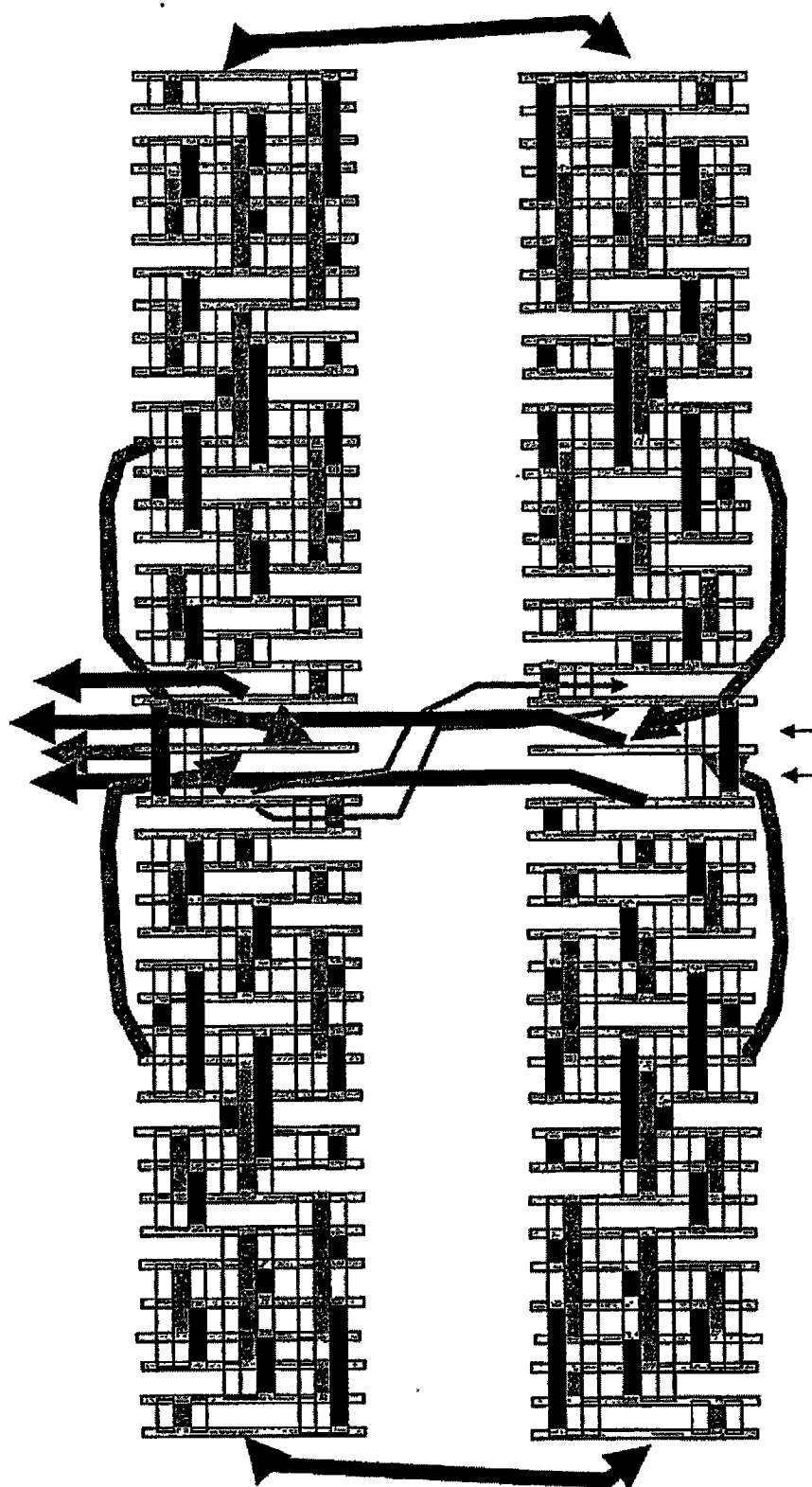


图 122

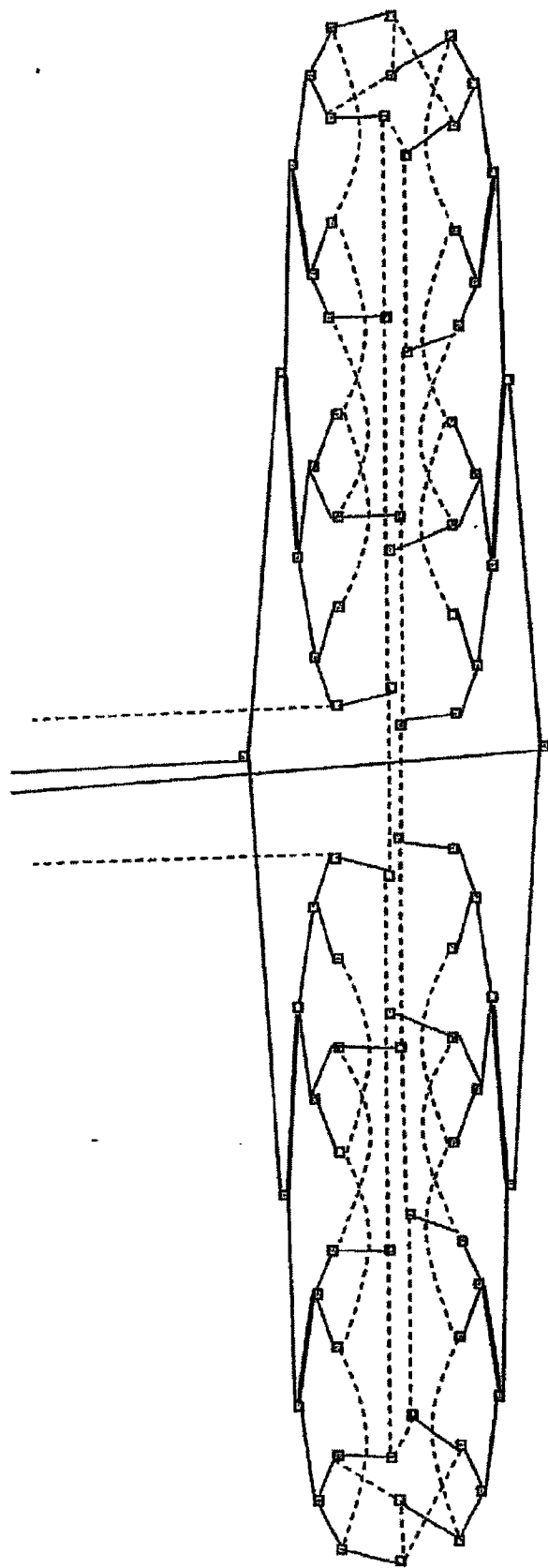


图 123

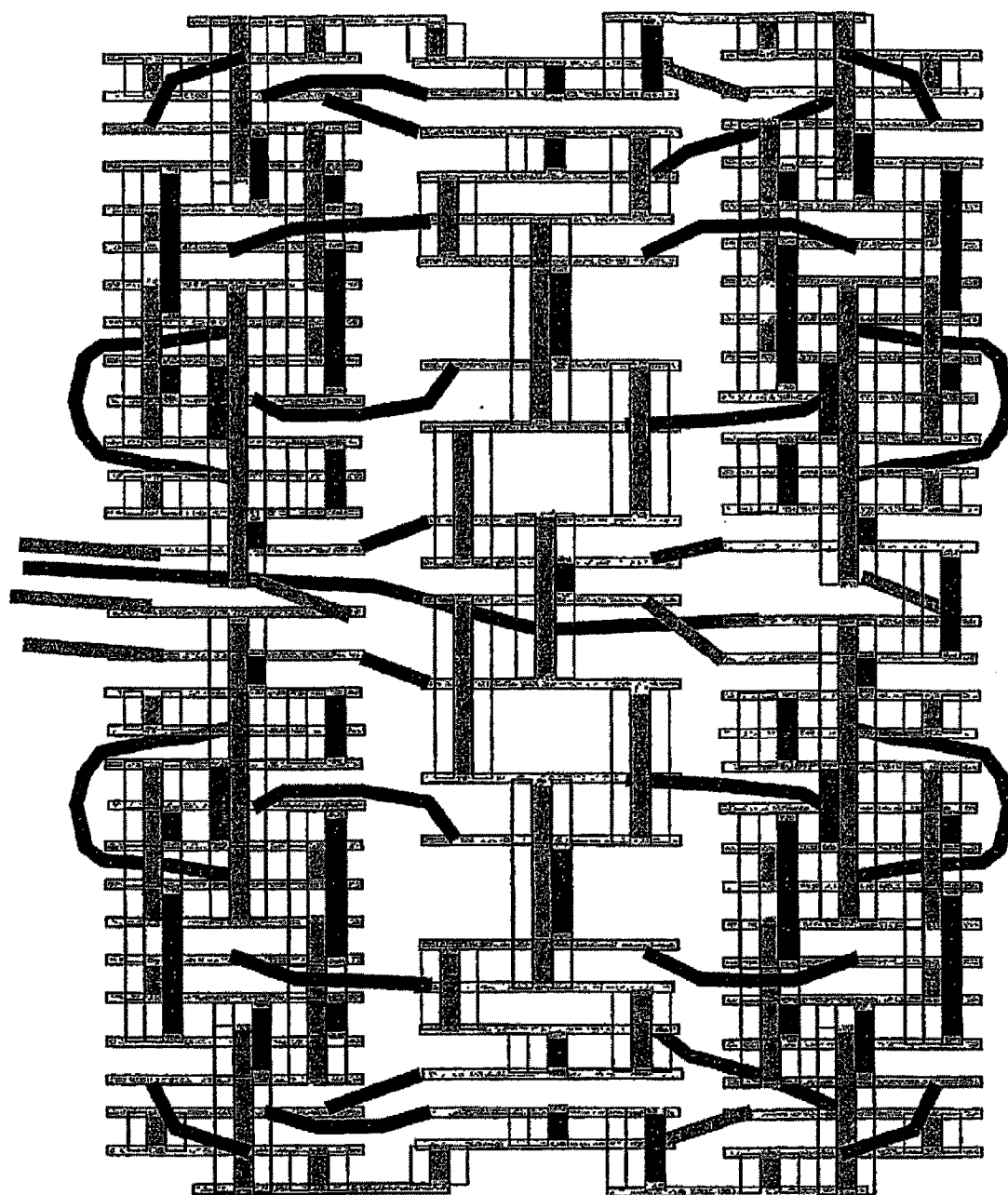


图 124

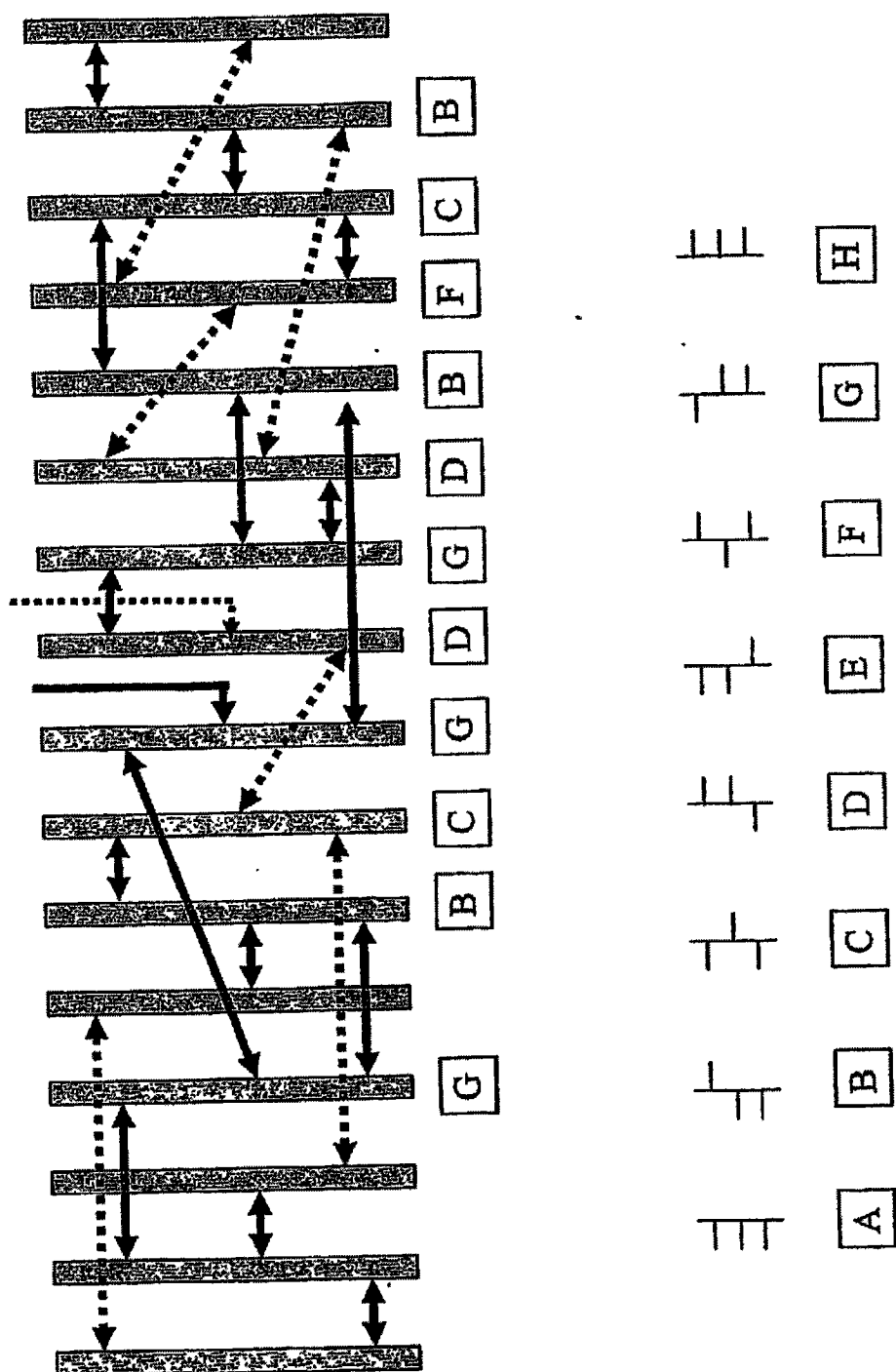


图 125