

[19]中华人民共和国专利局

[51]Int.Cl.⁴

H04Q 11/04

H04J 3/12

H04M 7/00



[12]发明专利申请公开说明书

[11]CN 86 1 01329 A

[43]公开日 1986年11月12日

[21]申请号 86 1 01329

[22]申请日 86.2.6

[30]优先权

(32)85.2.7 (33)美国 (31)699,420

[71]申请人 美国电话电报公司

地址 美国纽约州10022

[72]发明人 马木德·阿马德

威尼·劳伦斯·施里纳

[74]专利代理机构 中国国际贸易促进委员会专利

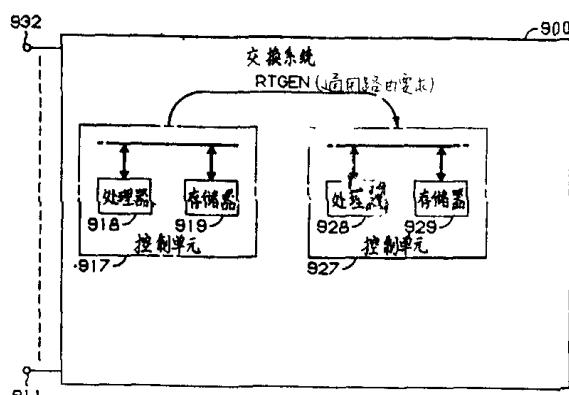
代理部

代理人 刘晖

[54]发明名称 用分布式数据库在分布式控制交换
系统中终点口的确定

[57]摘要

一个分布式控制交换系统，其中系统控制单元联合完成终端口的确定功能。与终端口相连的控制单元贮存顺序呼叫数据，以确定呼叫是接到终端口或是转接到另一个查询号上去。根据确定，一个给定呼叫是一个顺序呼叫，与起始端口相连的控制单元决定用第二查询号规定的终端口，并将顺序呼叫简化为起始端口和第二终端口间的简单呼叫。而与第二终端口相连的控制单元并不知道起始端口完成的呼叫是一个顺序呼叫或是一个简单呼叫。



CN 86 1 01329 A

权 利 要 求 书

1. 在一个有多个口的交换系统中，该交换系统至少由第一与第二控制装置所控制，其中上述第一控制装置贮存了多个第一数据库关系与一个第一程序，该第一程序有多个程序状态并规定该第一数据库关系的读出以判定该多个口中呼叫所要的终点口，上述第二控制装置贮存了多个第二数据库关系与一个第二程序，该第二程序有与上述第一程序的多个程序状态相对应的多个程序状态，并规定该第二数据库关系的读出以判定上述多个口中呼叫所要的终点口；

一个读上述第一与第二数据库关系中的一个或多个，以判定从上述多个口的一个起点口来的呼叫所要的上述多个口中的终端口，上述方法包括：

为了响应该呼叫，上述第一控制装置启动上述第一程序的执行；

上述第一控制装置执行上述第一程序时所明确的，读出上述第一数据库关系；

上述第一控制装置在判明继续执行上述第一程序所需的数据不在一个给定的第一数据库关系中时、终止执行上述第一程序；

上述第一控制装置向上述第二控制装置发送一份控制报文，指定上述给定第一数据库关系规定读出之前的上述第一程序的程序状态之一；

上述第二控制装置在上述控制报文指定的上述第一程序的程序状态所相应的程序状态上开始执行上述第二程序，而且

上述第二控制装置按执行上述第二程序时所指定的，读出上述第二数据库关系。

2. 在一个有多个口的交换系统中，包括一个多个交换模块的控

制分布，其中每个都与上述口的一个相应的子集结合，该交换模块中的每一个包括：

贮存多个数据关系与一个路由程序的装置，该路由程序规定了对上述数据库的顺序存取以决定上述多个口的终点口；

响应从结合上述多个口的子集收到的地址信号，以供执行上述路由程序；以及

当继续执行上述路由程序所需的数据不在给定的一个上述数据库关系中时，发送一个路由继续要求给上述交换模块中另外模块的装置，该路由继续要求在上述路由程序中访问上述给定数据库关系之前规定一个点，还指明在该规定点开始执行上述路由程序时需要的数据；以及

其中上述每个交换模块还包括响应从上述交换模块的其他模块之一接到的路由继续要求，以启动上述路由程序在该路由继续要求规定的一点上开始执行上述路由程序的装置。

3. 一种按照权利要求 2 所述的控制分布，其中上述数据库关系中的第一个，对于一个特定的上述地址信号的子集，指明由该地址信号规定的上述多个口中的终点口；

其中上述数据库关系中的第二个，对上述特定子集以外的地址信号，标明上述交换模块中的其他模块；以及

其中上述发送装置在响应上述地址信号中不在上述特定子集中的一个地址信号，使向上述第一个数据库关系试作的访问失败时，发送一个路由继续要求给上述交换模块中由上述第二数据库关系指明的一个交换模块。

4. 按照权利要求 3 所述的一个控制分布

其中与上述每个交换模块结合的上述口的子集中，每一个口都由

上述地址信号中的上述特定子集标定。

5. 按照权利要求 2 所述的一个控制分布，其中上述数据库关系中的第一个关系，对上述口的许多多口寻线组的每一组，指明为该组贮存的寻线数据的上述交换模块中的一个交换模块，以及

其中上述发送装置在响应上述地址信号中的一个给定信号并判定该给定地址信号指明上述多口寻线组之一的一个口，而该多口寻线组的寻线数据贮存在上述交换模块的其他模块之一中时、发送一个路由继续要求给上述交换模块中被上述第一数据库关系指明的一个交换模块。

6. 根据权利要求 5 所述的一个控制分布，其中上述多口寻线组中某些组的口是接到话线上而上述多口寻线组的其他组的口是接到干线上。

7. 一个交换系统，包括多个口，多个交换模块，其中每个交换模块都和上述多个口的一个相应的子集结合，以及模块间的连接装置；

上述每个交换模块包括交换装置，用来在与其结合的上述口的子集之间以及相关的上述口的子集与上述模块间的连接装置之间提供通信通路。

上述模块间的连接装置包括用来在上述交换模块的交换装置之间提供模块间通信的通信通路；

上述每个交换模块进一步包括：

用来贮存多个数据关系和一个路由程序的装置，该路由程序规定上述数据库关系的顺序存取以决定上述多个口中的终点口；

响应地址信号的装置，用来在上述每个交换模块结合的上述口的子集上，对收到的指明所要求呼叫的地址信号作出响应并执行上述路

由程序；

当继续执行上述路由程序的必要数据不在上述数据关系中的一个给定关系中时，发送一个路由继续要求给上述交换模块中其他模块之一的装置，该路由继续要求在上述路由程序中访问上述给定数据库关系之前标明一个点，还指定在该标明点上开始执行上述路由程序时必需的数据；以及

响应从上述交换模块中其他模块收到的每个路由继续要求的装置用来在每个路由继续要求所标明的点上开始执行上述路由程序。

8. 一个按照权利要求 7 所述的交换模块进一步包括：

一个响应给定执行上述路由程序的装置，该路由程序是对上述口中一个起点口来的一个给定呼叫决定上述口中的终点口，该装置用来在与上述起点口结合的一个第一交换模块的交换装置和终点口结合的一个第二交换模块的交换装置之间，选择和控制建立上述模块间连接装置的上述通信通路中一个给定通路。

9. 一个按照权利要求 8 所述的交换系统，其中上述第一交换模块还包括用来在上述起点口与上述模块间连接装置中的上述给定通信通路之间，控制上述第一交换模块的交换装置，以建立一个通信通路的装置；以及

其中上述第二交换模块进一步包括用来在上述终点口与上述模块间连接装置中的上述给定通信通路之间，控制上述第二交换模块的交换装置以建立一个上述通信通路的装置。

10. 在一个有多个口的交换系统中，一种控制分布包括：

一个中央控制，包括用来贮存多个中央数据库关系与一个中央路由程序，该中央路由程序规定了用以判明上述多个口中的终点口对该中央数据库关系的顺序存取；还有

至少有一个交换模块包括：

用来贮存相应于上述多个中央数据库关系的多个局部数据库关系的装置，以及规定用以判明上述多个口中的终点口对该局部数据库关系的顺序存取的局部路由程序；

响应从上述口的一个子集收到的指明所要求呼叫的地址信号装置，用来执行上述局部路由程序；以及

当继续执行上述局部路由程序所需要的数据不在上述局部数据库关系的一个给定关系中时，发送一个路由继续要求给上述中央控制的装置，该路由继续要求在上述中央路由程序中，对相应于上述给定局部数据库关系的中央数据库关系访问之前标明一个点，并在标明的点上开始执行上述中央路由程序，指定所需要的数据，以及

其中上述中央控制进一步包括：响应上述路由继续要求，用来在上述标明的点上开始执行上述中央路由程序的装置。

1 1. 一个按照权利要求 1 0 所述的控制分布，

其中上述中央数据库关系中的第一个关系对上述多个口中的每一个口都指定上述地址信号中的一个来标明那个口；以及

其中相应于上述第一中央数据库关系的上述局部数据库关系，对上述每个子集中的每个口都指定上述地址信号中的一个来标明那个口。

1 2. 在一个具有多个口的交换系统中，一个控制分布包括：

一个中央控制以及

至少一个第一交换模块，包括响应从标定要求呼叫的上述多个口中一个第一子集上收到地址信号的装置，用来发送路由要求给上述中央控制；

其中上述中央控制包括：

用来贮存多个中央数据库关系和一个中央路由程序的装置，该中央路由程序规定上述中央数据库关系的顺序存取以决定上述多个口中的终点口；以及

响应上述路由要求用来执行该中央路由程序的装置；以及

其中上述控制分布进一步包括至少一个第二交换模块，该模块包括：

用来贮存相应于上述多个中央数据库关系的多个局部数据库关系的装置，以及规定该局部数据库关系的顺序存取以决定上述多个口中的终点口的局部路由程序；

响应从上述多个口的一个第二子集收到的地址信号的装置，用来执行上述局部路由程序；以及

当继续执行上述局部路由程序所需要的数据不在上述局部数据库关系的一个给定的关系中时，用来发送一个路由继续要求给上述中央控制的装置，在上述中央路由程序中访问相应于该给定的局部数据库关系的中央数据库关系之前，上述路由继续要求标明一个点并且在上述标明的点上指定开始执行上述中央路由程序所需要的数据；以及

其中上述中央控制进一步包括响应上述路由继续要求的装置，用来在上述标明的点上开始执行上述中央路由程序。

1 3. 一个交换系统包括多个口，多个交换模块，每个交换模块与上述口的一个相应的子集结合，模块间的连接装置，以及中央控制装置；

上述交换模块中的每一个都包括交换装置，用来在上述口结合的子集中以及在上述口结合的子集和上述模块间连接装置之间提供通信通路，以及控制装置用来控制上述交换装置建立的通信通路；

上述模块间连接装置包括在上述交换模块的交换装置之间用来为

模块间通信提供通信通路的装置；

上述中央控制装置包括用来控制上述模块间连接装置建立的通信通路的装置，以及用来贮存多个中央数据库关系和中央路由程序的装置，该中央路由程序规定在上述多个口中判定终点口的上述中央数据库关系的顺序存取；

上述每个交换模块的控制装置进一步包括：

用于贮存多个相应于上述多个中央数据库关系的局部数据库关系装置和在上述多个口中判定终点口的上述局部数据库关系规定顺序存取的局部路由程序；

响应地址信号装置，该地址信号是从与上述每个交换模块相结合口的子集接收到的并指明要求的呼叫，用来执行上述局部路由程序，以及

当继续执行上述路由程序所需要的数据不在上述局部数据库关系中的一个给定关系中时，用来给上述中央控制装置发出一个路由继续要求的装置，该路由继续要求在上述中央路由程序访问相应于上述给定局部数据库关系的中央数据库关系之前标明一个点，并指定在该标明的点上开始执行上述中央路由程序时需要的数据；以及

其中上述中央控制装置进一步包括响应该路由继续要求的装置，用来在该路由继续要求标明的点上开始执行上述中央路由程序。

1 4. 一个按照权利要求 1 3 所述的交换系统，

其中上述中央控制装置进一步包括响应从上述多个口中的一个起点口来的给定的呼叫决定上述多个口中的一个终端口并对上述中央路由程序作一次给定的执行的装置，用来在跟上述起点口结合的交换模块的交换装置和跟上述终点口结合的交换模块的交换装置之间，选择和控制建立上述模块间连接装置的通信通路中的一个给定通路。

15. 一个按照权利要求 1 3 所述的交换系统，

其中上述每个交换模块的上述控制装置进一步包括响应为了从一个上述多个口的一个起点口来的一次给定的呼叫决定上述多个口中的终点口，对上述局部路由程序进行一次给定的执行装置，用来向上述中央控制装置发出一份为上述给定的呼叫指明上述终点与起点口的控制报文；以及

其中上述中央控制装置进一步包括响应上述控制报文的装置用来在跟上述起点口结合的交换模块的交换装置和跟终点结合的交换模块的交换装置之间，选择和控制上述模块间连接装置的上述通信通路中建立的一个给定通路。

16. 一个按照权利要求 1 3 所述的交换系统

其中上述中央控制装置进一步包括：当继续执行上述中央路由程序所必需的数据不在上述中央数据库关系的一个给定关系内时，用来发送一个第二路继续要求给上述交换模块中的一个给定的模块的装置，该第二路由继续要求在上述局部路由程序访问相当于上述给定中央数据库关系的局部数据库关系之前标明一个点，并指定在该标明的点上开始执行上述局部路由程序所需要的数据，以及

其中上述给定交换模块的控制装置进一步包括响应上述第二路由继续要求的装置，用来在该第二路由继续要求标明的点上开始执行上述局部路由程序。

17. 在一个有多个口的交换系统中，一个中央控制装置和多个分布的控制装置，其中每个都跟上述多个口的一个子集结合，其中上述中央控制装置贮存一个规定拨出信息的翻译以决定上述多个口中的终点口的中央数据库，而上述分布控制装置中的每一个都贮存一个分布数据库，该分布数据库是上述中央数据库的一个子集，并规定拨出

的信息的翻译以决定与那个分布控制装置相结合口的子集中的终点口；

一种决定从上述口中的一个起点口收到的拨出信息所标明的上述多个口中的终点口的方法，该方法包括：

由结合在上述起点口的分布控制装置处理上述拨出的信息以取得数据库访问信息；

由结合在上述起点口的分布控制装置根据上述数据库访问信息访问存在该分布的控制装置中的分布数据库；

在不能按上述接收的拨出来的信息决定上述多个口中的终点口时，由结合在上述起点口的分布控制装置将上述数据库访问信息发送到上述中央控制装置，以及

由上述中央控制装置用上述数据库访问信息访问上述中央数据库以决定上述收到的拨出信息所标明的上述多个口中的终点口。

18. 在一个有多个口的电信交换系统中，一种控制分布响应起源于上述多个口之一的呼叫接通要求，控制建立接往终点口的一个交接途径，该控制分布包括多个交换模块，每个交换模块包括：

用来贮存多个数据库关系和在一个规定状态结束的多状态路由程序装置；

执行贮存在与多状态路由程序结合的交换模块内，为了顺序访问贮存在上述结合模块中的数据库关系以建立一个数据结构，为一个呼叫接通要求辨明终点口的身份；

响应一个呼叫连接要求的措施，用来开始执行上述结合的路由程序；

发送一个指明上述结合程序的一个终结状态的路由继续要求给上述交换模块中的另一个模块的措施；以及

响应从上述交换模块中的另一个模块收到的路由继续要求，在上述收到的继续要求所指明的状态上开始执行上述结合的路由程序的措施；

19. 一个按照权利要求18所述的控制分布，其中上述发送装置发送一个判明部分建立起来的数据结构并指定该结合程序的一个终端状态数据，其中上述响应收到的路由继续要求的装置能响应一个收到的要求，在接收要求指定的状态开始执行上述结合的路由程序以根据一个收到的部分建立起来的数据结构产生一个辨明一个口的身份的数据结构。

20. 在一个有多个口和第一与第二控制装置用来控制上述多个口响应该口中发生的呼叫接通要求互连的通信交换系统中，该控制装置中的每一个都有包括数据项目的数据库关系，响应起源于上述口之一的呼叫接通要求而决定终点口的方法，包括下列步骤：

以执行一个在上述第一控制装置中具有多个预定状态程序的方式在上述第一控制装置中读上述数据库关系；

在一个需要的数据项在上述第一控制装置中的上述数据库关系中找不到时结束在上述第一控制装置中的上述程序的执行；

从上述第一控制装置发送一份报文给上述第二控制装置指明上述第一控制装置中上述程序在结束以前的状态；

在上述第二控制装置中执行一个相应于上述第一控制装置中的上述程序的状态的预定状态读出上述数据关系，而起始状态是在上述发送报文指明的状态上进行的。

说 明 书

用分布式数据库在分布式控制交换系统中终点口的确定

本发明涉及分布式控制的交换系统，更具体地说，该系统呼叫建立功能是分布在系统的各个控制单元中的。

存储程序控制交换系统传统地包括一部按存储器中存储的程序来控制交换功能的中央计算机。虽然近来的一些交换系统已经将呼叫处理功能分布在几个系统控制单元之中，但是呼叫建立所牵涉到的许多耗费时间的任务仍旧由一个中央控制按经典方式完成。例如在一个大家知道的交换系统中，交换功能分布在许多交换模块内。每个交换模块具有多个口，并在接入模块各口的话线与干线之间提供连接。牵涉到接入不同模块的话线与干线的呼叫，由一个交接这些模块的时分割多路交换机来接通。每个交换模块包括一个控制本模块的交换功能的控制单元。系统还包括一个控制时分割多路交换机的交换功能的中央控制。在该系统中的呼叫处理，要求在接通线路之外还执行许多功能。虽然与呼叫有关的许多实时内涵工作，例如信号处理，是由那些交换模块控制单元完成的，其他一些工作，较明显的如决定每次呼叫的交换系统终端口的识别，是由系统中央控制所完成的。终端口决定功能包括这些步骤：如呼叫筛选，决定是否需要话线或干线，将拨号换成实际系统地址，以及寻找干线组或复线寻线组中的空闲线。这些都是耗费时间的工作，它们涉及广泛的数据库搜索和数据操作。

这种类型模块系统的重要优点之一是它的容量能紧密地匹配特定应用的要求。但当系统变得越来越大而交换模块的数目增加时，对与

终端口决定的功能有关的、按每个呼叫的任务，系统中央控制的性能在总的系统呼叫处理的容量上加了上限。

在终端口决定功能是分布的情况下，人所共知的一种安排是有多个由纽带干线相互联接的专用小交换机。一个给定的专用小交换机对向这个小交换机连接的话线或干线的呼叫，能完成终端口决定功能。但对接到其他专用小交换机上的话线或干线的呼叫，决定终端口的数据就不在给定的专用小交换机中出现。给定专用小交换机存储足够的数据来决定终点专用小交换机与相连的纽带干线。只是抓住一条交连的干线，把原来的拨号在这条干线上传送过去。终点专用小交换机按标准呼叫来响应这个拨号，即，这个号好象是接到该专用小交换机的线上拨来的一样，然后完成向终端口的接线。

虽然这个已知的安排有其优点，即对接入这个交连系统之一的话线与干线的呼叫均可接通而不需要中央控制，但总的呼叫处理能力降低了，因为在专用小交换机之间呼叫时，第一个专用小交换机的处理没有被第二个专用小交换机好好利用来接通这次呼叫。相反，第二个专用小交换机处理从交连纽带干线送过来的号码，正象处理从第二个专用小交换机发出的呼叫一样。因此，第一个专用小交换机做的处理工作，相当于在这个交连系统中对总处理要求增加了净工作量。

基于此种情况，该技术领域中的一个公认的问题，就是决定终端口的耗时任务由系统中央控制完成时，对分布控制交换系统的呼叫处理容量受到限制，或者将那任务分布于分布控制单元时，实际上导致系统总处理负荷的增加。

前述问题已被解决而且在一个交换系统中根据本发明已使技术前进一步，在这个系统中两个或更多的系统控制单元互相配合来完成决定终端口的功能，方法是当第一个控制单元不能完全决定终端口时，

这个单元得出的部分结果能被第二个或以后的控制单元所利用，使得这些耗时的数据库存取工作不会无必要地重复。

根据本发明的一种方法已用于有多个口的交换系统上，该系统至少是被第一个和第二个控制单元所控制的。第一个控制单元贮存了许多第一个数据库关系或表格，以及一个具有多种状态并能确定第一个数据库关系的读出以决定呼叫的终端口的第一个程序。第二个控制单元类似地贮存许多第二数据库关系和一个第二程序，这个程序有相应于第一程序状态的程序状态，并确定第二数据库关系的读出以决定呼叫的终端口。根据本发明，第一程序的执行由第一控制单元起动，在执行第一程序时定下的第一数据库关系中的一个被读出。在肯定了继续执行第一程序的必要数据在给定第一数据库关系中不存在时，第一程序的执行就告结束。一个控制报文于是就传送到第二个控制单元，在给定第一数据库关系中被确定读数之前确定了第一程序的程序状态之一。第二控制单元在控制报文确定的第一程序的程序状态相对应的程序状态上起动第二程序的执行，在执行第二程序时定出的第二数据库关系之一被读出。

一个说明本发明的典型控制安排，已用在一个多口交换系统上。这个控制安排包括多个交换模块，每个模块都结合一个相应口的子集。每个交换模块都包括一个存储一些数据关系和一个路由程序的存储器，路由程序规定了在找定终端口时，对数据库关系有次序的存取。每个交换模块还都有个处理器，在执行路由程序时对从有关接口子集收到的地址信号作出反应。当继续执行路由程序所必需的数据在一个给定数据库关系中不存在时，交换模块就向其他交换模块之一发送路由继续要求。路由继续要求指定了路由程序进入给定数据库关系之前的一点，还定出了在指定点开始执行路由程序时需要的数据。另一个交换

模块对路由继续要求作出反应，在要求所指定的点上，起动本身路由程序的执行。

参考附图阅读下列说明，可以对本发明更全面的了解。附图如下：

图1是一个分布式控制交换系统，用来说明本发明的重要原理。

图59是图1中包括的存储器存储的内容布置；

图2是一个时间分隔交换系统，这里称为系统I，与美国专利第4,322,843号所公开的系统实质上一样；

图3是系统I用的一个时隙交换单元和伴随的控制单元的较详细的框图；

图4是每个时隙交换单元都具有的、用来跟系统I中的时间分隔多路交换器通信的接口单元框图；

图5是时间分隔多路交换器的接口单元框图，用来跟系统I的时隙交换单元通信；

图6是系统I的数据字格式框图；

图7是说明系统I的一个典型呼叫接通步骤的功能框图；

图8是系统I路由程序状态框图；

图9——图13，按图57排列着，表示系统I所用路由程序的流程图；

图14——图18规定系统I的一些报文，数据结构和数据库关系；

图19——图21，当按照图22排列时，呈现出一个在此称为系统II的交换系统，是在系统I的基础上综合四个单独的远程交换模块到系统中去；

图23是说明系统II的第一个典型呼叫接通步骤的功能图；

图24是系统II中的路由程序状态图；

图25——图29是当按照图58排列时，用于系统II中的路由程

序流程图；

图 30 和 31，表示系统 II 的第二和第三典型呼叫的接通步骤；

图 32 规定了系统 II 用的一些报文、数据结构和数据库关系；

图 33 到 35，当按照图 36 排列时，呈现出一个在此称为系统 III 的交换系统图，该系统有四个远程交换模块，如系统 II 一样，但在系统 III 中不是四个独立的模块，而是被交连在一个组内，此处称为一个群集；

图 37——图 39 表示系统 III 中处理顺序呼叫时典型呼叫的接通步骤；

图 40 表示系统 III 的另一实施方案的典型呼叫接通步骤；

图 41 是一个此处称为系统 IV 的交换系统图，该系统用与系统 I 实质上相同的硬件体系结构，只是终端口决定功能在所有情况下都仅由交换模块的合并处理完成；

图 42 将图 25——29 的流程图作了某些修改，用以规定系统 IV 中路由程序的操作；

图 43 到 47 是系统 IV 的典型呼叫接通步骤的功能图；

图 48 是一个此处称为系统 V 的交换系统图，其中系统中央控制已完全不负担呼叫处理的职能；

图 49 是系统 V 用的另一种控制分布单元图；

图 50、图 51 和图 51——图 54 是按照图 55 排列的，在系统 V 中用以实现通路寻找程序的流程图；

图 56 是系统 V 的一个典型呼叫接通步骤的功能图。

图 1 是分布控制交换系统 900 的概略图，用来说明本发明的某些重要原理。交换系统 900 是由多个控制单元控制的，用来有选择地互相连接多个口，图 1 中只画出控制单元 917 和 927 以及口 911 与

932。这些口可连接到模拟或数字话线以及任何一种众所周知的各种类型的干线控制单元917包括处理器918和存储器919。

图59表示存储器919的一种存储安排。处理器918在其他许多程序中，也执行一个贮存在存储器919中的存储块913（图59）中的一个路由程序。处理器918将关键性的数据，包括程序变量的初始值和执行路由程序时得出的结果，写入存储块914的一组数据结构之中，如写入RDBLK、CFBLK，GRPBLK，TERMBLK。模块913中的路由程序是用来实现由多个程序状态组成的有限状态机，用以为呼叫完成终端口决定功能。终端口决定功能包括呼叫筛选，决定要求话线或干线，将拨号译为系统口的实际系统地址，以及寻找干线组或复线寻找组中的空线。为了完成这个功能，路由程序定出多个数据库关系的读出，例如FLXEDRI，SCRNLNG，DNTRAN，ROUTLNG，PORTGROUP，GROUPOORT，MHG，LNSTAT，TRKG，TKOWNER，TKQUE，TKSTAT，及MODTRAN关系。这种数据关系存进存储器919的存储块915中（图59）。

控制单元927类似地包括处理器928和存储器929。贮存在存储器929中供处理器928用的路由程序实质上跟存在存储器919中的路由程序一样。存储器929中也贮存了相当于存储器919的存储块914中存的一样的数据结构，和相当于存储器919的存储块915中存的一样的数据库关系，这些数据结构和数据库关系被处理器928用于执行它的路由程序，正如相应的结构和关系被处理器918使用的情况一样。但是应当注意到，存在相对应的数据库关系中的独立的数据元或组，一般说是各不相同的。例如：存储器919中存的DNTRAN关系也可包括查询号码中某一特定集的、或系统口中某子集的查询号码翻译信息，而存储器929中存的DNTRAN关系也许是其他集的查

询号码或口的查询号码翻译信息，或者也许是全部系统口和它们指定的查询号码需要的一切信息。此外，用来贮存在多口寻线组中寻找空闲所需的动态忙／闲数据的数据库关系，及复线寻线组的 LNSTAT 关系或干线组的 TKOWNER、TKQUE、TKSTAT 关系，也可将一个给定寻线组的这种数据只存入一个控制单元中。

作为一个例子，假定一个给定的地址信号，例如代表一个拨号的信号，一串拨号脉冲或双声多频序列或一个数字报文，由口 911 接收到。为了响应这个收到的地址信号，控制单元 917 的处理器 918 开始执行它的路由程序。这种执行定出了存在存储块 915 中某些数据库关系的读出。按照这个例子，读了 SCRNLNG 关系，这种读出的结果就存在存储器 919 的 CFBLK 数据结构中。假定规定的终点是话线而不是干线。读出 SCRNLNG 关系的结果被用作访问 DNTRAN 关系的键不过假定所要的查询号码翻译信息不在存储器 919 的 DNTRAN 关系之内。当访问 DNTRAN 关系失败时，处理器 918 就结束了路由程序的执行，而控制单元 917 向控制单元 927 发出了一个路由继续请求，此处也称为通用路由请求或 RTGEN 报文。路由继续要求将程序状态定在已定的 DNTRAN 关系读出之前，并定出在这一程序状态中起始执行程序所需的任何变量的值。在控制单元 927 中，处理器 928 响应路由继续请求，在相应于要求定下的状态中，开始执行它的路由程序。换句话说，处理器 928 从处理器 918 偏离的点上，开始执行其路由程序。不是再去存取 SCRNLNG 关系，而是将处理器 928 的路由程序变量置在处理器 918 已获得诸值的同一初始值上，然后在读出存储器 929 中的 DNTRAN 关系。在本例中，假定存储器 929 内的 DNTRAN 关系含有给定地址信号所需的查询号码翻译信息，而这种信息表明了口 932 的全局口的识别。因此处理器 928 能完成这个呼叫的终点口

决定任务。于是交换系统900的各个不同控制单元合作，在起点口911与终点口932之间建立通信通道。

作为第二个例子，假定在口911上收到另一个地址信号，但必要的查询号码翻译信息是存在存储器919的DNTRAN关系中。因此，当处理器918执行它的路由程序时，对SCRNLNG和DNTRAN关系的存取都是成功的。但是读出DNTRAN关系的结果表明与收到的特定地址信号有关连的线是复线寻找组的一部分而不是个别的线。于是就访问PORTGROUP关系以取得这个复线寻找组的号码。反过来用这个号码去读MHG关系，这又指明这个特定组的寻找数据是贮存在控制单元927的存储器929里。存储器919内的存有这类复线寻找组的寻找数据的LNSTAT关系已被访问，但访问失败了。与前一个例子一样，控制单元917对控制单元927发出了路由继续要求。路由继续要求将程序状态规定在已明确的LNSTAT关系的读出之前，并定出在这个状态开始执行程序所需要的任何变量的值。在控制单元927中，处理器928响应这个路由继续要求，在对应于要求所明定的状态的状态下开始执行它的路由程序。在这个例子中，路由继续要求在MHG关系的读出前定了一个程序状态。因此处理器928开始执行它的路由程序，并成功地读了MHG和LNSTAT关系。于是处理器928按照路由程序，根据事先对这个寻线组规定的寻线算法，在寻线组上找到一条空线去接通这次呼叫的终点口。

在一个分布控制交换系统的控制单元之间，终点口决定功能的这种分布可用阶层式或非阶层式的编排来实现。阶层式编排包括一个系统中央控制和多个分布式控制单元。当分布式控制单元之一访问它的一个数据库关系而失败时，一个路由继续要求就传送到中央控制的较高层。如果中央控制以后访问一个数据库关系也失败时，中央控制就

传送一个路由继续要求回到下面的分布式控制单元之一。

在非阶层式编排中，所有的系统控制单元都在同一水平上工作。当任何控制单元在读出一个数据库关系而失败时，一个路由继续要求就被传送到另一个控制单元。不论是在阶层式或非阶层式编排中，参予找定一个呼叫终点口任务的下一个控制单元的决定，是在每个控制单元所存的数据库关系中规定好了的，当然在阶层式编排中，一个分布式控制单元访问失败后，下一个参予的控制单元总是系统中央控制。

在下面的详细描述里，四个典型系统，称它们为系统Ⅱ到V，阐明了根据本发明分布控制式交换系统中终点口的决定方法。系统Ⅱ和Ⅲ是包括系统中央控制的阶层式编排。虽然终点口决定功能仅分布到这些系统内包括的远程交换模块，显然这种功能可以类似地分布到系统内所有交换模块上去。系统Ⅳ和V是非阶层式编排，其中不包括系统中心控制参予的终点口决定功能。

详细说明

下面的叙述涉及五个时间分隔交换系统，此处称为系统I到V，它们在呼叫处理功能分布于全系统的程度上各有不同。

系统I是一个时分隔交换系统，其中交换功能被分布到多个交换模块上去，每个模块都接着许多话线和干线。每个交换模块都在连接本模块的话线和干线中间完成接线。牵涉到不同模块上接着的话线和干线的呼叫，是由交连各个模块的时分隔多路交换器来接通的。每个交换模块包括一个控制本模块交换功能的控制单元。系统也包括一个控制时分隔多路交换器交换功能的中央控制。所有系统内部的呼叫，要求选用一个所谓网络时隙。对模块之间的呼叫，网络时隙 用于从一个交换模块，经过时分隔多路交换器到另一个交换模块的传送。对

模块内部的呼叫，网络时隙 用于在交换模块内从一条话线或干线接到另一条话线或干线。（在目前的实施方案中，模块内部的呼叫用了两个网络时隙，每个传输方向用一个。）虽然在系统 I 内，呼叫处理功能是分布的，这就是说与呼叫有关的实时内涵任务，例如信号处理，是由交换模块控制单元完成的，路由功能，此处定义为找定终端口的功能，选用网络时隙，以及在呼叫是模块间进行时，建立时分隔多路交换通道，这些却被集中起来由系统中央控制来完成的。这是描述的系统 I 跟美国专利 4,322,843 号中公布的时分隔交换系统实质上是相同的。

建立在系统 I 之上的系统 II，只是在系统中综合了四个独立的远程交换模块。但在系统 II 中，路由功能是由远程交换模块控制单元和系统中央控制以分布方式完成的。分布做的很有成效，一个控制单元做过的工作，特别是耗时的数据库访问任务，无须由第二个控制单元重复进行。

系统 III 内也有四个远程交换模块，但它们不是独立的而是互相交连成一组，此处称为群集。在系统 III 内，路由功能也是以分布方式完成的。此处描述的系统 II 和 III，在许多方面跟美国专利申请书 493,683（1983 年 5 月 11 日递交的申请）中描述的包括远程交换能力时分隔交换系统是一样的。但在美国专利申请号 493,683 的系统中，路由功能没有分布而是由系统中央控制集中完成的。

系统 IV 表明在这种分布控制交换系统中处理的顺序呼叫，即转接传呼或依次接通的呼叫这是一种有效的方法，其中顺序呼叫总是被简化为简单呼叫，即只牵涉到两个口的呼叫。

系统 V 用了实质上跟系统 I 同样的硬件体系结构，但所有呼叫处理功能，除开选择网络时隙和为模块间的呼叫建立时分隔多路交换器

通道之外，都分布给交换模块而不涉及系统中央控制。特别是对每次呼叫决定终端口的功能，只用交换模块进行合作处理。

在系统Ⅳ中，查询号翻译功能是以这种方式做的，使其对用户话机指定查询号时有灵活性但又将每个交换模块需要的存储设施的容量减到最小。组员跨多个交换模块的多口寻线组是以一种有效的方式管理的，这就是对每个多口寻线组指定交换模块之一作为那个多口寻线组的组控制单元。

在系统Ⅴ中，余下的呼叫处理功能，即选定网络时隙，本文称途径寻找，以及对模块间呼叫以此建立时分隔多路交换途径，也分布到这些交换模块里去了。在系统Ⅴ中，系统中央控制完全解除了呼叫处理功能，只执行管理与维护功能。

系统Ⅰ

图2中中时分隔交换系统，本文称为系统Ⅰ6，用于互相连接用户话机，例如连接用户话机23到26，以及连接干线，例如干线443到46，并包括一个时分隔多路交换器10，其中包括带64个输入端与64个输出端的时共享空间分隔交换器。还包括29个时隙交换单元，其中专门画出作为代表的时隙交换单元119012。每个时隙交换单元11与12都包括一个双向时隙交换器。此外，每个时隙交换单元11与12都接到时分隔多路交换器10的两个输入端与两个输出端上。在系统中，时隙交换单元11是经由时分隔多路线13与14接到两个时分隔多路交换器输入端的并经由时分隔多路线15与16接入两个输出端的。

在下面的说明中，时分隔多路交换器10的输入与输出端被称为输入／输出端对。

用这个术语是因为一个给定的输入／输出端对的输入端数据字的来源，也就是从这个端对的输出端出来的数据字的终点。如图2所示，输入／输出端对 P1 是与时分隔多路线 13 和 15 相结合的。每个时分割多路线，从 13 到 16，都以帧速度 125 微秒传递数字信息，每帧由 256 个时间分开的通道。于是在每个 125 微秒的一帧期间内，每个时隙交換单元发出和接收高达 512 路数字信息。

每个时隙交換单元唯一地与一个控制单元结合，其中控制单元 17 是与时隙交換单元 11 结合的，而控制单元 18 是与时隙交換单元 12 结合的。此外，每个时隙交換单元都通过个别的时分隔多路线连接多个外围单元，其中话线单元 19 到 22，干线单元 39 到 42 可从图2看出。一个时隙交換单元和它结合的控制单元及外围单元，在此处总称为一个交换模块。话线单元 19 与 20、干线单元 39 与 40 都接到交换模块 201 的时隙交換单元 11 上，话线单元 21 与 22、干线单元 41 与 42 都接在交换模块 229 的时隙交換单元 12 上。每个话线单元都接到几部用户话机上，其中话机 23 到 26 可从图中看到。具体一个时隙交換单元接几个话线单元，一个话线单元接几部话机，要由所服务的用户数目以及这些用户呼叫的频繁程度来决定。每个话线单元从多个用户话机，例如 23 到 26 的终端与众所周知的模拟环相接，并转换呼叫信息，包括模拟式语言信号，转换为数字数据字，然后送到与其结合的时隙交換单元。此外，每个话线单元收听用户话机的服务要求，要为这些用户话机产生某些控制信息。为之语言取样并编码的具体用户话机以及在话线单元和与其结合的时隙交換单元之间传送

产生的编码所使用的具体时分隔多路通路，都是由这个时隙交换单元中的控制单元来决定的。

干线单元，例如 39 与 40，替干线完成类似的功能，例如检测干线的占用，以及控制与探测对其他系统的干线信号。这些干线可以是模拟式的，也可以是数字式的。这种数字式干线的一个例子就是美国专利 4,059,731 号发表的 T1 载波系统，在这个系统上，24 个互相分开的通信电路可以同时进行工作。

用户话机，话线单元和时隙交换单元的关系，对每个这样互连的单元组来说基本上是一样的。因此，虽然下列说明只牵涉到用户话机 23，话线单元 19，和时隙交换单元 11，但也表明其他这种单元组的关系。此外，在干线，干线单元与时隙交换单元之间，也存在类似的关系。话线单元 19 扫描接到每个用户话机上的线，以发现服务要求。当发现这种要求时，话线单元 19 就向控制单元 17 发送一个报文，指出这个要求和提出该要求的用户话机的身份。该报文是通过通信通道 27 传到控制单元 17 的。根据所要求的服务、请求者用户话机的身份和可用的设备，控制单元 17 完成了必要的翻译，并经过通信通道 27 送一个报文到话线单元 19，这个报文规定了在话线单元 19 和时隙交换单元 11 之间的多个时分离电路中，哪一个可用来将信息从用户话机 23 传到时隙交换单元 11。根据这个报文，话线单元 19 就将用户话机 23 来的模拟式信息编码成为数字式数据字，并将得出的数据字在指定的电路上发送出去。话线单元 19 也在指定电路上传出与用户话机 23 结合的用户环路直流情况，即开路、闭路的一个标志。

在将话线单元 19 与时隙交换单元 11 之间的一条时间分离电路指定给前述用户话机之后，控制单元用对指定电路传送的信息进行采样的方法，探听用户话机发来的信号。这种采样操作是经由通信通路

28 进行的。控制单元 17 对用户通路来的信号作出反应，也对从其他控制单元，例如 18，以及中央控制单元 30 来的控制报文作出反应，这些都是用控制时隙交换单元 11 的时隙交换功能来实现的。前面说过，一个时隙交换单元与时分隔多路交换器 10 之间的每条时分隔多路线上，都在每 125 微秒帧上有 256 条通路。这些通路按出现的次序被标上从 1 到 256 的号码。这一序列通路重复出现，使一个给定的通路每 125 微秒都能见到一次。时隙交换功能在控制单元 17 与 18 的控制下取话线单元送来的数据字并将它们放在时隙交换单元与时分隔多路交换器 10 之间的时分隔多路线的一个通路上。

时分隔多路交换器 10 以重复出现时隙帧的方式工作，每 125 微秒帧含有 256 个时隙。在每一时隙中，时分隔多路交换器 10 能按照存在控制存储器 29 中的时隙控制信息、将从它的 64 个输入端中任何一个收到的数据字接到它的 64 个输出端中的任何一个上去。通过时分隔多路交换器接线的编排图形，每 256 个时隙 重复一次，每个时隙都依次加上从 1 到 256 的编号。这样，在第一时隙 TS 1 中，时分隔多路线 13 上通路 (1) 中的信息可能被时分隔多路交换器 10 接到输出端 P64，而在下一个时隙 TS 2 中，时分隔多路线 13 上的下一个通路 (2) 也许就被接到输出端 P57 上。时隙控制信息是由中央控制器 30 写入控制存储器 29 的，中央控制器 30 从各个控制单元，例如 17 与 18，送来的控制报文，产生这个控制信息。

中央控制器 30 和控制单元 17 与 18 利用时隙交换单元与时分隔多路交换器之间的时分隔多路线，例如 13 到 16 中特选的通路，称为控制通路，来交换控制报文。每个控制报文包括多个控制字，而每个 256 个时分离通路帧中每个控制通路能发送一个控制字。结合在一个给定的输入／输出端对上的两条时分隔多路线的同一通路，被事先规

定为一个控制通路。此外，一个给定通路只在一对时分隔多路线上用作控制通路。例如，如果通路 1 被用在时分隔多路线 13 和与它结合的时分隔多路线 15 上作为一条控制通路，其他的时分隔多路线就都不能用通路 1 作为控制通路。在每一个跟控制通路有同一编号的时隙中，时分隔多路交换器 10 将占据那个控制通路的数据字到输出端 P64，并将输入端 P64 接到跟上述控制通路结合的输出端上。下面是操作系统 I 的一个例子，其中通路 1 是时分隔多路线 13 与 15 的控制通路，而通路 2 是时分隔多路线 14 与 16 的控制通路。在时隙 TS 1 中，控制存储器内的信息，除规定了其他连系之外，还规定时分隔多路 13 的通路 1 上的控制字接到输出端 P64，而输入端 P64 的通路 1 上的控制字接到时分隔多路线 15 上。类似地在时隙 TS 2 中，控制存储器 29 内的信息规定将时分隔多路线 14 的通路 2 上的控制字接到输出端 P64，而输入端 P64 的通路 2 上的控制字接到时分隔多路线 16 上。按这种方式操作时，输出端 P64 在与它们送到时分隔多路交换器时的通路号码相同的通路上，从时分隔多路交换器 10 收到所有的控制码。此外，每个控制通路都在与控制通路编号相同的时隙内，被连接以便从输入端 P64 接收控制字。转接到输出端 P64 的控制字被送到一个控制分配单元 31，它把这些字临时存在与那个控制通路结合的位置上。控制通路与控制分配单元 31 中存储位置的结合，使所存信息的来源能分辨清楚。

时隙交换单元发出的每个控制报文都包括一个开头字符，一个终点部分，一个信号部分，和一个结尾字符。终点部分唯一地规定了这个控制报文希望送到的终点。控制分配单元 31 解释了每个控制报文的终点部分，决定每个控制报文的正确终点，然后在与终点结合的控制通道编号相同的通路上，将报文重新发送到时分隔交换器 10 的输

入端 P64。

在按上述情况操作时，时隙交换单元 11 在重复出现控制通路时发送控制字，形成一个终点部分标明为时隙交换单元 12 的报文，这样就将控制报文发送到时隙交换单元 12 了。控制分配单元 31 收集了这些控制字，解释终点部分，并在与时隙交换单元 12 相结合的控制通道编号相同的通路上将报文重新发送给输入端 P64。将控制报文的终点部分定为中央控制器 30，就可将一个控制报文发送到中央控制器 30。在这种情况下，控制分配单元 31 通过通信电路 32 将报文发送到中央控制器 30，而不是将它回送到时分隔多路交换器 10 中去。类似地，将一个终点部分定为某一特定时隙 交换单元的控制报文发送到控制分配单元 31，中央控制器 30 就可将一个报文发送到时隙交换单元之一。这也是利用了通信电路 32 来传送的。控制分配单元 31 的一个特定实施方案的运用情况，在上述美国专利 4,322,843 号中有详细说明。

每一个控制单元，例如 17 与 18，都包括一个存储器 57（图 3），其中贮存了控制与其结合的控制单元的程序，以及关于控制单元的主要功能和与之结合的时隙交换单元与用户的数据。控制单元 17 的主要处理单位是一个处理器 66（图 3），它响应在存储器 57 内贮存的指令而工作的。控制单元 17 包括一个控制接口电路 56，它经过总线 59 从处理器 66 收到指令，对它作出响应，经过通信电路 27 跟外围单元，例如话线单元 19 与 20，干线单元 39 与 40 进行通信。控制单元 17 还包括一个信号处理器 65 和一个数字服务单元 67。信号处理器 65 接收时分隔交换单元 11 收到的每一个数据字，并分析其中信号的部分（见图 6 中 A—G 位），这样就减轻了处理器 66 的实时负荷要求。数字服务单元 67 接受时分隔交换单元 11 收到的每个字的数

据部分去发现从用户来的、转换为 PCM 信号的音调信号。数字服务单元 67 也用来发送 PCM 格式的音调和信号：经过门 51 到用户，经过门 52 到时分隔多路交换器 10。控制接口电路 56，信号处理器 65，数字服务单元 67，以及话线单元 19 的工作情况，都在上述美国专利 4,322,843 号中有详细说明。一个干线单元 39 的例子包括上述美国专利申请书 493,683 号中描述的、是用于 T1 载波系统的数字设备接口。

每个外围单元发送重复出现的帧，每帧包括 32 到 64 条 16 位的数字通路。这个信息被发送到时隙交换单元内的多路调制单元 60（图 3）。多路电路 60 从外围设备接收输出的信号，将这些信号换一个格式从输出时分隔多路线 62 上按每 125 微秒帧 512 条通路发送出去。类似地一个多路解调电路 61 接收时分隔多路线 63 上的 512 条 16 位通路，这些通路按照预定的安排分配给外围设备，例如分配给话线单元 19。此外，多路单元 60 将信息进入的通路从串连形式转换为并连形式，多路解调器 61 则将它收到的信号从并连转到串连形式。时分隔多路线 62 上一条给定通路发送的信息被贮存在一个接收时隙交换单元 50 的与给定通路唯一结合的存储位置上。

一个给定的数据字存入的特定存储位置，是由时隙计数器 54 产生的时隙编号信号规定的。时隙计数器 54 以每时隙一个时隙编号的速率产生 512 个时隙编号的重复序列。在收到一个给定数据字的时隙期间内产生的特定时隙编号，决定了接收时隙交换单元 50 中存这个数据字的存储位置。数据字也以每时隙一个数据的速率从接收时隙交换单元 50 读出。在一个给定时隙期间内从接收时隙交换单元 50 读出的数据字的存储位置是由读出控制 RAM 55 给出的。控制 RAM 55 在时隙计数器 54 的时隙编号规定的地址上每时隙读出一次，这样读出来

的数值被传送到接收时隙交换器 50 作为这个时隙的读出地址。从接收时隙交换器 50 读出的数据字经过一条时分隔多路线 68、一个门 8、一条时分隔多路线 68' 和一个接口单元 69，传送到时分隔交换器 10。从时分隔多路交换器 10 来的数据字，通过接口单元 69 被时隙交换单元 11 收到，然后经过时分隔多路线 70' 门 9 和时分隔多路线 70 传到发送时隙交换器 53。对于接在时隙交换单元 11 的外围单元之间的呼叫，控制 RAM 55 指挥门 8 与 9 的操作，使接收时隙交换器 50 在时分隔多路线 68 上发送的数据字，通过门 8 和 9 以及时分隔多路线 70 传送到发送时隙交换器 53。发送时隙交换器 53 将送来的数据字存在控制 RAM 55 送出的地址所规定的位置上。数据字按时隙计数器 54 规定的地址从发送时隙交换器 53 读出。这样读出的数据字，在时分隔多路线 63 上被发送到一个外围单元，例如话线单元 19。应当指出，控制 RAM 55 可用几个控制存储器来实现，每个存储器与一个特定的电路，例如发送时隙交换器 53 相结合。控制存储器的特定编排，对目前的说明并不重要，可以根据时隙交换单元 11 内部的计时与电路要求而变化。接收时隙交换器 50，控制 RAM 55，时隙计数器 54 以及发送时隙交换器进行的时隙交换的一般原理，在本技术领域是大家熟悉的，此处就不再细述。在时隙存储器内读写数据字的一种安排，在美国专利 4,035,584 号中作了详细的说明。

就要叙述的系统 I 的控制信息交换的主要模式，是将控制报文从一个源时隙交换单元，经过时分隔多路交换器 10 和控制分配单元 31，再回到终点时隙交换单元。还应用第二种通信模式，其中关于一个给定呼叫的控制信息是通过时分隔多路交换器 10、利用指定给这次呼叫的时隙，从源时隙 交换单元发送到终点时隙交换单元去的。呼叫时隙中数据字的 E 位被用作第二种模式通信。但可以看出任何或所有

信号位都可以在第二通信模式中应用。三位起通信路途连续性校验和信号收到表示两种作用。三位累加器 48 和经过导线 193, 194, 和 195 与处理器 66 连系的三位校验电路, 在完成这双重任务时的工作情况, 已在上述美国专利 4,322,843 中作了详细说明。

下面是这个交换系统中各个控制部件的第一种通信模式的说明。处理器 66 在响应一个完整拨号时, 对这个拨号作了翻译, 并编一个控制报文给中央控制器 30 (图 2), 使其能通过时分隔多路交换器 10 为这次呼叫建立一个空闲时隙。这个控制报文被处理器 66 存在存储器 57 里。一个在本技术部门常见型类的 DMA 单元 58 以每帧一个控制字的速率读这个控制报文, 并将这个字发送到接口单元 69 内的一个控制字源寄存器 80 (图 4), 以便在时分隔多路线上发送到时分隔多路交换器 10 上去。类似地, 从其他控制单元和中央控制器发来的控制报文, 由接口单元 69 的一个控制字终点寄存器 92 (图 4) 收下, 并由 DMA 单元 58 传送到存储器 57, 然后由处理器 66 读出。接口单元 69 (详见图 4) 包括一个多路调制/解调电路 75 和两个链接口 78 与 79。多路调制/解调电路 75 接收从接收时隙交换器 50 经由时分隔多路线 68' 传来的数据字, 并将数据字经由时分隔多路线 70' 传送到发送时隙交换器 53 里去。可以回忆一下, 两个条时分隔多路线 68' 和 70' 都是以每 125 微秒帧 512 通路的速度传送数据字的。

多路调制/解调电路 75 将经由时分隔多路线 68' 收到的信息分为两路时分隔多工线 76 与 77, 将每个偶数编号通路中的数据字由时分隔多路线 77 送出, 而将每个奇数编号通路中的数据字在时分隔多路线 76 上送出。这样, 时分隔多路线 76 和 77 每一根都是以每帧 256 个通路的速度传递信息。此外, 多路调制/解调电路 75 将两条

256 通路时分隔多路线 85 和 86 上来信息，合并到 512 通路时分隔多路线 70' 上去。组合是这样进行的，交替发送时分隔多路线 85 和 86 上的数据字，使时分隔多路线 85 上的数据字由时分隔多路线 70' 的奇数通路上传出，而时分隔多路线 86 上的数据字则由偶数通路上传出。时分隔多路线 76 和 85 接到链接口 78，而时分隔多路线 77 和 86 则接到链接口 79。应当注意的是，时隙交换单元 11 是在每帧 512 个时隙（通路）的基础上工作的，而链接口 78 和 79 以及时分隔多路交换器 10 则都是在每帧 256 个时隙（通路）的基础上工作的。此外，发送到时隙交换单元 11 和从它那里接收的数据字通路是完全同步的。这就是说，只要链接口 78 从时隙交换单元 11 接到一个带给定编号的通路，对时隙交换单元 11 来说，两个链接口 78 和 79 都会在接收和发送同一编号的通路。为了保持分隔后的同步，所有在时分隔多路线 68' 上的奇数通路都被多路调制／解调电路 75 延迟了一段时间，使得这些奇数通路与紧接着的偶数通路分别在时分隔多路线 76 和 77 之一上面且基本上同时发送。类似地，时分隔多路线 86 上的从链接口 79 来的每个数据字，都被多路调制／解调电路 75 延迟了一段时间，使它紧跟着本来由多路调制／解调电路基本上同时收到的由时分隔多路线 85 传来的数据字的后面，在时分隔多路线 70' 上发送出去。在下面的说明中，一个给定数据字的时隙是指它对链接口 78 与 79 和对时分隔多路交换器 10 而言的时隙。例如，时分隔多路线 68' 上通路 1 和 2 的数据字，都结合于链接口 78 与 79 及时分隔交换器上的时隙 1 上。链接口单元 78 与 79 中的每一个都唯一地与时分隔多路交换器 10 中的一个输入／输出口对相结合。

链接口 78（图 4）包括接收器 82，用来接收时分隔多路交换器 10 经由时分隔多路线 15 依序传来的数据字，并将这个信息在导线

83 上依序转发出去。一个时钟回收电路 84 接到导线 83 上，从那里回收一个 32,768 兆赫的时钟信号。这个时钟信号是供链接口 78 计时用的。由于后面还要详加说明这里，从时分隔多路线 15 收到的信息，并不一定与时分隔多路线 13 上传出去的有通路同步。为了达到时分隔多路线 76 和 85 上的数据字之间的通路同步，导线 83 上的进入数据字在一个随机存取存储电路 87 中缓冲。导线 83 上的数据字在一个写入地址产生器 88 所规定的位置上写入随机存取存储器 87。写入地址产生器 88 从时钟回收电路 84 收到一个 2,048 兆赫的时钟信号，并作出响应、产生一个和导线 83 上进入的数据字同步的、重复出现的 256 个写入地址的序列。传送到时隙交换单元 11 的数据字是按一个读出地址产生器 89 规定的地址从随机存取存储器 87 读出的，读出地址产生器 89 产生着重复出现的 256 个读出地址的序列。读出地址是从一个偏移电路 90 的信息中导出的，偏移电路 90 接收写入地址产生器 88 所产生的写入地址，然后有效地从这个数字减去一个预先规定的数字。做了减法以后的结果被传送到读出地址产生器 89。以此方式，读出地址产生器 89 产生的读出地址序列写入地址产生器 88 产生的地址大约晚四分之一个帧（64 个时隙）。

接口单元 69 的链接口 78 和 79 以主／奴模式工作，从而保持通路同步。在目前的实施方案中，链接口 78 是主。它不断按上述方式工作。但是链接口 79 的读出地址产生器是由链接口 78 的读出地址产生器 89 的读出地址驱动的。应当注意到，由于时分隔多路线 15 与 16 的长度可能有差别，链接口 79 利用的写入地均与读出地址之间的间隔可能比四分之一帧多一点或少一点。这种情况之发生是由于时分隔多路线 85 和 86 上传送的数据字是通路同步的，但在时分隔多路线 15 和 16 上并无这种同步要求。

在一个给定的链接口上，同一条通路是同时用来发送和接收控制报文的。一个给定的链接口（例如链接口 78）用来传送控制报文的特定通路是预先规定并贮存在一个控制通路寄存器 81 中。读出地址产生器 89 产生的读出地址，被传送到一个比较器 91 里，该比较器将这个读出地址与存在控制通路寄存器 81 中预置的控制通路编号作比较。当比较器判定一个瞬间读出地址跟这个控制通路编号相同时，它就产生一个开门信号传到控制字源寄存器 80 与一个控制字终点寄存器 92 上去。控制字终点寄存器响应比较器 91 的开门信号，将时分隔多路线 85 上的信息贮存下来。在这个特定通路中，时分隔多路线 85 上的信息包括控制单元 17 要利用的控制通路的内容。由于运用 DMA 单元 58，控制字寄存器 92 的内容，在下一个控制通路之前传送到存储器 57 里去。类似地，控制字源寄存器 80 响应比较器 91 来的开门信号，将其内容放入时分隔多路线 76，这样就发出了控制字。链接口 79 以基本上相同的方式发送和接收控制字，但是与链接口 79 结合的特定控制通路编号，却与链接口 78 结合的不同。

读出地址产生器 89 产生的读出地址也传送到一个帧序列产生器 93。帧序列产生器 93 对此作出响应，以每通路一个比特的速率产生一个唯一的帧比特序列。在每个通路中，帧序列产生器产生的比特被送到一个帧插入电路 94，该电路将帧比特放入从时隙 交换单元 11 来的数据字的 G 位位置。包括帧比特的数据字于是就经由一个并连——串连寄存器 95 和一个驱动器电路 96，送到时分隔多路线 13，而这条线是接到时分隔多路交换器 10 的一个唯一的输入口上。链接口 78 收到的每一个数据字上都包括由时分隔多路交换器 10 产生和发送的帧比特。一个帧校验器 97 读出时分隔多路交换器 10 送来的每个数据字的每个帧比特，以判定时分隔多路交换器 10 与它本身之间的

通信是否仍然同步。如果是同步的，就不需修正；但是如果发现不同步，就用本技术领域众所周知的方式，与时钟回收电路 84 通信，重新进行定帧。

时分隔多路交换器 10 的输入和输出端可以成对地考虑，因为两个端都跟同一个链接口交接。此外，时分隔多路交换器 10 的每一对输入和输出端，都接到一个与链接口 78 和 79 类似的时分隔多路交换器链接口上。链接口 78 接到一个时分隔多路交换器链接口 100（图 5），接口 100 包括一个接收器 101，该接收器 101 从时分隔多路线 13 接收数据字，并经过一条时分隔多路线 103 将那些数据字送到一个串连——并连寄存器 102。从时分隔多路线 103 来的比特流也被送到一个时钟回收电路 104 和一个帧检验电路 105，前者从中导出时钟信号而后者判定是否有帧同步。时分隔多路交换器链接口 100 还包括一个写入地址产生器 106，响应时钟回收电路 104 发来的信号，产生一序列写入地址。于是每个传送到串连——并连寄存器 102 的数据字，都按写入地址产生器 106 产生的地址写入一个随机存取存储器 107。

时分隔多路交换器 10 还包括一个时共享空间分隔交换器 108，它按每帧 256 个时隙、每时隙 约 488 毫微秒的帧速，接通输入和输出端间的通道。规定每个时隙内应该连接的输入和输出端之间的交换通道的控制信息，是存在控制存储器 29（图 2）内的，每个时隙中间这个存储器读出一次，以建立这些交换通道，记住每一个时隙有一个数码编号，而在一个给定的时隙内，有同一数码编号的数据字通路应被接上。因此，在具有一个给定数马编号的通路里的所有数据字，在它们结合的时隙内，必须都送入时共享空间分隔交换器 108 内，以避免不正确的交换。为了这个目的，时分隔多路交换器 10 内还包

括一个主时钟电路 109 用来产生一个重复出现的 256 个读出地址的序列，该序列基本上同时被送到各个时分隔多路交换器链接口。因此，随机存取存储器 107 以及包括在所有其他时分隔多路交换器链接口中的相应的随机存取存储器，在基本上同一时间读出与同一时隙结合的数据字。从随机存取存储器 107 读出的数据字被传送到并连——串连移位寄存器 111，从那里它们又被传送到时共享空间分隔交换器 108。

所有应由时分隔多路线 15 上传送到链接口 78 的数据字，都在它们传送到时共享空间分隔交换器 108 一个时隙 中，在导线 111 上从时共享空间分隔交换器 108 收到。时分隔多路交换器链接口 100 包括一个帧序列产生器 112，它以每时隙一比特的速率产生一序列帧比特。这些帧比特被传送到帧插入电路 113，电路 113 将帧比特放在导线 111 上的每个数据字的 G 位上。于是导线 111 上的每一个数据字都经过驱动器电路 114 由时分隔多路线 15 送到链接口 78。

集中式的路由

在系统 I 里，总的控制功能是由中央控制器 30 和交换模块内的控制单元，例如交换模块 201 内的控制单元 17，合作完成的。为了下面讨论的目的，交换模块控制单元完成的控制功能被简单地说成由交换模块完成的。系统的总处理任务被分解成为许多个称为程序进程的大任务。每个进程包括一组过程，每个过程都完成进程中的一些子任务。与进程结合的有一个存储器块，称为进程控制块，它存有可用于整个进程的数据，此外还有一个称为堆栈的存储器块，它贮存了进程中对个别过程有用的数据。进程通过报文互相通信。跟同一处理器的另一个进程通信，以及跟另一处理器内的另一进程通信，都用同一

类型的报文。

在系统 I 里，进程有两种类型：端进程与系统进程。只要系统在工作，系统进程就存在。另一方面，端进程只在个别呼叫或服务事项（例如诊断测试或服务品质估计）进行时期才存在。每一次呼叫都产生两个端进程——一个在连接于起点话线或干线的交换模块内的起点端进程，和一个在连接于终点话线或干线的交换模块内的终端端进程。作为一个例子，设想接到交换模块 229 的用户话机刚刚离钩。话线单元内的扫描发现了这个离钩状态。交换模块 229 内的一个呼叫处理控制系统进程 2001（图 7）被通知发现了这种离钩，作为反应，就建立了一个起点端进程 2002。起点端进程 2002 负责控制传送拨号音到用户话机 25，以及随后从用户话机 25 接受拨号。起点端进程 2002 分析拨号以取得四个变量：PI，DI，DLGCNT，及 TREAT 之值。变量 PI 是前缀指标，判定是否拨了一个前缀，如果拨了，是什么类型的前缀，例如人工长途呼叫的 0 + 前缀和直接长途呼叫的 1 + 前缀。变量 DI 是终点指标，判定呼叫的几个可能的终点类别之一，例如根据七位查询号的前三位（nxx 位），终点指标就可以决定终点是本地话线还是可经由本系统与某些其他交换系统连接的多个干线群之一接通的。变量 DLGCNT 只是定下拨号拨了几位。变量 TREAT 定出了拨号位数是否能处理而接通一个呼叫，或者象呼叫方只部分地拨出他所要的号码时，要给用户话机 25 发出一个适当的通知。此外，起点端进程 2002 根据起点话线的特性，例如它是否为典型的住宅话线，或者是接在一个专用小交换机（PBX）或键系统上。定下筛选指标 SI 的值。起点端进程 2002 于是在一个报文缓冲器中编一个路由要求报文 RTREQ。如图 14 所示，RTREQ 报文包括五个字段：PATHDES，RTGDATA，DLALDATA，GPI，和 TREAT。

(此处提到的其他数据结构，报文和关系一样，RTREQ 报文可能含有别的、对理解目前的说明并不重要的字段。)

PATHDES 字段贮存一个途径描述语，用来指定这次呼叫在交换系统内走的途径。这种途径可以用指定起点外围时隙，网络时隙和终点外围时隙的方式全部描述出来。起点外围时隙 是接收时隙交换器 50 (图 3) 从起点话线或干线收到信息、和发送时隙交换器 53 (图 3) 发送到起点话线或干线所用的 512 个时隙中的特定一个。类似地，终点外围时隙是用来跟终点话线或干线通信的 512 个时隙中的一个。网络时隙是起点交换模块中的接收时隙交换器 50 发出的 512 个时隙和终点交换模块中的发送时隙交换器 53 接收到的 512 个时隙中选出来的双方都可使用的时隙。为了建立途径的全程，必须同时在起点和终点交换模块的控制 RAM55 中 (图 3) 贮存规定时隙交换器应完成的外围时隙与网络时隙之间的映射的信息。模块内部的呼叫不经过时分隔多路交换器 10 来发送。但是，对于模块之间的呼叫，控制存储器 29 中存有指明在一次给定呼叫选用的时隙期间、时分隔多路交换器 10 必需提供一个从起点交换模块到终点交换模块的途径的信息。在此例中，起点终端进程 2002 在这个时候只知道呼叫的起点外围时隙。PATHDES 字段中其余部分是空白的。

RTG DATA 字段被用于贮存某些对目前描述的理解并不重要的实现呼叫处理功能所需要的变量，这些此地就不再提起了。RTG DATA 字段也存有以后用于定出这次呼叫的终点类别，也就是话线，干线还是宣告终点的一个变量 TERMTYP。DLALDATA 字段用来贮存由起点端进程 2002 决定的。也由收到的拨号位数决定的变量 PI, DI, SI, 和 D_GCNT。GPI 字段用来贮存接到起点用户话机的口的全局口身份。一条给定话线或干线接到图 2 中交换系统的一点上、此处称

为一个口。（在多通路数字设备中，每条通路都认为是接到一个不同的口上）。系统中每一个口都有一个唯一的全局口身份。数字服务单元 67（图 3）有许多宣告电路，每条都有它唯一的全局口身份。对于接在合用线上的口，GPI 字段也识别了这些线上的每个用户的身份。在 RTREQ 报文中，GPI 字段定出了起点口的全局口身份。TREAT 字段是用来贮存起点终端进程 2002 定下的 TREAT 变量。

一旦 RTREQ 报文编成了，它就被起点终端进程 2002 送到中央控制器 30 里面的一个路由系统进程 2003（图 7）里去。路由系统进程 2003 将 RTREQ 报文存在一个称为路由数据块 (RDBLK) 2101 的数据结构里。路由系统进程 2003 按 RTREQ 报文中的信息，以本文将详述的一种方式去访问一个集中的数据库，去决定终点口的全局口身分。路由系统进程 2003 也选用一个供这次呼叫用的、可用的网络时隙，而且如果终点口是接到一个与起点口不同的交换模块上，就将决定选用的时隙的信息写入控制存储器 29。路由系统进程 2003 于是根据 TERMTYP 变量的值，在一个报文缓冲器内编一个话线终点要求 (LNTREQ) 报文，一个干线终点要求 (TKTREQ) 报文，或者一个宣告终点要求 (ANTREQ) 报文。图 14 中表示出 LNTREQ 报文包括四个字段：PATHDES，RTGDATA，FARP D 和 GPI。

PATHDES 和 RTGDATA 字段在前面讨论 RTREQ 报文时已经提到。但是路由系统进程 2003 定出来的网络时隙也加进了 PATHDES 字段。FARP D 字段用来贮存决定起点终点进程的进程身分标志，在此例中，就是从 RTREQ 报文的报头中定的起点端进程 2002。GPI 字段贮存由路由系统进程 2003 决定的终点口的全局口身分。当终点口是接到一个干线或一个宣告电路时，就编定一个 TKTREQ 报文或一个 ANTRREQ 报文。图 14 表示出，TKTREQ 报文和 ANTRREQ 报文都含

有与 LNTREQ 报文相同的字段，此外，TKTREQ 报文还包括贮存经由这条干线发送到另外一个交换系统去的数字的 D GDATA 字段。在这个例子中，假定由路由系统进程 2003（图 7）定出来的终点口是接用户话机 23 的。在报文缓冲器中编出来的 LNTREQ 报文就由路由系统进程 2003 传送到交换模块 201 上的终点系统进程 2004 上去。进程 2004 作出响应，读一个存在交换模块 201 内的一个忙／闲映射表（此后面称为 PORT STATUS 关系）去判定用户话机 23 目前是忙还是闲。如果用户话机 23 是闲的，进程 2004 就建立一个终端进程 2005，并经过一个话线终点（LNTTERM）报文将从 LNTREQ 报文中收到的信息转送到进程 2005 里去（或者，如果收到的是 TKTREQ 报文或者是 ANTREQ 报文，则经过一个干线终点（TKTERM）报文或一个宣告终点（ANTERM）报文将信息转入进程 2005）。终端进程 2005 进行传送振铃电压到用户话机 23，并传送 E 位连续性信号，此内容在前述美国专利 4,322,843 号中曾有说明，以及将能听到的振铃音送到交换模块 229 里去。终端进程 2005 于是发送一个包括现在已经完备的途径描述语 PATHDES 的接线完成（SETUPCOMP）控制报文到交换模块 229 内的起点端进程 2002。作为响应，起点端进程 2002 将 E 位连续性信号送到交换模块 201。当交换模块 201 收到交换模块 229 送来的 E 位连续性信号，终端进程 2005 就决定和用户话机 23 通信的终点外围时隙，并在交换模块 201 的控制 RAM 55 写入规定终点外围时隙与网络时隙之间的映射信息。类似地，当交换模块 229 收到了交换模块 201 送来的 E 位连续性信号时，起点端进程 2002 就决定跟用户话机 25 通信的起点外围时隙，并在交换模块 229 的控制 RAM 55 写入规定起点外围时隙与网络时隙之间的映射信息。用户话机 25 与 23 之间的通信途径现在已被接通了。

回忆一下前面描述的例子中路由系统进程 2003 完成了三项基本功能——决定终点口和它的全局口身分，选一个可用的网络时隙，并对模块间的呼叫，建立经过时分隔多路交换器 10 的途径，即，在控制存储器 29 内写入规定所选用的时隙信息。在图 9 到 13 内画出了路由系统进程 2003 在执行这些功能时进行的路由程序的流程图。图 8 画出了指明路由系统进程 2003 的操作状态的状态图。在现在描述的系统 I 中，单个路由系统进程 2003 对系统内所有的呼叫执行了终点口决定与网络时隙选定的任务。路由系统进程 2003 还对所有模块间的呼叫执行建立时分隔多路交换器 10 途径的功能。路由系统进程 2003 在一个时间内处理一个呼叫——即，它以执行路由程序去建立一个 LNTREQ 报文或一个 TKTREQ 报文或一个 ANTREQ 报文的方式，响应每个 RTREQ 报文。图 7 中表示出，路由系统进程 2003 在程序执行时用了四种数据结构——一个路由数据块 (RDBLK) 2101，一个呼叫流程块 (CFBLK) 2102，一个群块 (GRPBLK) 2103 和一个终点块 (TERMBLK) 2104。路由系统进程也能对由 12 个关系，2105 到 2116 组成的集中数据库进行存取，有关内容下面将予以说明。

关系数据库被认为是一组关系，如 C. J. Date 所著的数据库系统引论 (An Introduction to Database System) 第三版 (Addison-Wesley 公司 1981 年出版) 上所说的那样。一个关系可以认为是一个长方形的表。表中的行称为元组，而列则是有唯一名称的一些属性。一个特定的元组中的一个有名称的属性称为一个项。一个键是属性的一个子集，它的值是用来唯一地判定关系中一个元组的身分的。一个键如果由多于一个属性组成的称为复合的。偶然地，一种关系有不止一个候选选用的键。在那种情况下，候选的属性中的一

个被指定为关系的主键。每个属性都能采用规定的一组值，称为这个属性的值域。表 1 列出了一个称为部件的作为说明例子的关系。

表 1

关系 部 件				
P#	P'名称	颜色	重量	城市
P1	螺母	绿	13	阿姆斯特丹
P2	螺栓	红	18	特拉维夫
P3	螺栓	蓝	18	罗 马
P4	螺丝	蓝	15	伦 敦
P5	凸轮	黄	13	巴 黎
P6	嵌齿	黑	20	罗 马

属性 P# 是关系的主键，因为指定了它的值就可以唯一地判定关系中一个元组的身份。例如，指定 P# = P4 就定出了元组 (P4, 螺丝, 蓝, 15, 伦敦)。

在收到一个 RTREQ 报文时，路由程序（图 9 到 13）的执行从 START 状态 3001（图 8）开始。在方块 1010（图 9）期间，收到的 RTREQ 报文存入路由数据块 RDBLK（图 15）的前两个字段—— HEADER 字段和 TEXT 字段——之内。报文的报头被分析，起点端进程的进程身分被存在 RDBLK 的 ORGTP1 字段之内。RDBLK 的 RTGSTATE 字段判定路由程序目前在图 8 的状态图中的哪一个状态内。RTGSTATE 字段被更新着使在每一个状态转换发生前定出下一

个状态。 RICOUNT 字段用于干线路由，文中以后将说明。

执行首先进行到方块 1025 (图 9)，在此期间，许多程序变量都根据 RTREQ 报文初始化了。下一步到判定块 1030，那时要判定 RTREQ 报文中变量 TREAT 是否指定一个固定路由的要求，例如指路到一条宣告线，通知起点用户说号码只拨了一部分。如果变量 TREAT 指定固定路由，执行就进入方块 1240，那时就进入了 FIXEDRRT 状态 3002 (图 8)。一个初始化了的变量就是 TERMTYP，它定下了要求的终点是一个宣告电路。用 TREAT 作为键，读出 FIXEDRI 关系 (图 16) 以取得随后将用于寻找合适的宣告电路的路由指标 (RI)。键 TREAT 定出的 FIXEDRI 关系元组，存入呼叫流程块 CFBLK (图 15)。

但如果变量 TREAT 没有定固定路由，执行就从方块 1030 进入方块 1040，那时就进入 SCREEN 状态 3003 (图 8)。在方块 1050 中用变量 DI，SI 和 PI 作为复合键，读出 SCRNING 关系 (图 16)。SCRNING 关系含有属性 RI，NOC 和 ROUTETYPE。ROUTETYPE 属性现定了终点口是接到话线还是干线。如果是话线，NOC 属性就规定了终点口的标准局码。标准局码代表一个七位查询码的前三位 (nxx) 的编码。例如一个典型的中央局的 nxx 号码 355，357 和 420，可能被编为标准局码 1，2 和 3。如果 ROUTETYPE 属性规定是干线，RI 属性就规定一个进入 ROUTING 关系 (图 16) 的路由指标，ROUTING 关系随后被读出以取得一个特定的干线组的号码。从 SCRNING 关系中由复合键 DI，SI 和 PI 规定而读出的元组，被存在 CFBLK (图 15) 里。

执行进行到决策块 1060，那时就检查了 ROUTETYPE 属性来决定要求的终点是话线还是干线。如果 ROUTETYPE 属性规定为话线，

执行就进到方块 1070，并进入 DNTRAN 状态 3004（图 8）。

TERM TYPE 变量就被设置来定出要求的终点是话线。记住起点用户话机的拨号是作为 RTREQ 报文的一部分传过来的，而标准局码（NOC）则是从 SCRNING 关系读得的。为了查询号码翻译而存在中央控制器 30 里的查询号码（DNs），不是按七位数码而是作为五位数码存储的，这五位是由一位的 NOC 与后四位拨号所组成。将 SCRNING 关系得来的 NOC 与 RTREQ 报文中最后四位拨号连成的 DN 被用为读 DNTRAN 关系（图 16）的键。DNTRAN 关系中包括一个 TERMCLASS 属性，它规定由键定出来的线是一个独立的线还是属于一个多线寻线组的一部分，并定出分辨这条线的全局口身份的 GPI 属性。由键 DN 规定的 DNTRAN 关系的元组，存在 CFBLK（图 15）里面，而执行进入决策块 1090。

在方块 1090 中，根据 TERMCLASS 属性，判定指定的线是个别的线还是多线寻线组中的一部分。如果标明是单个的线，决定终点口的功能就已经完成，而执行到方块 1180，那时终点口的 GPI 就被存在 TERM_BLK（图 15）里。注意，GPI 包括两个字段——判别哪一个交换模块包括终点口的 MODULE 字段，以及在那个交换模块的口中判别一个特定的口的 PORT 字段。

执行过程进到方块 1190 就进入了 INTEGRITY 状态 3012（图 8）。中央控制器 30 周期地跟每个交换模块的控制单元通信，去证实它们的操作状态，并在一张状态表上保留这种状态信息。在方块 1190 期间，用 TERM_BLK 中存储的 MODULE 字段读这个状态表，用来证实所定交换模块中的处理器是在正常工作的。在方块 1200 中，进入了 NWCONN 状态 3013（图 8）。在方块 1200 期间，选定可用的网络时隙，如果呼叫是模块间的呼叫，规定所选时隙的指令就存在控制存

储器 29 内。在方块 1210 时，用 TERMTYP 变量去判定要在报文缓冲器编一个 LNTREQ 报文，一个 TKTREQ 报文还是一个 ANTREQ 报文。在方块 1220 期间，就用 RDBLK，CFBLK，和 TERM_BLK 数据结构中的数据建立合适的报文。执行于是进到方块 1230，那时报文缓冲器中存的报文被送到终点交换模块的控制单元，然后进入 DONE 状态 3014（图 8）。

回到决策块 1090，如果 TERMCLASS 属性不是规定独立的线而是规定一个多线寻线组，执行就从方块 1090 到了 1100。一个多线寻线组是一组同用一个查询号或一组查询号码的线。在方块 1100 期间，用从 DNTRAN 得来的 GPI 为键，读出 PORTGROUP 关系（图 16）。PORTGROUP 关系包括规定多线寻线组的号码的 GRPNUM 属性，以及规定一给定组的特定成员的 MEMBER 属性。从 PORTGROUP 关系中读出的元组存入 GRPBLLK（图 15），执行进到方块 1110 而进入了 MLGRREHUNT 状态 3005（图 8），在方块 1110 期间，用 GRPNUM 属性作为键，读出 MHG 关系（图 17）。MHG 关系中包括 HTYPE 特性，它规定了贮存多线寻线组的动态忙／闲数据的许多关系之一，目前作为例子来讨论，只提到 LNSTAT 关系（图 17）。从 MHG 关系读出的元组，存入 GRPBLLK（图 15）。给定 HTYPE 属性规定了 LNSTAT 关系，执行就进到 1120 方块。在方块 1120 期间，以 GRPNUM 属性为键、读 LNSTAT 关系。LNSTAT 关系包括分辨寻线组中每个成员的忙／闲状态的 GMFLAG 比特映射。不贮存整个映射而只将一个指向这个映射的指针存入 GRPBLLK（图 15），执行转到方块 1140，进入 MLGHUNT 状态 3006（图 8）。在方块 1140 期间，选取寻线组中的一个闲的成员。通过贮存的指针可以访问的 GMFLAG 比特映射表，被用来决定闲的成员。选择是按照一个事先拟

定的、基于 HTYPE 属性的寻线算法。执行推进到方块 1150，那时就判定了在方块 1140 期间进行的寻线是否已经成功地找到了一个闲的寻线组 MEMBER。如果找不到这样的 MEMBER，执行就推进到方块 1160，进入 MLGBUSY 状态 3007（图 8），而呼叫失败。但如果找到了一个闲 MEMBER，执行就推到方块 1170，并用这个闲 MEMBER 和 GRPNUM 属性作为键去读 GROUPOPORT 关系（图 16）而取得终点口的 GPI。决定终点口的任务现在已经完成，执行推进到方块 1180，然后象前面所说的那样按方块 1190, 1200, 1210, 1220 和 1230 逐步推进下去。

回到决策块 1060，如果 ROUTETYPE 属性没有规定话线而是规定了干线，执行就从方块 1060 推进到方块 1250。变量 TERM TYPE 是这样设置着的，如果方块 1250 是从方块 1060 那边过来的，要求的终点就是干线。方块 1250 也是可以从方块 1240 那边过来的。在方块 1250 期间，进入了 RTING 状态 3008（图 8），RDBLK 中的 RICOUNT 变量增加一，执行转入决策块 1260。在方块 1260 期间要判定为了接通呼叫而试试路由指标（RI s）是否已超过一个固定数目——四次。如已经起过，执行转到方块 1270 而呼叫失败。但如果只试四个 RI 或少于四个，执行就转到方块 1280，那时就用 RI 为键去读 ROUTING 关系（图 16）。ROUTING 关系含有 GRPNUM 和 SECRI 两个属性，GRPNUM 属性是一个特定的干线组的号码，SECRI 属性是一个二次路由指标，这是在呼叫不能由规定的干线组接通时使用的。从 ROUTING 关系取得的元组存入 CFBLK（图 15）。

执行转到方块 1290，并进入 TRKPRESHUNT 状态 3009（图 8）。以 GRPNUM 为键，读 TRKG 关系（图 18）。TRKG 关系包括规定这一组应使用的寻线方式的 HTYPE 属性。读出的 TRKG 元组存入

GRPBLK (图 15)，执行转到方块 1300，那时要判定 HTYPE 属性规定的是一个先进先出 (FIFO) 组，一个旋转组，还是一个前进／后退组。在目前的例子里，寻线组或者在单向外岀的干线的情况下是 FIFO 组，或者在宣告电路的情况下是旋转组，或者在双向干线的情况下是前进／后退组。在 FIFO 寻线组中，干线是按照它们闲下来的次序来指定的。在旋转寻线组中，宣告电路是在旋转的基础上被指定的，所以它们的使用是等量分布的。在前进／后退寻线组中，一个给定的交换系统总是在组表的最前端开始寻找闲组员的，而接在干线另一端交换系统总是从组表的尾端开始寻找闲组员的，这样就减少了顶牛的可能性。如果 HTYPE 规定了一个 FIFO 组或一个旋转组，执行就进到方块 1310，并进入 TRKHUNT 状态 3010 (图 8)。先读一下 TKOWNER 关系 (图 18) 以取得属性 QKEY，然后以它为键去读 TKQUE 关系 (图 18)。TKQUE 关系包括 GPI 属性，它规定了此次呼叫要用的空闲组员的全局口身份。TKQUE 关系还含有 NIM 属性，它规定了下一次 TKQUE 关系被访问时可用的下一个空闲组员。指向 TKOWNER 关系和 TKQUE 关系中的元组的指针，存入 GRPBLK (图 15)。

回到决策块 1300，如果 HTYPE 规定一个前进／后退组，执行就进到方块 1330，那时就以 GRPNUM 为键去读 TKSTAT 关系 (图 18)。TKSTAT 关系包括一个规定干线组中每个成员的忙／闲状态的 GMFLAG 比特映射表。不存整个比特映射而只将指向那个比特映射的一个指针存入 GRPBLK (图 15)，执行转到方块 1350，进入 TRKHUNT 状态 3010 (图 8)。在方块 1350 期间，按予订的前进／后退算法，从经由存入的指针可以访问的 GMFLAG 比特映射表，选择干线组的一条空闲组员。以 GRPNUM 与选出的空闲 MEMBER 为复

合键，读出 GROUPOPORT 关系（图 16）以定出 GPI。

在完成了方块 1350 或方块 1310 之后，执行转到决策块 1370，那时要判明在方块 1350 或 1310 期间的寻线，已否找到一条空闲的干线组 MEMBER。如果没有找到这样的 MEMBER，执行就进到方块 1380，进入 TRKBUSY 状态 3011（图 8）。使从 ROUTLNG 关系读的二次路由指标（SECRI）成为下一个路由指标（RI），执行就回到方块 1250。但如果找到一根空闲 MEMBER，决定终点口的任务已经完成，执行就进到方块 1180，并如前述逐一通过方块 1190，1200，1210，1220 和 1230。

系统 II

图 19 到 21，当按照图 22 排列时，表示一个具有远程交换能力的时分隔交换系统。这个系统，本文中称为系统 II，包括一个主交换系统 800（图 19 与 20）和四个单独的远程交换模块 501，502，503 与 504（图 21）。主交换系统 800 包括上述图 2 的时分隔交换系统，以及两个主交换模块 301 和 302，模块 301 接到时分隔多路交换器 10 的输入／输出端对 P59 与 P60，模块 302 接到输入／输出端对 P61 与 P62。在这个实施方案中，每个远程交换模块都经过四条双向数字传送设备，例如美国专利 4,059,731 号发表的 T1 载波系统，接到一个主交换模块。具体地说，主交换模块 301 通过传送设备 421 到 424 接到模块 501，通过传送设备 431 到 434 接到模块 502，主交换模块 302 通过传设备 441 到 444 接到模块 503，通过传送设备 451 到 454 接到模块 504。主交换模块 301 含有一个时间片交换单元 311 和一个相结合的控制单元 317，它们分别与时隙交换单元 11 及控制单元 17 基本上相同。时隙交换单元 311 通过

接入时分隔多路交换器 10 的输入／输出端对 P59 与 P60 的两条 256 路时分隔多路线发送并接收信息。输入／输出端对 59 上的控制通路 59 和输入／输出端对 60 上的控制通路 60 被用于传递控制单元 317 P46 与控制分配单元 31 之间的控制报文。接口时隙交换单元 311 与传送设备 421 到 424，431 到 434 的数字设备接口 321 到 328，基本上是完全相同的。数字设备接口 321 在前面提到的美国专利申请书 493,683 号上有详细说明。

主交换模块 302 由时隙 交换单元 312，控制单元 318 和数字设备接口 331 到 338 组成，它跟模块 301 基本上完全一样。控制单元 318 跟控制分配单元 31、用输入／输出端对 P61 的控制通路 61 以及输入／输出端对 P62 的控制通路 62 来交换控制报文。

四部远程交换模块 501 到 504 基本上全是一样的。每部远程交换模块都包括一个设备接口单元，它在目前这个实施方案中和来自一个主交换模块的四条数字传送设施交接。举例说，远程交换模块 501 (图 21) 包括一个设备接口单元 505，它和来自主交换模块 301 的设施 421 到 424 交接。设施接口单元 505 将从所连接的四条传输设施上收下的信息加以多路调制并发送到连在时隙交换单元 511 上的一对 256 路时分隔多路线 515 与 516 上的预定通路上去，并将经一对 256 路时分隔多路线 513 与 514 从时隙交换单元 511 收到的信息、多路解调并发送到四条传输设施的预定通路上去。设施接口单元 505 在前面提到的美国专利申请书 493,683 号上有详细说明。远程交换模块 501 包括一个跟时隙交换单元 511 结合的控制单元 517 以及多个外围单元，例如对话机 528 与 529 那种用户话机服务的话线单元 519 与 520 和接到干线 543 与 544 上的干线单元 539 与 540。时分隔多路线 513 到 516，时隙交换单元 511，控制单元 517，话线

单元 519 与 520，用户电话 528 与 529，干线单元 539 与 540 和干线 543 与 544 之间的关系，基本上同时分隔多路线 13 到 16，时隙交换单元 11，控制单元 17，话线单元 19 到 20，用户电话 23 与 24，干线单元 39 与 40 和干线 43 与 44 之间的一样。

在目前的实施方案中，交接一个给定的远程交换模块，例如 501，和主交换模块 301 的四条传输设施中的两条中的通路 1，被定为一条控制通路。因此，在四部远程交换模块 501 到 504 与控制分配单元 31 之间，有八条控制通路。时隙交换单元 311 从传输设施 421 到 424 和 431 到 434 接到的四条控制通路，由输入／输出端对 P59 的通路 63 与 64 以及输入／输出端对 P60 的通路 65 与 66，传送到时分隔多路交换器 10。类似地，时隙交换单元 312 从传输设施 441 到 444 和 451 到 454 接到的四条控制通路，由输入／输出端对 P61 的通路 67 与 68 以及输入／输出端对 P62 的通路 69 与 70，传送到时分隔多路交换器 10。中央控制器 30 向控制存储器 29 写入必要的指令，使输入端 P59 的通路 63 与 64，输入端 P60 的通路 65 与 66，输入端 P61 的通路 67 与 68 和输入端 62 的通路 69 与 70，总是经由输出端 P64 传送到控制分配单元 31，并使输入端 P64 的通路 63 与 64 向输出端 P59 发送，输入端 P64 的通路 65 与 66 向输出端 P60 发送，输入端 P64 的通路 67 与 68 向输出端 P61 发送，而输入端 P64 的通路 69 与 70 向输出端 P62 发送。在此实施方案中，控制分配单元 31 必须在输入输出端对 P64 容纳 256 条可能控制通路中的 70 路，而不是系统 I 的控制分配单元的仅仅只有 58 路。

虽然远程交换模块与主交换模块之间的控制通信的主要模式是经由上述的时分隔多路交换器 10 与控制分配单元 31 的控制通路，控制通信也按前面提到的美国专利申请书 493,683 号所描述的方式进行，

用这些传输设施上的所谓导出的数据链，例如 421 到 424 上的。导出数据链在美国专利 4,245,340 号有说明。

分 布 式 路 由

与路由功能由中央控制器 30 集中地完成的系统 I 对比，在系统 II 中，路由功能是分布到远程交换模块 501 到 504 中去的。回忆一下，在系统 I 中，只有中央控制器 30 才有一个路由系统进程，进程 2003 (图 7)，以及与其结合的数据结构，RDBLK 2101，CFBLK 2102，GRPBLK 2103，与 TERMBLK 2104，和集中的数据库包括 FLXEDRI 关系 2105，SCRNLNG 关系 2106，DNTRAN 关系 2107，ROUTLNG 关系 2108，PORTGROUP 关系 2109，GROUPOPORT 关系 2110，MHG 关系 2111，LNSTAT 关系 2112，TRKG 关系 2113，TKOWNER 关系 2114，TKQUE 关系 2115 与 TKSTAT 关系 2116。在系统 II 中，中央控制器 30 也类似地有一个路由系统进程，进程 3603 (图 23)，以及有关的数据结构，RDBLK 3101，CFBLK 3102，GRPBLK 3103 与 TERMBLK 3104，以及数据库包括 FLXEDRI 关系 3105，SCRNLNG 关系 3106，DNTRAN 关系 3107，ROUTLNG 关系 3108，PORTGROUP 关系 3109，GROUPOPORT 关系 3110，MHG 关系 3111，LNSTAT 关系 3112，TRKG 关系 3113，TKOWNER 关系 3114，TKQUE 关系 3115 与 TKSTAT 关系 3116。但是此外，每个远程交换模块都有路由系统进程和有关的数据结构与数据库。例如，远程交换模块 501 有路由系统进程 3602 (图 23)，有关的数据结构 RDBLK 3201，CFBLK 3202，GRPBLK 3203 与 TERMBLK 3204，以及数据库包括 FLXEDRI 关系 3205，SCRNLNG 关系 3206，DNTRAN 关

系 3207，ROUTLNG 关系 3208，PORTGROUP 关系 3209，
GROUPOPORT 关系 3210；MHG 关系 3211，LNSTAT 关系 3212，
TRKG 关系 3213，TKOWNER 关系 3214，TKQUE 关系 3215，与
TKSTAT 关系 3216。远程交换模块 502，503 与 504 每个都类似
地有一个路由系统进程和有关的数据结构与数据库。在系统中，
RDBLK 数据结构包括一个 RTSEG 字段与一个 SWRANK 字段，而 LBL
与 TRKG 关系每个都包括 MODULE 字段如图 3.2 所示，本文以后将作
说明。在此实施方案中，中央控制器 30 与远程交换模块 501 到 504
中 FIXEDDRI，SCRNLNG，ROUTLNG，MHG 和 TRKG 关系都有冗
余。对于这些关系中的每一个，所有系统中的有关数据，既存入中央
控制器 30，也同样存入远程交换模 501 到 504 中的每一个。中央控
制器 30 中的 DNTRAN 关系 3107 贮存了所有接到这个系统的线的查
询号翻译信息。但是每个远程交换模块中的 DNTRAN 关系，例如远程
交换模块 501 中的 DNTRAN 关系 3207，只存入接到那个远程交换
模块的线路的查询号翻译信息。类似地，中央控制器 30 的 PORTGROUP
关系 3109 与 GROUPOPORT 关系 3110，存入了所有交换系统口的组
翻译信息。每个远程交换模块中的相应关系，例如远程交换模块中的
PORTGROUP 关系 3209 与 GROUPOPORT 关系 3210，就只存入那
个远程交换模块中的口所需要的这种信息。每个远程交换模块中用以
贮存多口寻线组的动态忙／闲数据的关系，例如远程交换模块 501 中
多线寻线组的 LNSTAT 关系 3212 以及干线组的 TKOWNER 关系
3214，TKQUE 关系 3215 与 TKSTAT 关系 3216，只存入所有它
们的话线或干线都接到那个远程交换模块的那些组的数据。在系统中
所有其他多口寻线组的动态数据都存入中央控制器 30 的 LNSTAT 关
系 3112，TKOWNER 关系 3114，TKQUE 关系 3115 与 TKSTAT

关系 3116。MHG 与 TRKG 关系(图 32)的 MODULE 字段，对每个多口写线组，定出了远程交换模块 501 到 504 之一或中央控制器 30，作为那一组的动态数据的位置。

所有路由系统进程，例如 3603 与 3602，都执行同一个路由程序，它的流程图可见图 25 到图 29。与那些路由系统过程有关的状态图见图 24。

作为第一个例子，设想用户话机 528 刚刚离钩。话线单元 519 内的扫描发现了这个离钩状态。远程交换模块 501 内的一个呼叫处理控制系统进程 3601(图 23)被通知探测的这种离钩情况，然后作为响应，建立一个起点端进程 3604。起点端进程 3604 负责控制传送拨号音给用户话机 528 及随后接收用户话机 528 拨出的号码。起点端进程 3604 分析拨号以取得前缀指标(PI)，终点指标(DI)，拨号位数(DLG CNT)与处理(TREAT)变量值。起点端进程 3604 根据起点线的特性定出筛选指标(SI)值。起点端进程 3604 于是在报文缓冲器内编写一个路由要求报文 RTREQ。这个 RTREQ 报文(图 14)已在前面叙述系统 I 时作过说明。

一旦编定 RTREQ 报文，就被起点端进程 3604 传递到仍在远程交换模块 501 内的路由系统进程 3602(图 23)。路由系统进程 3602 将 RTREQ 报文存入 RDBLK 3201。路由系统进程 3602 用 RTREQ 报文中的信息去访问与它结合的数据库。假定在这个例子中用户话机 528 拨出来的号代表也接在远程交换模块 501 内的用户话机 529 的查询号码。因此，DNTRAN 关系 3207 包括必要的查询号翻译信息，因为终点口与起点口同在一部远程交换模块上。在这种情况下，路由系统进程 3602 能完成终点口的决定。路由系统进程 3602 也选一个时隙，该时隙为交换单元 511 的接收时隙 交换器与发送

时隙交换器都可以使用的，且用来将起点外围时隙接到终点外围时隙。路由系统进程 3602 于是在报文缓冲器内根据 TERMTTP 变量编订一个话线终点要求 (LNTREQ) 报文，一个干线终点要求 (TKTREQ) 报文或一个宣告终点要求 (ANTREQ) 报文。其中每个报文都已在图 14 表示出来，并在文中作了说明。在此例中，编定的是 LNTREQ 报文。LNTREQ 报文中的 PATHDES 字段包括接收时隙交换器与发送时隙交换器之间选用的呼叫时隙的规定。在报文缓冲器里编定的 LNTREQ 报文被路由系统进程 3602 发送到一个终点系统进程 3606。作为响应，进程 3606 读出存在远程交换模块 501 内的忙／闲映射，来判定用户话机目前是忙还是闲。如果用户话机 529 是闲的，进程 3606 就建立一个终端进程 3605，并通过一个话线终点 (LTERM) 报文（或如果收到了一个 TKTREQ 报文或一个 ANTREQ 报文就用干线终点 (TKTERM) 报文或宣告终点 (ANTERM) 报文）将从 LNTREQ 报文收到的信息送到进程 3605。终端进程 3605 进行传送振铃电压到用户电话 529，并传送可听到的振铃音返回到用户电话 528 里去。终端进程 3605 于是发送一个 SETUPCOMP 报文到起点端进程 3604，其中包括现在已完备的途径描述语 PATHDES。起点端进程 3604 与终端进程 3605 将信息写入时隙交换单元 511 的控制 RAM，分别说清了起点外围时隙与路由系统进程 3602 选定的共用时隙之间的映射，以及终点外围时隙与所选共用时隙之间的映射。用户话机 528 与 529 之间的通信途径，现已接通。

一个很类似于刚才对图 23 描述的情况，也可应用到对远程交换模块 501 内部控制的多口寻线组的呼叫，例如，对所有组员都接到远程交换模块 501 的话线或干线组。

作为第二个例子，假定用户话机 528 拨的号仍和以前一样代表用

户话机 529 的查询号，但是用户话机 529 属于不是由远程交换模块 501 控制而由中央控制器 30 控制的一个多线寻线组的一部分。跟上面一样，呼叫处理控制系统进程 3601（图 30）被通知发现离钩，并建立一个起点端进程 3611。起点端进程 3611 于是发送一个 RTREQ 报文给路由系统进程 3602，它就将收到的 RTREQ 报文存入 RDBLK 3201。路由系统进程 3602 于是执行它的路由程序（图 25 到 29）。当程序执行达到访问 LNSTAT 关系 3212 这一点上，找不到包括用户话机 529 的多线寻线组的判定忙／闲状态的动态数据。因此就在报文缓冲器内编一个通用路由报文 RTGEN（图 32）。RTGEN 报文包括 PATHDES，RTGDATA 与 ORLGGPI 字段，这些都在以前讨论 RTREQ 报文时解释过的。RTGEN 报文还包括一个 REQTERM 字段，该字段规定了当路由被下一个处理器继续进行时、应进入路由程序的状态。此外，RTGEN 报文还包括 RTCONTDA 字段，它定下那些已经被路由系统进程 3602 判明的一些变量值，例如存入 CFBLK 3202 的变量，使在路由继续时不致重复不必要的工作。此外，RTGEN 报文也包括一个 ORLGGPI 字段和一个 TERMGPI 字段，它们分别存入起点口的全局口身分与终点口的全局口身分。当然，不能填入 TERMGPI 字段，除非等到决定终点口的任务已经完成。RTGEN 报文被送到中央控制器 30 的路由系统进程 3603，它在 REOTERM 字段规定的点上进入它的路由程序。从 RTGEN 报文传来的信息存入 RDBLK 3101 和 CFBLK 3102 的适当的字段量。由于包括用户电话 529 的多线寻线组的判别忙／闲状态的动态数据是在 LNSTAT 关系 3112 里面，路由系统进程 3603 就能完成决定终点口的任务。假定接在交换模块 201 上的用户话机 23 是与用户话机 529 处于同一多线寻线组内，而作为路由系统进程 3603 寻线的结果，用户话机 23

被指定给这次呼叫。路由系统进程 3603 选一个可用的网络时隙 给这次呼叫用，而且由于终点口是接在与起点口不同的交换模块上，就将判明选用的时隙的信息写入控制存储器 29。路由系统进程 3603 于是发送一个 RTGEN 报文，其中 PATHDES 字段中包括所选的网络时隙，并包括一个已完备的 TERMGPI 字段，该报文被送到交换模块 201 的一个终点系统进程 3610。作为响应，进程 3610 读一个存入交换模块 201 的忙／闲映射，以判定用户话机 23 目前是忙还是闲。如果用户话机 23 目前空闲，进程 3610 就建立一个终端进程 3612，并经由一个 LTERM 报文将 RTGEN 报文中的信息转送到进程 3612。终端进程 3612 进行传送振铃电压给用户话机 23，并传送一个 E 位连续性信号及可闻的振铃音到主交换模块 301。终端进程 3612 于是给远程交换模块 501 的起点端进程 3611 发一份 SETUPCOMP 报文。作为响应，起点端进程 3611 在传输设施 421 到 424（图 20）之一（例如 421）上选一个时隙 给这次呼叫用，并与主交换模块 301 进行控制通信使时隙 交换单元 311 将传输设施 421 上选定的呼叫时隙接到时分隔多路交换器 10 选定的网络时间片上。这种用主交换模块 301 的控制通信在前面提到的 Chodrow 等专利申请书 493,683 号中已有说明。一旦交换模块 201 的 E 位连续性信号经过主交换模块 301 被远程交换模块 501 收到，起点端进程 3611 就在时隙交换单元 511 的控制 RAM 中写入信息，规定起点外围时隙与传输设施 421 上选定的呼叫时隙 之间的映射。类似地，一旦交换模块 201 收到了 E 位连续性信号，终端进程 3612 就在时隙交换单元 11 的控制 RAM 55 上写下信息，规定终点外围时隙与网络时隙之间的映射。用户话机 528 与 23 之间的通信路径现在就建立了。

一个非常类似于刚才对图 30 描述的情景，也适用于从远程交换模块 501 到不是接在远程交换模块 501 上的单个话线上的呼叫，也适用于对不受控于远程交换模块 501 的干线组的呼叫。

作为第三个例子，设想接到交换模块 201 的用户话机 24 刚好离钩。呼叫处理控制系统进程 3609（图 31）被通知发现离钩，并作为响应，建立一个起点端进程 3621。起点端进程 3621 分析用户话机 24 的拨号以获得 PI, DI, DLGCNT 与 TREAT 之值，并根据起点线的特征定出 SI。注意，交换模块 201 没有一个路由系统进程。因此，起点端进程 3621 就发送一个 RTREQ 报文到中央控制器 30 中的路由系统进程 3603。路由系统进程 3603 将 RTREQ 报文存入 RDBLK 3101 并开始执行路由程序（图 25 到 29）。假定用户话机 24 拨的号要求用一组全部接入远程控制模块 501 的干线，例如干线 543 与 544。再假定干线组是一个先进先出（FIFO）组。由于该组受远程交换模块 501 而不是受中央控制器 30 的控制，判定干线组中干线的忙／闲状态的动态数据不在中央控制器 30 的 TKOWNER 关系 3114 与 TKQUE 关系 3115 里。因此，当路由程序的执行达到要访问 TKOWNER 关系 3114 与 TKQUE 关系 3115 的一点时，因为取不到所要的数据，就编出一个 RTGEN 报文。TRKG 关系 3113 的 MODULE 字段指出这一组的动态数据是在远程交换模块 501 里。路由系统进程 3603 选这次呼叫用的、通过时分隔多路交换器 10 的网络时隙，然后将这个 RTGEN 报文发送给远程交换模块 501 的路由系统进程 3602。路由系统进程 3602 在 RTGEN 报文的 REQTERM 字段中规定的一点上进入路由程序。RTGEN 报文中的信息存入 RDBLK 3201 与 CFBLOCK 3202 的适当字段。由于判定所要求的干线组的忙／闲状态的动态数据是在远程交换模块 501 的 TKOWNER 关

系 3214 与 TKQUE 关系 3215 里，路由系统进程 3602 就能够完成决定终点口的任务。假定作为路由系统进程 3602 进行寻线的结果，干线 543 被指定给这次呼叫。路由系统进程 3602 于是就发送一个包括完备的 TERMGPI 字段的 RTGEN 报文给终点系统进程 3606。作为响应，进程 3606 建立一个终端进程 3622，并将 RTGEN 报文中的信息通过一个 TKTERM 报文传到进程 3622。终端进程 3622 确定用于跟干线 543 通信的终点外围时隙。终端进程 3622 在传输设施 421 到 424 中之一（例如 422）上选一个呼叫时隙，并与主交换模块 301 进行控制通信，使时隙交换单元 311 将所选的传输设施 422 上的呼叫时隙接到时分隔多路交换器 10 上的选定的网络时隙上去。终端进程 3622 进行传送一个 E 位连续性信号经由主交换模块 301 到交换模块 201，并且也发出一个 SETUPCOMP 报文到交换模块 201 的起点端进程 3621。为响应这个 SETUPCOMP 报文，起点端进程 3621 开始传送一个 E 位连续性信号经过主交换模块 301 返回到远程交换模块 501。为了响应这些 E 位连续性信号，起点端进程 3621 与终端进程 3622 在各自的控制 RAM 中写入信息，使起点外围时隙映射到网络时隙而终点外围时隙映射到传输设施 422 的选定呼叫时隙。用户话机 23 与干线 543 之间的通信途径现在就已接通。

一个与刚才对图 31 的描述非常类似的情景，也可以用在从交换模块 201 到远程交换模块所控制的多线寻线组的呼叫上。由于贮存在中央控制器 30 的 DNTRAN 3107 关系有该系统全部线上的查询号翻译信息，对所有从交换模块 201 到个别线的呼叫，决定终点口的任务都可由中央控制器 30 的路由系统进程 3603 来完成。

贮存在中央控制器 30，也贮存在远程交换模块 501 到 504 中的每一个图 25 到 29 的路由程序，是作为系统 I 的集中路由的图 9

到 13 的路由程序的修改本。因此流程图中完成同样或类似功能的方块，在两套图上都用同一号码来表明身份。类似地，图 24 的状态比图 8 的状态图多一个状态和几路状态转换，而两图相对应的状态也都用同一号码来表明身份。

文中对图 25 到 29 的路由程序是以对图 9 到 13 的路由程序用必要的术语进行更改来叙述的。图 25 到 29 的路由程序是在路由系统进程收到一个 RTREQ 报文或者一个 RTGGN 报文，从 START 状态 3001 (图 24) 起动的。在方块 1010 (图 25) 期间，收到的报文被存入 RDBLK。RDBLK 的 RTGSEQ 字段 (图 32) 用于分辨目前路由程序的执行是响应一个 RTREQ 报文还是一个 RTGEN 报文。执行进到决策块 1020，那时根据收到的报文的类型，出现一个分支。如果收到的是一个 RTREQ 报文，执行就跟以前对图 9 到 13 的路由程序所描述的同样方式进行，除非访问的关系之一没有所要求的数据。记住 FIXEDRI, SCRNING, ROUTING, MHG 和 TRKG 关系在中央控制器 30 与远程交换模块 501 到 504 之间是冗余的。因此，除非有误差，对这些关系的尝试访问都应该是成功的。但是中心控制器 30 贮存了所有接在系统上的线的查询号翻译信息，而远程控制模块 501 到 504 中的每一个都只贮存接入那个远程控制模块的线的查询号翻译信息。因此访问一个远程控制模块的 DNTRAN 的企图，只对接入那个模块的线才会成功。在流程图中，用增加一个决策块 1080 的方式反映了这种情况。在决策块 1080 期间要判定在方块 1070 期间对 DNTRAN 关系的访问尝试是否成功。如果未查到所要的数据，执行就从决策块 1080 进到方块 1400 而进入了 SWITH 状态 3015 (图 24)。在方块 1400 期间，在 RDBLK (图 32) 里存入一个 SWREQ 变量，它指出下一个路由系统进程应从图 24 的状态图的哪一

个程序状态进入。从决策方块 1080 转到方块 1400 时，存入的 SWREQ 变量规定了 DNTRAN 状态 3004 (图 24) 作为下一个路由系统进程进入的程序状态。于是执行移到方块 1190 而程序从 SWITCH 状态 3015 移入 INTEGRITY 状态 3012 (图 24)。在方块 1190 期间，要决定下一个处理器。如果目前的处理器是在一个远程交换模块内，下一个处理器一定是中央控制器 30。如果目前的处理器是中央控制器 30，就要用 TERMBLK 中的 MODULE 字段去判定下一个处理器的位置。要查一下状态表来证实下一个处理器是在正常运行，然后执行转到方块 1200。在方块 1200 期间要进入 NWCONN 状态 3013。在一个远程交换模块中，如果不跟中央控制器 30 通信就能决定终点口，选择接收时隙交换器与发送时隙交换器共同能用的时隙去用以将起点外围时隙接到终点外围时隙的功能，是在方块 1200 期间进行的。在中央控制器 30 内，要选定呼叫的共同可用的网络时隙，而且如果是模块之间的呼叫，在方块 1200 期间还要在控制存储器 29 内写入信息以接通网络途径。执行转到方块 1210，要决定传入下一个处理器的报文的型类。RDBLK 中的 RTGSEQ 字段被用以判定路由程序，目前的执行是由于收到了一个 RTREQ 报文或者是一个 RTGEN 报文。如果执行是收到一个 RTGEN 报文的结果，则要建立一个 RTGEN 报文。如果执行是由于收到一个 RTREQ 报文，不过要进行一个转接，这也要建立一个 RTGEN 报文。如果不要转接，则用 RTGDATA 字段中的 TERMTYP 变量 (在 RDBLK 中作为 TEXT 字段的一部分存入) 来判定是否要建立一个 LNTREQ，一个 TKTREQ，还是一个 ANTREQ 报文。执行转到方块 1220，报文缓冲器里的报文被传送出去，而执行结束于 DONE 状态 3014 (图 24)。

回忆一下，每个远程交换模块用来存储多口寻线组，即多线寻线

组或干线组的动态忙／闲数据的关系，只存有接在远程交换模块上的这些组的话线或干线的数据。还要回忆一下，系统中所有其他多口寻线组的动态数据都存在中央控制器 30 里。这就意味着任何给定寻线组的动态数据，都只贮存在一个地方。因此，虽说对 DNTRAN 的失败访问只能在一个远程交换模块上发生，对 LNSTAT, TKOWNER, TKQUE 或 TKSTAT 关系的失败，可以在远程交换模块也可以在中央控制器 30 发生。访问 LNSTAT 关系失败的可能，是这样反映在程序流程图上的：当试图访问 LNSTAT 关系时，方块 1120 后面加上一个决策块 1130。如果从 LNSTAT 关系找不到所要的数据，执行从方块 1130 转到方块 1390。在方块 1390 时期，方块 1110 读 NHG 关系（图 32）得到的 MODULE 字段被存入 TERMBLK，而执行转到 1400。在方块 1400 期间，进入 SWITCH 状态 3015（图 24），规定 MLGPREHLINT 状态 3005（图 24）为进入下一路由系统进程的状态的 SWREQ 变量，被存入 RDBLK。执行于是经过方块 1190, 1200, 1210, 1220 与 1230，而跟以前一样建立一个 RTGEN 报文并发送出去。

类似地，在试图访问 TKOWNER 与 TKQUE 关系的方块 1310 后面加上了决策块 1320，在试图访问 TKSTAT 的方块 1330 后面加上了决策块 1340。执行从决策块 1320 或 1340 的任一个前进，象以前一样，经过方块 1390, 1400, 1190, 1200, 1210, 1220 与 1230。在这两种情况下，在方块 1400 期间，贮存在 RDBLK 内的 SWREQ 变量都规定 TRKPREHUNT 状态 3009（图 24）作为进入下一个路由系统进程的程序状态。

以上所叙述的是响应收到 RTREQ 报文时，图 25 到 29 的路由程序的执行情况。在收到 RTGEN 报文时，执行进程从决策块 1020

转到方块 1410。在方块 1410 期间，要分析 RTGEN 报文中的 REQTERM 字段以决定执行开始的程序状态。REQTERM 还存有应访问的第一个关系的键值。RTGEN 报文的 RTCONTDA 字段中还包括其他需要的变量值，使一个路由系统进程已经完成的工作不再在下一个路由系统进程中重复。这类信息随后就填入 CFBLK。在目前的实施方案中，REQTERM 指定 DNTRAN 状态 3004，MLGREHUNT 状态 3005 或 TRKPREHUNT 状态 3009 作为进入的程序状态。从流程图可以看出，按照 REQTERM 字段的规定，执行从方块 1410 转到方块 1070，方块 1110 或方块 1290。

应当了解，虽然在系统Ⅱ中路由功能只分布到远程交换模块，分布路由的概念可以扩充使路由功能类似地分布到所有系统交换模块里去。

系 统 Ⅲ

图 33 到 35，按图 36 排列着，表示一个本文中称为系统Ⅲ的一个时分隔交换系统；作为系统Ⅱ的一种改型，系统Ⅲ的远程交换模块 501，502，503 与 504 被交接成为一个称为群集的组。在系统Ⅲ中，每一对远程交换模块都由一个象前述的 T1 载波系统那样的数字双向传输设施互相交连着。模块 501（图 35）由传输设施 425，426 与 427 分别接到模块 502，503 与 504，模块 502 由传输设施 435 与 436 分别接到模块 503 与 504，而模块 503 与 504 则由传输设施 445 互连。在系统Ⅲ中，每个设施接口单元，例如 505，都跟七条传输设施交接。

跟系统Ⅱ一样，两部远程交换模块之间的控制通信模式，也是经过时分隔多路交换器 10 的控制通路，以及控制分配单元 31。但是，

由于远程交换模块的群集是作为整体运行的，即使在孤立模式时，控制通信也能在直接交连这些远程交换模块的传输设施上进行。这种控制通信在前面提到的美国专利申请书 493,683 号上有详细说明。在该专利申请书中描述的那样，控制通信是用直接交连的传输设施的 24 条通路之一，或用这种设施上的导出数据链完成的。

也跟系统 II 一样，系统 III 的路由功能是分布在远程交换模块 501 到 504 里的。每个远程交换模块都有一个路由系统进程，与它有关的数据结构 RDBLK, CFBLK, GRPBLK 和 TERMBLK，以及包括 FLXEDRI, SCRNLNG, DNTRAN, ROUTLNG, PORTGROUP, GROUPOPORT, MHG, LNSTAT, TRKG, TKOWNER, TKQUE 和 TKSTAT 关系的数据库。所有路由系统进程都执行跟系统 II 一样的路由程序，程序的流程图见图 25 到 29。跟这些路由系统进程有关的状态图见图 24。跟系统 II 一样，FLXEDRI, SCRNLNG, ROUTLNG, MHG 和 TRKG 关系在中央控制器 30 与远程交换模块 501 到 504 之间是冗余的。对每个这种关系，该系统的有关数据都贮存在中央控制器 30 和远程交换模块 501 到 504 的每一个里面。再者，中央控制器 30 的 DNTRAN 关系贮存所有接到系统线的查询号翻译信息。但是与系统 II 对比，系统 II 的每个远程交换模块中的 DNTRAN 关系只存入接在那个远程交换模块上的线路的查询号翻译信息，而系统 III 的每个远程交换模块的 DNTRAN 关系都存有接到整个远程交换模块 501 到 504 这个群集上的线路的查询号翻译信息。这样，对起源于一个给定的远程交换模块到这个群集中任何远程交换模块上的个别线路的呼叫，决定终点口的功能都可由给定远程交换模块的路由系统进程完成。远程交换模块中贮存的 PORTGROUP 与 GROUPOPORT 关系也存入了与远程交换模块群集中所有口的信息。跟系统 II 一样，

每个远程交换模块中用以贮存多口寻线组的动态忙／闲数据的那些关系，即多线寻线组的 LNSTAT 关系与干线组的 TKOWNER，TKQUE 和 TKSTAT 关系，只存有那些组内全部话线或干线都接到一个远程交换模块上的那些组的这类数据。因此对一个起源于第一个远程交换模块而到达连接第二个远程交换模块的一个多口组的呼叫，第一个远程交换块的路由系统进程能执行它的路由程序直到访问动态数据那一点。然后发出一个 RTGEN 报文（图 32）到第二个远程交换模块中路由系统进程，由它来完成决定终点口的任务。

两部远程交换模块之间的一条给定传输设施上的 23 个通路或时隙，用于这两个模块之间的呼叫。第二十四路则用以替其他二十三路传送信号比特。（在当前这个系统Ⅲ的实施方案中，传输设施上的导出数据链被用于在独立模式中操作的控制通信，而不是用二十三条通路之一）。这两部远程交换模块中的每一部都是它们之间的传输设施上 23 个时隙中 11 个或 12 个的控制器。例如，远程交换模块 501 是传输设施 435 上时隙 1 到 12 的控制者，而远程交换模块 502 则是时隙 13 到 23 的控制器。每个远程交换模块都维持一个时隙状态映射表，辨明接到这个模块上的每条传输设施中的每个时隙的忙／闲状态。如果远程交换模块 501 需要将传输设施 435 上的一个时隙指定给一次呼叫，它首先读它的时隙状态映射表来判定是否时隙 1 到 12 中有一个可用。如果时隙 1 到 12 中有一个或多个可用，它就将一个时隙指定给这次呼叫。但如果没有任何可用的，远程交换模块 501 就将需要的指定通知远程交换模块 502，而模块 502 就读它的时隙状态映射表以判定是否时隙 13 到 23 中有一个可用。如果时隙 13 到 23 中有一个或多个可用，就选用可用的时隙之一给这次呼叫用。如果一个可用的也没有，远程交换模块 501 和 502 就跟中央控制器 30 通信，来接通

这次呼叫。呼叫是经由传输设施 421 到 424 之一，传输设施 431 到 434 之一，以及主交换模块 301 接通的。此外，如果两个远程交换模块是接在不同的主交换模块上的，例如远程交换模块 501 是接在主交换模块 301 上，而远程交换模块 503 则是接在主交换模块 302 上的，这次呼叫就用时分隔多路交换器 10 中的一个网络时隙来接通。

顺序呼叫是一种不跟原来的终点口接通，而是要跟另一个终点口接通的呼叫。在系统Ⅲ中，顺序呼叫是以一种有效方式提供的，这种方式就是在接通最后的终点口之前，将这种呼叫简化为简单呼叫，即只涉及两个口的呼叫。这避免了使服务于最后终点口的交换模块必须根据呼叫是顺序呼叫还是简单呼叫作不同处理的复杂性。

顺序呼叫的两个例子是转发呼叫，即无论第一个号码是忙是闲都必须按照用户提供的号码接通的呼叫，和依次接通呼叫，此时只有在第一个号码忙时才接通另一个号码。为了使模块之间的控制报文减到最少，系统Ⅲ用的接通呼叫的步骤，使在交接远程交换模块的传输设施上时隙的指定延迟到决定了终点口以后。接到终点口上的远程交换模块将呼叫接通到终点口时，并不知道这次呼叫是一个简单呼叫还是一个顺序呼叫。允许顺序呼叫简化为简单呼叫的性能，本文中简称为封闭。

作为系统Ⅲ中顺序呼叫的第一个例子，设想接到远程交换模块 503 的用户话机 548，拨接在远程交换模块 502 上的用户话机 538 的查询号码。在远程交换模块 503（图 37）里，呼叫处理控制系统进程 4001 为响应发现离钩状态而建立的起点端进程 4003，收下了拨出的查询号码。（为简单起见，在图 37 中省略了远程交换模块 504 和接到模块 504 上的传输设施）。起点端进程 4003 分析拨号以取得前缀指标（PI），终点指标（DI），数字计数（DLGCNT）与

处理(TREATMENT)变量之值。起点端进程4003也根据起点线的特征定下筛选指标(SI)。然后起点端进程4003在报文缓冲器内编一个路由请求报文RTREQ。RTREQ报文(图14)已在系统I的说明中描述过。

一旦形成RTREQ报文，就被传送到远程交换模块503内的路由系统进程4002。路由系统进程4002在其结合的RDBLK中存下这个RTREQ报文。(每个路由系统进程都有与它结合的数据结构RDBLK, CFBLK, GRPBLK和TERMBLK，以及包括FLXEDRI, SCRNLNG, DNTRAN, ROUTLNG, PORTGROUP, GROUPPORT, MHG, LNSTAT, TRKG, TKOWNER, TKQUE和TKSTAT关系的数据库)。路由系统进程4002用RTREQ报文中的信息去访问与它结合的数据库。由于存在远程交换模块503中贮存的DNTRAN关系包括接到远程交换模块501到504的所有线上的查询号翻译信息，路由系统进程4002就能完成决定接到用户话机538的口的全局口身分(GPI)的任务。GPI的MODULE字段确定了用户话机538是接在远程交换模块502上的。不过远程交换模块503并不在当时连接着模块503与模块502的传输设施435上指定呼叫时隙。路由系统进程4002发送一个在图32描述过的通用路由要求(RTGEN)报文给远程交换模块502中的终点系统进程4005。进程4005作出响应，以GPI为键访问图32中表示的PORTSTATUS关系。PORTSTATUS关系包括GPI字段，判明GPI所定的口的忙/闲状态的BUSY/IDLE字段，判明是否要转送呼叫，如果要，则必须有呼叫要向之转送的查询号的CF字段，以及类似地要判明呼叫是否要依次接通，连同适当的查询号的SC字段。一个给定远程交换模块中的PORTSTATUS关系包括那个远程交换模块的所有口的有关数据。在此例中，假定接

到用户话机 538 的线路的 PORTSTATUS 关系元组判明要进行转发呼叫，而且呼叫要转发到随后即将判明的、与接到远程交换模块 501 的用户话机 528 的查询号上去。作为这个 PORTSTATUS 关系读出的响应，终点系统进程 4005 建立一个转发端进程 4007 并向这个进程发送一个呼叫转发 (CF) 报文。CF 报文包括从远程交换模块 503 的路由系统进程 4002 收到的 RTGEN 报文中出现的所有关于产生这次呼叫的信息。为了响应这个 CF 报文，呼叫转发端进程 4007 向用户话机 538 发送一个短期振铃电压，以产生所谓的乒乓通知在用户话机 538 的任何人，一个进来的呼叫已被转发出去了。然后呼叫转发端进程 4007 给远程交换模块 502 的路由系统进程 4006 发送一个重定路由 (RERTE) 报文。在这一点上，这次呼叫已被简化为一个简单呼叫。路由系统进程 4006 对这个 RERTE 报文作出响应的方式和对 RTREQ 报文一样，而且由于终点口是接在远程交换模块 501 到 504 的群集之一上的，路由系统进程 4006 就完成了接到用户话机 528 终点口的 GPI 的决定。GPI 的 MODULE 字段指定了远程交换模块 501。因此路由系统进程 4006 就对远程交换模块 501 的终点系统进程 4008 发出一个 RTGEN 报文。重要的是应当注意到，由于封闭特性，进程 4008 收到的 RTGEN 报文包括与路由系统进程 4002 发出的 RTGEN 报文中出现的关于呼叫起源的同样信息。因此，终点系统不管呼叫是一个简单呼叫还是一个顺序呼叫，终点系统 4008 的响应都是一样的。终点系统进程 4008 读出 PORTSTATUS 关系 (图 32)，然后，按照目前这个例子，判定用户话机 528 现在是闲的。进程 4008 于是建立一个终端进程 4010，并用一个本文以前描述过的 LTERM 报文 (图 14) 将呼叫信息转送进程 4010 里去。在这一点上，知道了呼叫的终点口，又知道了起点口是接在远程交换模块 503 上的，进程

4010 就读时隙状态映射表以判别交接远程交换模块 501 和 503 的传输设施 426 的时隙 1 到 12 的忙／闲状态。假定 12 个时隙中有一个可用，进程 4010 就为这次呼叫指定时隙。（12 个时隙中没有一个可用的另一种情况，前面已讨论过）。呼叫接通步骤的余下部分就以典型的方式来完成，包括向远程交换模块 503 的起点端进程 4003 发送一个 SETUPCOMP 报文，以及将起点外围时隙与终点外围时隙映射到传输设施 426 上的呼叫时隙。如果呼叫接通步骤要在知道需要转发之后再回到远程交换模块 503 去解除一个以前指定了的呼叫时隙，那就要发四个报文了。

作为第二个例子，假定跟前面一样用户话机 548 拨出与用户话机 538 结合的查询号，但 PORTSTATUS 关系的适合的元组表明呼叫是要依次接通到接入交换模块 201 的用户话机 23 的。这个例子在建立起点端进程 4021（图 38），由远程交换模块 503 中的路由系统进程 4002 来判定原来的终点口，以及发送 RTGEN 报文到远程交换模块 502 的终点系统进程 4005 等方面，与前面的例子是类似的。但在现在的情况，进程 4005 在读了 PORTSTATUS 关系之后，知道用户话机 538 目前是忙的，而到用户话机 538 的呼叫应依次接到与用户话机 23 有连系的查询号上去。因此进程 4005 就建立一个依次接通端进程 4022，并发送一个包括起源信息的依次接通（SC）报文到进程 4022。进程 4022 随后用一个 RERTE 报文将起源信息与依次接通信息传到路由系统进程 4006。在这一点上，这次呼叫已被简化为一个简单呼叫。由于远程交换模块 502 中存入的 DNTRAN 关系没有关于用户话机 23 的查询号翻译信息，执行路由程序（图 25 到 29）的结果就是发送一个 RTGEN 报文，这是因为访问 DNTRAN 关系的尝试失败了。正象关于系统 II 的描述一样，RTGEN 报文包括表明当路

由被下一个处理器继续下去时进入路由程序的状态的 REQTERM 字段，以及读下一个关系时需要的键值。 RTGEN 还包括一些已经被路由系统进程 4006 判明的变量值的 RTCONTDA 字段，使路由继续进行时不致于重复不必要的工作。 RTGEN 报文发到中央控制器 30 中的路由系统进程 4004。由于中央控制器 30 存的 DNTRAN 关系有系统中所有线路的查询号翻译信息，路由系统进程就完成了决定终点口的任务。由于路由系统进程 4004 知道起点口是在接到主交换模块 302 的远程交换模块 503 上，而终点口是在交换模块 201 上，路由系统进程就指定这次呼叫用的网络时隙（经过连接交换模块 201 与主交换模块 302 的时分隔多路交换器 10），并这样接通这个途径。路由系统进程 4004 于是发送一个包括已更新的 PATHDES 字段与已完成的 TERMGPI 字段的 RTGEN 报文到交换模块 201 的终点系统进程 4011。假定用户话机 23 目前是闲的，进程 4011 就建立一个终端进程 4023，并给它发送一个 LINTERM 报文。呼叫接通步骤的其余部分包括有给远程交换模块 503 的起点端进程 4021 发送一个 SETUPCOMP 报文，将终点外围时隙映射到时分隔多路交换器 10 的被指定的网络时隙上，在交接远程交换模块 503 与主交换模块 302 的传输设施 441 到 444 之一上面给这次呼叫指定一个时隙（例如在传输设施 441 上），将起点外围时隙映射到传输设施 441 上的呼叫时隙，以及将传输设施 441 上的呼叫时隙映射到时分隔多路交换器 10 的被指定的网络时隙上。

作为第三个例子，设想接在远程交换模块 503 上的用户话机 548 拨出与接在交换模块 201 上的用户话机 23 相结合的查询号。再假定存在交换模块 201 中的 PORTSTATUS 关系的合适的元组判明用户话机 23 要实行转发，而呼叫要转发到与接在远程交换模块 502 上

的用户话机 538 相结合的查询号上去。按前述的方式建立一个起点端进程 4031 (图 39)，并由它发送一个 RTREQ 报文到路由系统进程 4002。由于远程交换模块 503 所存的 DNTRAN 关系不包括用户话机 23 的查询号翻译信息，当路由程序被路由系统进程 4002 执行时，访问 DNTRAN 关系的尝试失败，而一个 RTGEN 报文就被发送到中央控制器 30 的路由系统进程 4004。路由系统进程 4004 完成决定接到用户话机 23 的终点口的 GPI 的任务。路由系统进程 4004 也在交换模块 201 与主交换模块 302 之间的时分隔多路交换器 10 上为这次呼叫指定一个网络时隙，从而接通了这条途径。路由系统进程 4004 于是发送一个包括更新了的 PATHDES 字段与已完备的 TERMGPI 字段的 RTGEN 报文给交换模块 201 中的终点系统进程 4011。作为响应，进程 4011 读 PORTSTATUS 关系并判明到用户话机 23 的呼叫都要转发到跟用户话机 538 结合的查询号上去。进程 4011 于是建立一个呼叫转发端进程 4032 而向它发送一个 CF 报文。呼叫转发端进程 4032 发送短期振铃电压给用户话机 23，然后发回一个 RERTE 报文给中央控制器 30 内的路由系统进程 4004。作为 RERTE 这份报文的响应，路由系统进程 4004 解除了以前指令的网络时隙，并完成了接在用户话机 538 上的终点口的 GPI 的任务。在这一点上，这次呼叫已经简化为简单呼叫。GPI 的 MODULE 字段判定终点口是在远程交换模块 502 上。知道起点口是在远程交换模块 503 上，路由系统进程 4004 不替这次呼叫指定一个新的网络时隙，而是发送一个包括已完备的 TERMGPI 的 RTGEN 报文到远程交换模块 502 的终点系统进程 4005。进程 4005 读 PORTSTATUS 关系而判明用户话机 538 目前是闲的，然后在交接远程交换模块 502 与 503 的传输设施 435 的时隙 1 到 12 中选一个可用的时隙作为呼叫时隙。进程 4005 建立

一个终端进程 4033 并向它发送一个 LTERM 报文。呼叫接通步骤的余下的部分按照正常方式进行，包括发送一个 SETUPCOMP 报文给远程交换模块 503 中的起点端进程 4031，并将起点外围时隙和终点外围时隙映射到传输设施 435 的呼叫时隙上去。

由于每个远程交换模块都能完成对所有接到模块 501 到 504 这个群集的个别线路呼叫的终点口决定功能，一个起点远程交换模块能在判明终点远程交换模块之后，立即选择而指定交接传输设备上的呼叫通路。在系统Ⅲ的另一个实施方案中就是这样做的。但是一旦终点远程交换模块辨明这次呼叫是一个顺序呼叫，接通步骤就要回到起点远程交换模块，使原先指定的呼叫通路被取消以简化这次呼叫为一个简单呼叫。

作为系统Ⅲ的另一个实施方案的一个顺序呼叫的例子，设想接在远程交换模块 503 上的用户话机 548 拨接在远程交换模块 502 上的用户话机 538 的查询号。在远程交换模块 503 中（图 40），一个呼叫处理控制系统进程 4001 响应发现离钩状态而建立的起点端进程 4043，接收拨来的查询号码。起点端进程 4043 分析拨号以取得 PI, DI, DLGCNT 与 TREAT 的值。起点端进程 4043 也根据起点线的特征定出 SI 之值。起点端进程 4043 于是发送一个 RTREQ 报文给路由系统进程 4002。

路由系统进程 4002 用 RTREQ 报文中的信息去访问与它有关的数据库。由于远程交换模块 503 中存的 DNTRAN 关系包括接到远程交换模块 501 到 504 上所有线路的查询号翻译信息，路由系统进程 4002 能够完成决定接到用户话机 538 的口的 GPI 的任务。GPI 的 MODULE 字段指明用户话机 538 是接在远程交换模块 502 上的。路由系统进程 4002 于是判定交接模块 503 与 502 的传输设施 435 的

通路 13 到 23 中是否有一条可用。在此例中，假定通路 13 可用，并被进程 4002 选中，路由系统进程 4002 发出一个 RTGEN 报文（包括选用的通路 13）给远程交换模块 502 中的终点系统进程 4005。进程 4005 的响应是用 GPI 为键去访问 PORTSTATUS 关系。假定 PORTSTATUS 关系中关于接到用户话机 538 的话线的元组指明要进行转发，而呼应回应转发到一个查询号，这个号随后将判明是跟接到远程交换模块 501 的用户话机 528 相结合的。读这个 PORTSTATUS 关系的反应是终点系统进程 4005 建立一个呼叫转发端进程 4047 并发出一个呼叫转发（CF）报文给进程 4047。为了响应这个 CF 报文，呼叫转发端进程 4047 进行发送一个短期振铃电压给用户话机 538 去产生铃声。呼叫转发端进程 4047 于是向远程交换模块 503 的路由系统进程 4002 发出一个 RERTE 报文。路由系统进程 4002 完成决定接到用户话机 528 的终点口的 GPI 的任务。GPI 的 MODULE 字段指明远程交换模块 501。路由系统进程 4002 于是取消了原先指定的传输设施 435 上的呼叫通路 13。在这一点上这次呼叫已被简化为一个简单呼叫。下一步，路由系统进程 4002 判明交接模块 503 与 501 的传输设施 426 上通路 13 到 23 中是否有一条可用。假定通路 18 可用并被路由系统进程 4002 选出指定给这次呼叫使用。路由系统进程 4002 发出一个 RTGEN 报文（包括选用的呼叫通路 18）给远程交换模块 501 上的终点系统进程 4008。必须注意，进程 4008 收到的 RTGEN 报文包括终点系统进程 4005 收到的 RTGEN 报文中出现的同样的关于呼叫起源的信息。呼叫接通步骤中余下的部分按以前参照图 37 时说明的例子的同样方式进行。

系统N

系统N(图41)用的硬件体系结构基本上和本文中以前描述的系统I(图2)是一样的。但是在系统N中，所有呼叫处理功能，除开选择网络时隙并对模块间的呼叫建立时分隔多路交换器10的途径以外，都分布到这些交换模块而不涉及到系统中央控制。特别是决定终点口的功能，对所有呼叫都只依靠这些交换模块的合作处理就能完成。因此系统N的中央控制器30能作为一个比较廉价的处理机来实现，虽然两个系统都有29部交换模块，但因为比起系统I的中央控制器30来，所必须处理的处理负担要少得多。或者换一个方式说，同样的处理机，用用于包括更多交换模块的系统中。

在系统N中，交换模块201到229的每一个都有一个路由系统进程，与它结合的数据结构RDBLK，CFBLK，GRPBBLK与TERMBLK以及包括FIXEDRI，SCRNING，DNTRAN，ROUTING，PORTGROUP，GROUPPORT，MHG，LNSTAT，TRKG，TKOWNER，TKQUE与TKSTAT关系的数据库。每个交换模块也都有一个MODTRAN关系(图32)。所有路由系统进程执行同样的路由程序，这个程序对系统II和III的程序(它的流程图见图25到29)作了更改。这些更改可见图42，本文以下将说明。这些路由系统进程的状态图跟图24一样，只不过不需要NWCONN状态30B。中央控制器30'没有路由系统进程而只有一个途径寻找系统进程，它只负责选定网络时隙和为模块间的呼叫建立时分隔多路交换器10的途径。

查询号翻译

在交换模块201到229的每一个中，FIXEDRI，SCRNING，ROUTING，PORTGROUP，GROUPPORT，

MHG和TBKG关系都冗余的。每个这些关系都存有整个系统的有关数据。DNTTRAN关系在各个交换模块中不同。例如，交换模块201的DNTTRAN关系存入所有接到交换模块201的线路的查询号翻译信息，此外还有从1000到11799一组1800个DN的信息。（记住DN的第一位是NOC而DN的最后四位是拨号的最后四位。）交换模块202中存入所有接到交换模块202的线路的查询号翻译信息，还有从11800到13599的一组1800个DN的信息。类似地，每一个其他的交换模块都存有所有接到那个模块上的线路的查询号翻译信息，此外还有如表2中规定的一组1800个DN的信息。

表 2

交换模块	nxx	DN
201	355	10000-11799
202	355	11800-13599
203	355	13600-15399
204	355	15400-17199
205	355	17200-18999
206	357	20000-21799
207	357	21800-23599
208	357	23600-25399
209	357	25400-27199
210	357	27200-28990
•	•	•
•	•	•
•	•	•

.	.	.
.	.	.
.	.	.
221	491	50000-51799
222	491	51800-53599
223	491	53600-55399
224	491	55400-57199
225	491	57200-58999
226	493	60000-61799
227	493	61800-63599
228	493	63600-65399
229	493	65400-67199

当然，没有必要每个交换模块结合同样数目的DN。目前这个系统里每个交换模块规定1800DN只不过为了示范。此外，虽然一般讲接到一个给定的交换模块的用户话机不必限于一个特定的1800DN组，但是使这一组1800个DN与指定给接到那个交换模块的用户话机的那一组DN之间的交叉变为最大是有好处的。MODTRAN关系（图32）在每个交换模块中都有冗余，它包括NOCD4D3属性与MODULE属性；MODTRAN关系是用于规定那些贮存着在起点交换模块中找不到需要信息的那种来源的查询号翻译信息的交换模块的位置。目前这个例子的MODTRAN关系的内容、综列于表3。

表 3

M O D T R A N	
NOOD 4 D3	MODULE
1 0 0 - 1 1 7	2 0 1
1 1 8 - 1 3 5	2 0 2
1 3 6 - 1 5 3	2 0 3
1 5 4 - 1 7 1	2 0 4
1 7 2 - 1 8 9	2 0 5
2 0 0 - 2 1 7	2 0 6
2 1 8 - 2 3 5	2 0 7
2 3 6 - 2 5 3	2 0 8
2 5 4 - 2 7 1	2 0 9
2 7 2 - 2 8 9	2 1 0
•	•
•	•
•	•
•	•
5 0 0 - 5 1 7	2 2 1
5 1 8 - 5 3 5	2 2 2
5 3 6 - 5 5 3	2 2 3
5 5 4 - 5 7 1	2 2 4
5 7 2 - 5 8 9	2 2 5
6 0 0 - 6 1 7	2 2 6
6 1 8 - 6 3 5	2 2 7
6 3 6 - 6 5 3	2 2 8
6 5 4 - 6 7 1	2 2 9

NO CD 4 D 3 属性代表一个N O C 与拨号中的千位数D 4 及百位数D 3 的组合。例如，由于交换模块2 0 1 存有DN 1 0 0 0 到1 1 7 9 9 的查询号翻译信息，M O D T R A N 关系就指明对于N O C D 4 D 3 值从1 0 0 到1 1 7 的来源，必要的查询号翻译信息在交换模块2 0 1 里。（注意，如果结合到交换模块上的DN 是以千数计的，N O C 只要加上千位数D 4 就足以指明信息的地点了。）多口寻线组的动态数据，即，L N S T A T ，T K O W N E R ，T K Q U E 与T K S T A T 关系，它们的分布在后面将有说明。

系统N 对流程图图2 5 到2 9 要求的更改见图4 2 。在决策块1 0 8 0 与方块1 4 0 0 之间插入了一个方块1 0 8 5 。记住在方块1 0 7 0 中，用DN 为键读了D N T R A N 关系，然后在决策块1 0 8 0 中要判明在D N T R A N 关系中有没有所要的数据。如图4 2 表明的那样，如果在方块1 0 8 0 中判定没有所要的数据，执行就转到方块1 0 8 5 。在方块1 0 8 5 期间，用N O C D 4 D 3 属性为键读出M O D T R A N 关系（图3 2 ）以取得判明需要的数据存在哪个交换模块里的M O D U L E 字段。将M O D U L E 字段存入T E R M B L K ，执行转到方块1 4 0 0 。在方块1 4 0 0 期间，进入S W I T C H 状态3 0 1 4 （图2 4 ），指明D N T R A N 状态3 0 0 4 作为下一个路由系统进程进入程序的状态的变量S W R E Q 、存入B D B L K 。执行转到方块1 1 9 0 ，在此期间要决定下一个处理器。如果决定终点口的功能已经完成，下一个处理器必定是中央控制器3 0 ' 。但如果终点口还没有定出，下一个处理器就在T E R M B L K 的M O D U L E 字段规定的交换模块里。于是要查明下一个处理器的运行状态，而执行转到方块1 2 1 0 ，在此期间要决定发出报文的类别。如果下一个处理器处于交换模块之一，就编定一个前文已描述过的R T G E N 报文（图3 2 ）。如果下一个处理器是中央控制器3 0 ' ，就编定一个途径要求（P R ）

报文。用 PR 报文去要求中央控制器 30' 为这次呼叫选定一个网络时隙，而且如果呼叫是一次模块之间的呼叫，就要建立时分隔多路交换器 10 的途径。PR 报文包括与 RTGEN 同样的一些字段，只是 PR 报文不需要 REQTERM 与 RTCONTDA 字段。执行转到方块 1220，用 RDBLK，CFBLK 与 TERMBLK 中的数据编一个合乎要求的报文，然后发出报文。

作为系统 N 中呼叫接通的第一个例子，设想接到交换模块 229 的用户话机 25 刚刚离钩。话线单元 21 内的扫描发现了这个脱钩状态。交换模块 229 内的呼叫处理控制系统进程 5001 (图 43) 被通知发现脱钩，而作为响应，建立一个起点端进程 5008。起点端进程 5008 负责控制发送拨号音给用户话机 25，以及随后接收用户话机 25 的拨号。此例中假设拨号是 355-2289。起点端进程 5008 分析拨号以取得前缀指标 (PI)，终点指标 (DI)，拨号位数 (DIGCNT) 与处理 (TREAT) 等变量之值。起点端进程 5008 根据起点线的特征、决定筛选指标 (SI) 之值。起点端进程 5008 于是在报文缓冲器内编一个路由要求报文 RTREQ (图 14)。

一旦 RTREQ 报文编成，它就被起点端进程 5008 发送到仍在交换模块 229 内的路由系统进程 5002 (图 42)。路由系统进程 5002 在与它结合的 RDBLK 里存下这份 RTREQ 报文。路由系统进程 5002 然后按路由程序 (按图 42 的更改的图 25 到 29) 依次访问与它相关的数据库。由于 nxx 数字位 355 在此例中被编为 NOC 码 1，拨出来的号码 355-2289 就由 DN12289 来代表。交换模块 229 的 DNTRAN 关系中没有 DN12289 的查询号翻译信息。因而当执行路由程序时用 DN12289 为键去访问 DNTRAN 关系而失败了。用 NOCD4D3 码 122 为键去读 MODTRAN，从而知道确实贮存了所要的查询号翻译信息的交换模

块的身分。按本例的情况，查出来信息是存在交换模块202中(见表3)。路由系统进程5002编一个包括REQTERM字段的通用路由(RTGEN)报文(图32)，该报文指明DNTBAN状态3004(图24)作为下一个路由系统进程应进入的程序状态，并指定DN12289作为访问DNTTRAN关系的键。然后路由系统进程5002将这份RTGEN报文发送给交换模块202的路由系统进程5004。

为了响应这份RTGEN报文的REQTERM字段，路由系统进程5004开始在DNTTRAN状态3004(图24)执行它的路由程序。用DN12289为键访问了DNTTRAN关系。这一次，正如交换模块229中读到的MODTRAN关系指出的，必要的信息有了。在此例中假定DN12289译为交换模块201中接用户话机23口的全局口身分。由于路由系统进程5004已经完成这次呼叫的终点口决定，进程5004就编一个包括在ORIGGPI字段中指明交换模块229接用户话机25的口、并在TERMGPI字段中指明交换模块201口接用户话机23的口的途径要求(PR)报文(图32)。进程5004将PR报文发给中央控制器30'的一个路由寻找系统进程5007。

中央控制器30'存有一个对交换模块201到229与时分隔多路交换器10之间的所有时分隔多路线上所有时隙、辨明忙／闲状态的网络映射表。记住在每个交换模块与时分隔多路交换器10之间，都有两路256时隙-时分隔多路线对。因而在一个给定交换模块与时分隔多路交换之间有512个可用的时隙。中央控制器30'内的途径寻找系统进程5007用选定交换模块229与交换模块201之间一个共同可用的时隙的方式来响应这份PR报文。途径寻找系统进程5007也将规定选用的网络时隙的信息写入控制存储器29。进程5007于是发出一个在PATHDES字段中包括这个规定时隙的INTREQ报

文(图14)给交换模块201的终点系统进程5005。

为了响应这份LNTREQ报文，终点系统进程5005读存在交换模块201内的PORTSTATUS关系(图32)以判明用户话机23当时是忙还是闲。如果用户话机23是闲的，进程5005就建立一个终端进程5009，并经由一份LTERM报文将从LNTREQ报文中收到的信息转给进程5009。终点进程5009进行发送振铃电压给用户话机23，并发出一个E位连续信号和可闻的振铃音给交换模块229。然后终端进程5009给交换模块229的起点端进程5008发出一份包括完整途径描述语PATHDES的接线完毕(SETUPCOMP)报文。作为响应，起点端进程5008进行传送E位连续性信号给交换模块201。交换模块201从交换模块229收到E位连续性信号后，终端进程5009就决定将使用跟用户话机23通话的终点外围时隙，并在交换模块201的控制RAM65中写入规定终点外围时隙与网络时隙之间的映射的信息。类似地，交换模块229从交换模块201收到E位连续性信号后，起点端进程5008就决定将使用跟用户话机通话的起点外围时隙，并在交换模块229的控制RAM55中写入规定起点外围时隙与网络时隙之间的映射的信息。用户话机25与23之间的通信途径现已建立完毕。

作为第二个例子，设想接在交换模块229上的用户话机25拨出493-5433的号。呼叫处理控制系统进程5001在被通知发现离钩状态后建立的起点端进程5018、收收拨号，然后发一份RTREQ报文给路由系统进程5002。作为响应，路由系统进程5002执行它的路由程序。在当前的情况，最好是这样，DN65433是配给接在同一个交换模块229的一个终点用户话机，例如用户话机26；在模块229里存有DN65433的查询号翻译信息(见表2)。因此在执行路由程序时，访问DNTTRAN是成功

的，决定终点口的任务就被路由系统进程5002完成。然后路由系统进程5002给中央控制器30'的途径寻找系统进程5007发出一份PR报文（图32）。由于PR报文中的ORG GPI字段与TERM GPI字段两者都指定交换模块229，途径寻找系统进程5007选定两个可用的网络时隙去连接交换模块229内部的接收时隙交换器50到发送时隙交换器53。每个传输方向用一个网络时隙。由于呼叫是模块内部的呼叫，所以不需要时分隔多路交换器10的途径。因而进程5007不在控制存储器29里写入信息。进程5007发出一份在PATHDES字段中包括选定的网络时隙的LNTEREQ报文给终点系统进程5010。呼叫接通步骤的其余部分包括建立一个终端进程5019，从进程5010发出一份LNTERM报文给进程5019，从进程5019发出一份SETUPCOMP报文给起点端进程5018，以及写入交换模块229的控制RAM 55以指明起点与终点外围时隙对网络时隙的映射，是按以前描述过的方式进行的。

作为第三个例子，设想接到交换模块229上的用户话机25拨出493-5552的号码。由于DN 65552的查询号翻译信息存在交换模块229的DNTTRAN关系里（见表2），路由系统进程5002（图45）能完成决定终点口的任务。假定在此例内，DN 65552被指定给连在交换模块201上的用户话机23。路由系统进程5002发出一份包括完备的TERM GPI字段的PR报文给中央控制器30'的途径寻找系统进程5007。作为响应，进程5007选一个在交换模块229与201大家可用的网络时隙，并用写入控制存储器29的方式建立这条途径。连接用户话机25与23的呼叫接通步骤的余下部分，按以前描述的方式完成。作为第四个例子，设想交换模块229上的用户话机25拨出355-1566的号码。由于DN 11566的查询号翻译信息没有存在交换模块229的DNTTRAN

是已接通的呼叫所有的通路以及为未解决的呼叫保留为候选组的一部分的任何通路。这种通路的保留将在后文讨论。) 执行进程从方块6 04 0 转到方块6 05 0，而编好的P H R 报文被发送到起点交换模块的途径寻找系统进程。

一个给定的交换模块中的途径寻找系统进程为响应一份P H R 报文而执行的程序的流程图见于图5 1。在方块6 11 0期间收到了P H R 报文，执行转到方块6 12 0。访问T I M E S L O T 关系，并为这次呼叫选定一组候选时隙。候选组中每个时隙必须符合下列准则：1) 在给定的交换模块的T I M E S L O T 关系中该个时隙必须标明为可用，2) 在这次模块间呼叫的对方交换模块的P H R 报文中，这个时隙必须标明为可用。在目前描述的实施例方案中，候选组的大小是预先规定的，例如包括四个时隙。在方块6 13 0中候选时隙组被保留下来，使得这些时隙不会被给定交换模块选作任何其他候选组的一部分，直到这次特定的呼叫被解决或至少离开了候选组为止。必须注意到，在方块6 13 0中保留的时隙能被选为其他交换模块与给定的交换模块之间的呼叫的网络时隙。方块6 13 0中的保留只意味着这些被保留的时隙不会被这个给定的交换模块选为其他候选组的一部分，直到它们脱离这个候选组为止。执行从方块6 13 0转到方块6 14 0，在此期间，候选组的四个时隙之一被选为首选(F C)时隙。然后在方块6 15 0期间，编一份候选组(C S)报文，标明候选组中的四个时隙以及四个时隙中哪一个是F C 时间片。编好的C S 报文在方块6 16 0期间发送给对方交换模块中的途径寻找系统进程。

一个给定交换模块的途径寻找系统进程为响应一份C S 报文而执行的程序的流程图见图5 2到5 4。在方块6 21 0期间收到C S 报文，执行转到方块6 21 5，在此期间，候选时隙组被保留起来这些时隙不会被给定交换模块选为其他呼叫的候选组的一部分。执行从方块62 15

状态3 0 1 2 (图24)而发现称为主交换模块中的处理器不能工作时，就转而将R T G E N 报文另外发送给处理器正常工作的两个二次交换模块之一。

刚才描述的系统N的实施方案中，每一个交换模块的D N T R A N 关系中都存有本模块接上的所有线路的查询号翻译信息。此外还存有表2规定的一组1800个DN信息。在另一个实施方案中，每个交换模块的D N T R A N 关系中只存下一组1800个DN的查询号翻译信息，但没有该交换模块的任何查询号不在预定的1800 DN组内的线路上的信息。不是在读D N T R A N 关系失败后才读M O D T R A N ，相反地，先读M O D T R A N 关系而且并不去读D N T R A N 关系，除非 M O D T R A N 指出查询号翻译信息在其中。否则路由就在 M O D T R A N 指定的交换模块中继续。

控制多口寻线组

回忆在系统I中，多口寻线组，即多线寻线组与干线组的控制，无例外地都处在中央控制器30之内。再回忆一下在系统II与III中，某些多口寻线组的控制功能是分布在远程交换模块501到504中。但是这种分布的控制功能只是为那些所有组员都接在同一个远程交换模块的多口寻线组而设的。组员分跨两个或更多个模块的多口寻线组的控制，仍留在中央控制器30之内。在目前描述的系统N中，所有多口寻线组的控制都分布在交换模块201到229之内。每个多口寻线组都被指定一个交换模块201到229作为它的组控制单元。在此实施方案中，作这种指定方案时要遵循某些规定。对所有组员都接一个交换模块上的多口寻线组，组控制单元就定在那个交换模块上。对组员分跨两个或更多个模块的多口寻线组，有最多个组员接入的模块就被定为组控制单元。如果所有接入一组组员的几个模块都接入数目相同的组员，则任选其中

关系里(见表2)，路由系统进程5002(图46)不能完成决定终点口的任务。当执行路由程序而访问DNTRAN关系失败时，就以NOOD4D3码115为键、读MODTRAN关系，以判明所需的查询号翻译信息所处的位置。按此例来说，需要的信息判明是存在交换模块201中(见表3)。因此就编一份包括指明DNTRAN状态3004(图24)的REQTERM字段以及作为DNTRAN关系的键DN 11566的RTGEN报文。路由系统进程5002将RTGEN报文发给交换模块201的路由系统进程5006。

路由系统进程5006在DNTRAN状态3004进入它的路由程序。正如按照交换模块229中的MODTRAN关系的读出所期望那样，用DN 11566去访问DNTRAN是成功的。假定在此例中DN 11566是配给接在交换模块201上的用户话机23的。再一次，本例代表一个较好的安排，因为DN 11566的查询号翻译信息是贮存在配给DN 11566的用户话机接入的同一交换模块201里。由于路由系统进程5006能完成决定终点口的任务，进程5006就发一个PR报文给中央控制器301中的途径寻找系统进程5007。作为响应，进程5007选一个交换模块201与209之间共同可用的网络时隙，从而建立时分隔多路交换器10的途径。进程5007于是发出一份LNTREQ报文给交换模块201中的终点系统进程5005，而呼叫接通步骤的余下部分就按以前描述过的方式进行，从而将用户话机23与25接通。

虽然在图32上没有写出，MODTRAN关系却包括两个查询号翻译信息的二次位置作为属性，这两个二次位置是在MODULE属性指明的交换模块不能工作时使用的。一个给定的1800DN组的查询号翻译信息是存储在三个交换模块的DNTRAN关系内的，其中一个称为主，两个称为二次。在方块1190(图42)期间进入INTEGRITY

一个模块作为组控制单元。此外，信标DN，即同户呼叫一个多线寻线组用的DN，定在接到组控制单元里的一个组员上。虽然采用这些对指定的规定、结果在系统N中的模块间控制报文的数目方面取得一些效率显然也可用别种规定，包括纯粹任意的指定。重要的依据是每个组要指定一个控制单元。

指明多口寻线组的忙／闲状态的动态数据，即多线寻线组的INSTA关系与干线组的TKOWNER，TKQ忙和TKSTAT关系，都存入组控制单元。考虑一下表4中的典型组。

表 4

多口寻线组表	
组	组控制单元
多线寻线组A (所有组员均在SM201)	SM 201
多线寻线组B (2个组员在SM201 12个组员在SM202 2个组员在SM229)	SM 202
多线寻线组C (每个SM201-229上一个组员)	SM 208
⋮	
干线组A (所有组员均在SM201)	SM 201
干线组B (16个组员在SM201 32个组员在SM202)	SM 202
⋮	

多线寻线组A的所有组员都接入交换模块201。因而交换模块201就被指定为组A的组控制单元，而组A的LNSTAT关系就存入交换模块201。多线寻线组B有两个组员接入交换模块201，12个组员接入交换模块202和两个组员接入交换模块229。交换模块202被指定为组B的组控制单元，而组B的LNSTAT关系都存入交换模块202。多线寻线组C在交换模块201到229的每一个上都连接一个组员。交换模块208被任意指定为组C的组控制单元，而组C的LNSTAT关系都存入交换模块208。干线组A是一个先进先出(FIFO)组，所有它的组员都接入交换模块201。交换模块201被指定为组控制单元，而干线组A的TKOWNE和TKQUE关系都存入交换模块201。干线组B是一个前进后退组，它有16个组员接入交换模块201和32个组员接入交换模块202。交换模块202被指定为组控制单元，而干线组B的TKSTAT关系都存入交换模块202。

所有从非组控制单元的交换模块向多口寻线组的呼叫，需要发送一份控制报文给组控制单元。组控制单元用这组的动态忙/闲数据去进行规定的寻线算法，如果有一个闲的组员就对这次呼叫指定一个闲的组员。组控制单元立刻就标明被指定的组员为忙。组控制单元于是发送一份控制报文给接在指定组员上的交换模块，以完成到那个组员的接线。当这个指定的组员再一次可供使用时，就发送一份控制报文回到组控制单元，使这个指定组员能标明为闲。

作为涉及这种多口寻线组的呼叫的第一个例子，设想接入交换模块229的用户话机25拨355-1922的号，而355-1922是多线寻线组B的信标查询号，有两个组员接入交换模块201，12个组员接入交换模块202，两个组员接入交换模块229，(见表2与4)。交换模块229的路由系统进程5002(图43)用DN

11922去访问DNTRAN的尝试失败了。用NOOD4D3码119作为键去读MODTRAN关系(图32)而判明需要的查询号翻译信息是不是在交换模块202里存着(见表3)。路由系统进程5002编一份RTGEN报文，其中包括指定DNTRAN状态3004(图24)作为下一个路由系统进程进入的程序状态的REQTERM，以及作为访问DNTRAN关系的键的DN 11922。路由系统进程5002于是发送这份RTGEN报文给交换模块202的路由系统进程5004。

为了响应这份RTGEN报文，路由系统进程5004在DNTRAN状态3004进入它的路由程序，并用DN 11922为键、读出DNTRAN关系。按照前述的指定规定，信标DN 11922的查询号翻译信息贮存在交换模块202内，并指明接入组B组员之一的交换模块202的口的全局口身分。因而在交换模块202中访问DNTRAN关系是成功的。取出的DNTRAN元组中的TERMCLASS属性辨明DNTRAN 11922是配给一个多线寻线组而不是一条单独的线。随后路由系统进程5004按照它的路由程序读出PORTGROUP与MHG关系。MHG元组的MODULE属性指明交换模块202为指定的组控制单元。于是读出贮存在LNSTAT关系内的、辨别多线寻线组B的全体组员的忙/闲状态的动态数据。由于交换模块202是组控制单元，访问LNSTAT关系是成功的，而且按预定的寻线算法、给这次呼叫指定一个闲的组员。这个组员立刻被记作忙。按此例情况，假定接入交换模块201的用户话机23是多线寻线组B的一个组员，而用户话机23被指定给这次呼叫。由于路由系统进程5004已经完成了决定终点口的任务，进程5004就发送一份PR报文给中央控制器301的途径寻找系统进程5007。

根据PR报文的ORIGGPI与TERMGPI字段，途径选择

系统进程5 0 0 7 为这次呼叫选定一个交换模块2 2 9与2 0 1都可用的网络时隙，并将辨明选定的网络时隙的信息写入控制存储器2 9以建立时分隔多路交换器1 0 的途径。然后路径寻找系统进程5 0 0 7发出一份在P A T H D E S 字段包括的指定网络时隙的L N T R E Q报文给交换模块2 0 1的终点系统进程5 0 0 5，而呼叫接通步骤的余下部分按以前描述过的方式进行。当用户话机2 3 回到挂钩状态后，终点端进程5 0 0 9 被通知，而发出一份控制报文给组控制单元交换模块2 0 2的呼叫处理控制系统进程(未画出)，使这组的动态数据得以更新。

作为涉及多口寻线组的呼叫接通的第二个例子，假定接入交换模块2 2 9的用户话机2 5 再拨多线寻线组B的信标查询号3 5 5 - 1922° DN 1 1 9 2 2 的查询号翻译信息不在交换模块2 2 9的D N T R A N 关系内(见表2)。因而路由系统程序5 0 0 2(图4 7)在执行路由程序时，用DN 1 1 9 2 2访问D N T B A N 关系是失败的。于是以NOOD4 D3 码1 1 9 为键，读出M O D T R A N 关系。这样就明确了交换模块2 0 2存有所要的查询号翻译信息(见表3)。路由系统进程5 0 0 2编一份包括规定D N T R A N 状态3 0 0 4 为下一个路由系统进程进入的程序状态的R E Q T E R M 字段以及作为读D N T R A N 关系的键的DN 1 1 9 2 2的R T G E N 报文。路由系统进程5 0 0 2 将这份R T G E N 报文发给交换模块2 0 2的路由系统进程5 0 0 4。

作为对这份R T G E N 的响应，路由系统进程5 0 0 4 开始在D N T R A N 状态3 0 0 4 执行它的路由程序。以DN 1 1 9 2 2 为键去访问D N T R A N 关系，正如根据读出交换模块2 2 9的M O D T R A N 关系而期望的那样，是成功的。取出的D N T R A N 的元组中的T E R M C L A S S 属性指明D N T R A N 1 1 9 2 2 被指定给一个多线寻线组而不是给单独的线。随后路由系统进程5 0 0 4 按路由程序读P O R T G R O U P 与M H G 关系。M H G 的元组明确交换

模块202是指定的组控制单元。于是读出贮存在LNSTAT关系中的、辨明多线寻线组B所有组员的忙／闲状态的动态数据。由于交换模块202是这组的控制单元，LNSTAT关系的访问是成功的，而按照预定的寻线算法，给这次呼叫指定了一个闲组员。这个组员立即被标为忙。按照这个第二例，假定接入交换模块229的用户话机26是多线寻线组B的一个组员，而那个用户话机26就被指定给这次呼叫。由于路由系统进程5004能完成决定终点口的任务，进程5004就发送一份PR报文给途径寻线系统进程5007。由于PR报文的ORIGGPI字段与TERMGPI字段两者都指定交换模块229，所以不需要什么时分隔多路交换器10的途径。途径寻找系统进程5007选择一个可用的网络时隙供交换模块229内连接接收时隙交换器50到发送时隙交换器53之间，然后发送一份在PATHDES字段中包括的网络时隙的LNTREQ报文给终点系统进程5010。呼叫接通步骤的余下部分按以前描述的方式进行。

虽然在图32中没有写出来，MHG与TRKG关系还对每个多口寻线组指定交换模块之一作为这组的“影子”组控制单元。该影子控制单元也保持着这一寻线组的动态数据，即它被组控制单元的报文通知所有的忙／闲状态变化。在组控制单元不能工作的情况下，这一组的寻线任务就由这个影子组控制单元来执行。

系统V

系统V（图48）使用了与本文前面描述的系统I（图2）基本上相同的硬件体系结构。但是系统V包括一个另一种控制分配单元311。回忆中央控制器30用经由通信途径49向控制存储器29写入指令的办法来控制时分隔多路交换器10。在系统V（图48）中，中央控制器30”，同样还有交换模块201到229中的每一个，都可用向控

制分配单元3 1' 发送一份控制报文的方式来控制时分隔多路交换器1 0，而控制分配单元3 1' 则用经由一条控制与诊断访问链9 0 4 9向控制存储器2 9写入指令的方式来响应。由于在系统V中，时分隔多路交换器1 0途径寻找与接通功能，从中央控制器3 0" 移入了交换模块2 0 1 到2 2 9，所以交换模块2 0 1 到2 2 9最好能不牵涉到中央控制器3 0" 而直接控制时分隔多路交换器1 0。

系统V 中实现中央控制器3 0" 的处理器，比系统IV（图4 1）的中央控制器3 0' 的费用进一步得到降低。由于系统V的中央控制器3 0" 完全不管接通呼叫时的按每次呼叫进行的处理任务，对该处理器可靠性的要求就明显地降低了，因为即使在中央控制器3 0" 完全失效期间，系统仍能继续工作来交换电话呼叫。

控制分配单元3 1'

控制分配单元3 1' 经由接到时分隔多路交换器1 0的输出端P 6 4的时分隔多路线1 5 0、从交换模块2 0 1 到2 2 9接受控制信息。控制分配单元3 1' 在接到时分隔多路交换器1 0的输入端P 6 4的时分隔多路线1 5 1上向交换模块2 0 1 到2 2 9发送控制信息。在控制分配单元3 1' 内部，时分隔多路线1 5 0与1 5 1都耦合到一个链接口9 0 0 1（图4 9），这个接口基本上跟前面描述的系统I的链接口7 8（图4）完全一致，只不过链接口7 8的抽取与插入控制字的电路、在链接口9 0 0 1中不需要。时分隔多路线1 5 0和1 5 1每根都有2 5 6个通路或时隙。但在目前这个实施例方案中，只需要5 8条通路，交换模块2 0 1 到2 2 9中每一个有两条控制通路。链接口9 0 0 1完成一个信号转换功能，收下时分隔多路交换器1 0在时分隔多路线1 5 0上的信息、将它送到一个报文接口9 0 0 3。链接口9 0 0 1也从报文接口9 0 0 3收到信息，并将它在时分隔多路线1 5 1上发送到

时分隔多路交换器 10。报文接口 9003 将控制分配单元 31' 从时分隔多路交换器 10 收到的控制信息、分配到三族外围处理器上去，一族包括四个模块报文处理器 9201 到 9204，第二族包括四个模块报文处理器 9301 到 9304，和第三族包括一个基础外围控制器 9101。报文接口 9003 也将这几族来的控制信息进行多路调制、传送到时分隔多路交换器 10 里去。这几族经由 32 路串接报文接口总线 9110, 9210 与 9310 跟报文接口 9003 通信。每个模块报文处理器最多跟交换模块 201 到 229 之中的八个结合，并通过一个预定的键水平的规程，例如 HDLC 规程，跟与它结合的交换模块通信。基础外围控制器 9101 用于控制控制分配单元 31' 的许多部件的操作，但特别要用于经由一个控制及诊断访问链 9110 写准备存入控制存储器 29 以规定时分隔多路交换器 10 途径的信息。基础外围控制器 9110 发送到报文接口 9003 的控制指令，经由控制及诊断访问链 9049 写入控制存储器 29。

一个外围接口控制器 9500 以控制外围处理器族之间及中央控制器 30'' 的信息交换的方式、实行包交换功能。按照存储在一个微控制存储器 9501 中的指令运行的外围接口控制器 9500，经由一个 I/O 微处理机接口 9502 跟各族外围处理器通信。接口 9502 经由族数据/地址总线 9100, 9200 及 9300、耦合于各族上。以在一个地址总线 9503 上向 I/O 微处理机接口 9502 写源及终点地址的方法，外围接口控制器 9500 能进行传递控制信息，从一个模块报文处理器，例如 9301，到另一个模块报文处理器或者到基础外围控制器 9101。这种控制信息的传递能在不涉及中央控制器 30'' 的情况下进行。外围接口控制器 9500 能类似地从事从模块报文处理器 9301 到一个 16 位数据总线 9504 的控制信息传递，这个信息随后就被传递到中央控制器 30''。以在一个地址总线 9503' 上向

一个总线接口控制器 9505 写源及终点地址的方法，外围接口控制器 9500 能进行传递总线接口控制器 9505 从中央控制器 30" 接到的控制信息、通过数据总线 9504 和 I/O 微处理机器接口 9502、传到各族外围处理器上去。中央控制器 30" 通过一个双串接通路 32，与控制分配单元 31' 通信。一个选择器 9507 通过通路 12 从中央控制器 30" 接收信息，也通过一个第二双串接通路 32' 从一个备用中央控制器（未画出）接收信息。选择器 9507 选通路 32 或通路 32'，这要根据目前是中央控制器 30" 还是备用中央控制器在控制系统运行。选择器 9507 将从中央控制器 30" 或备用中央控制器收到的信息，从串连格式改为并连格式，并将已改过的信息在一个 32 位并连总线 9506 上传送到总线接口控制器 9505。总线接口控制器 9505 起 32 位选择器 9507 与 16 位外围接口控制器 9500 之间的缓冲器作用。总线接口控制器 9505 包括一个 16 字乘 32 位的 FIFO（未画出），这被分为两个 16 位字段，用于外围接口控制器 9500 的存取。

分布式途径寻找

回忆在一个给定的交换模块，例如模块 201（图 48），与时分隔多路交换器 10 之间，有 512 个时隙或通路。交换模块 201 与时分隔多路交换器 10 之间的两对时分隔多路线，即接到输入／输出端对 PI 的线 13 与 15 和接到输入／输出端对 P2 的线 14 与 16，它们之中每一对上面都有 256 个通路。为了下面讨论的目的，一个给定交换模块与时分隔多路交换器 10 之间的 512 个通路，被称为通路 TS1 到 TS512。时分隔多路交换器 10 能从一个给定交换模块过来的每一条通路接到任何其他交换模块上面相对应的通路上。例如，时分隔多路交换器能将交换模块 201 来的 TS1 接到交换模块 202 到 229

中任一个的T S 1 上去，交换模块2 0 1 来的T S 2 接到交换模块2 0 2 到2 2 9 中任一个的T S 2 上去，等等。不是象系统I 到N 那样，由中央控制器3 0 ” 对时分隔多路控制器1 0 所有的进出通道保持全局的了解并选定呼叫用的网络时隙，在系统V 中不保持这种全局了解。相反，每个交换模块贮存着一个指明从那个交换模块到时分隔多路交换器1 0 的通路T S 1 到T S 5 1 2 中每一条通路的状态的T I M E S L O T 关系，而且在每次模块之间的呼叫的两个当事交换模块就互相商量以选择这次呼叫用的网络时隙。一旦选定了网络时隙，交换模块之一就经由控制分配单元3 1' 向控制存储器发送一个C O N N E C T 命令，而不涉及中央控制器3 0 ” 。

系统V 中，终点口决定功能是以前述系统N 相同的方式完成的。一旦在终点交换模块中建立了终端进程，那个端进程就发出一份途径要求(P R) 报文给现在就在这交换模块中的途径寻找系统进程。途径寻找系统进程响应这份P R 报文而执行的程序的流程图见图5 0 。在方块6 0 1 0 期间收到了P R 报文。执行进行到决策块6 0 2 0 ，在其间要根据P R 报文中指明的终点与起点交换模块是同一或不同的交换模块来判定这次呼叫是否为一个模块间的呼叫。如果这次呼叫不是模块间呼叫执行就转到方块6 0 3 0 。要访问存入这个交换模块的T I M E S L O T 关系，在通路T S 1 到T S 5 1 2 之间选一个可用的作为这次呼叫的网络时隙，并将它标为忙。虽然不用时分隔多路交换器1 0 内的途径，选用的网络时隙仍可用来在交换模块内将接收时隙交换器接到发送时隙交换器。如果换一个情况，这次呼叫是一个模块间的呼叫，执行进程就从决策块6 0 2 0 转到方块6 0 4 0 。在方块6 0 4 0 中编定一个途径寻找要求(P H R) 报文，它用一个包括6 4 个八位位组的字段辨明交换模块与时分隔多路交换器1 0 之间的5 1 2 条通路T S 1 到T S 5 1 2 的可用性。(P H R 报文规定两种忙的通路为不可用，这两种忙通路就

程7013将这份PHR报文发给起点交换模块229中的一个途径寻找系统进程7011。进程7011访问它的TIMESLOT关系，以判明PHR报文中指出的78路可用的通路中哪些在交换模块229与时分隔多路交换器10之间的相对应的通路上可用。在本例中假定有49路这种通路。途径寻找系统进程7011挑选并保留49路通路中的四路作为这次呼叫的候选组，例如通路TS14，TS99，TS349与TS410。途径寻找系统进程7011也选这四条通路之一，例如通路TS99，作为首选(FC)时隙。途径寻找系统进程7011于是编一份规定候选组与FC时隙的候选组(OS)报文，并将这份CS报文发给终点交换模块201中的途径寻找系统进程7013。进程7013访问它的TIMESLOT关系以判定TS99是否忙，而如果不忙，它是否已被指定为另一未解决的呼叫的FC时隙。在本例中假定TS99不忙而且也未被指定为任何其他呼叫的FC时隙。进程7013在它的TIMESLOT关系中将TS99标为忙而将TS14，TS349与TS410标为可用，并编一份CONNECT报文给控制存储器29，文中规定时分隔多路交换器10应在与TS99结合的一个特时隙期间将交换模块201与交换模块229接通。(记住时分隔多路交换器10在256个时隙的基础上工作，但它是用两个时分隔多路线对接到一个给定的交换模块上去的。因此时分隔多路交换器10的每一个时隙是跟通路TS1到TS512中的两路结合的)进程7013也编一份指明被选通路TS99的途径寻找完成(PhC)报文。然后进程7013发那份CONNECT报文给控制存储器29，并发送这份PhC报文给终端进程7009。进程7009的响应是发一份指明选定的网络时隙的SETUPCOMP报文给起点端进程7008。进程7008访问TIMESLOT关系使TS99标为忙、并解放候选组的余下来的时隙，即TS14，TS349，与TS410，使这

转到方块6 240去访问给定交换模块的TIMESLOT关系以判明候选组的四个时隙中是否有任一个是忙的，即是否四个时隙中的任何一个已经被选为另外一个呼叫的网络时隙。任何这种忙的时隙都从候选组中排除出去、不作考虑，然后执行转到决策块6 250。在方块6 250期间，要判明接到的CS报文中指明的FC时隙是否在给定交换模块的TIMESLOT关系中标为忙。当然，如果这个FC时隙是忙的，它不能被选为这次呼叫的网络时隙，执行就转到方块6 310。如果FO时隙不忙，执行就进到决策块6 260，要判明这个FC时隙是否已被给定交换模块与另一交换模块之间的另一未解决的呼叫指定为FC时隙。如果没有，执行转到方块6 270，这个FC时隙被标为忙，候选组中其他时隙则在TIMESLOT关系中标明为可用。然后在方块6 280中编一份CONNECT报文与一份途径寻找完成(PHO)报文，指明这个FO时隙为这次呼叫选定的网络时隙。在方块6 280期间，这份CONNECT报文被送到控制存储器29，而它作为这份CONNECT报文的响应、明确时分隔多路交换器10将在选定的时隙上为这两个交换模块之间提供一条通信途径。还有在6 280期间，PHO报文被送到终端进程，而它知道了给这次呼叫选定的网络时隙以后，能以前述系统N相同的方式完成这次呼叫的接通。

如果换一种情况，在决策块6 260中判明这个FO时隙已经被指定为另一个未解决的呼叫的FO时隙，执行就换到方块6 300上去，从候选组上去掉这个FO时隙，并将它加入在给定交换模块的存储器存着的一个称为冲突的表上。冲突表是候选组中被指定为FO时隙而因为它已被指定为几乎同时提出的其他未解决的呼叫的FO时隙而未选上的时隙的表。无论这个FO时隙是在方块6 250中判明是忙的、或者在方块6 260中发现它已被指定为另一个未解决的呼叫的FO时隙，结果都是一样，即不选这个FO时隙为这次呼叫的网络时隙，而执行转到

决策块6310。在决策块6310中要判明候选组现在是否空了。如果候选组不空，执行就转到方块6340，而候选组中剩下的时隙之一被选为新的FO时隙。在方块6350中，编一个候选组减员(CSR)报文，指明新的FC时隙以及在执行这次程序的方块6240期间从候选组中删除的任何时隙。编出的CSR报文在方块6360期间被传送至对方交换模块中的途径寻找系统进程。

但如果在决策块6310期间判明候选集是空的，执行就改到决策块6370上去。N是一个存入的表示对一次给定的呼叫，候选集已经变空了几次的变量。在决策块6370中要判明候选集是否已经空了的预定次数，例如空过三次。若是如此，执行就转到方块6430而这次呼叫失败。但若候选组空了不到三次，执行就转到决策块6380，在此期间要判明冲突表是否也是空的。如果冲突表是空的，执行就再次转到方块6430而这次呼叫失败。但如果冲突表示不空，执行就转到方块6390，在此期间冲突表就成为这次呼叫的新候选组。在方块646400期间，冲突表于是就被清除，而在方块6410期间，变量N加一。然后象以前描述的那样、执行方块6340，6350与6360，并发送一份CSR报文给对方交换模块的途径寻找系统进程。

一个给定交换模块中的途径寻找系统进程，以执行与图52到54相同的程序来响应一个CSR报文。这份CSR报文在方块6220中收到。然后在方块6230期间，被对方交换块作为忙而除去的、而在CSR报文中提到的那些时隙，都解放了使它们随后包括在给定的交换模块所选的候选组中。于是执行转到方块6240，并以前述关于OS报文的同样方式继续下去。

应当注意到、对任何给定呼叫、可以由终点交换模块、也可以由起点交换模块来选定网络时隙。如果由终点交换模块选择，那个模块的途径寻找系统进程就发一份CONNECT报文给控制存储器29，也发

一份途径寻找完成 (PHC) 报文给终点端进程。在此实施方案中，如果由起点交换模块选择，起点交换模块中的途径寻找系统进程发出一份 PHC 报文给终点交换模块中的途径寻找系统进程。于是终点交换模块中的途径寻找系统进程按好象这个网络时隙是它自己选出来的那样进行当起点交换模块知道已经给这些呼叫选定了一个网络时隙，就将候选组中包括的其他组员标为可用，他们就可被起点交换模块选的另一个候选组引用。

作为系统V 的呼叫接通的第一个例子，再回到系统N的图4 3 的例子上来。假定接在交换模块2 2 9 上的用户话机2 5 拨出3 5 5 -2285 的号。引到终点口决定的初始步骤跟系统N的例子一样。建立一个起点端进程7 0 0 8 (图5 6)，它发出一份RTBEQ报文给路由系统进程7 0 0 2。进程7 0 0 2 不能完成终点口的决定，但读了MODTRAN 关系 (图3 2) 以后、判明所要的查询号翻译信息是贮存在交换模块2 0 2 内。于是进程7 0 0 2 发出一份RTGEN报文给交换2 0 2 中的路由系统进程7 0 0 4，而它判明了终点口是接在交换模块2 0 1 的用户话机2 3 上。进程7 0 0 4 不是象系统N那样发一份报文给中央控制器3 0、而是发一份RTGEN报文给交换模块2 0 1 中的终点系统进程7 0 0 5。进程7 0 0 5 判明用户话机2 3 是闲的，建立一个终点进程7 0 0 9，并发出一份LNTERM报文给进程7 0 0 9。作为响应，进程7 0 0 9发一份途径要求 (PR) 报文给也在交换模块2 0 1 内的一个途径寻找系统进程7 0 1 3。途径寻找系统7 0 1 3 根据PR 报文中的ORIGGPI 及TEMGPI 字段，判明这次呼叫是一个模块间的呼叫。因而进程7 0 1 3 访问它的TIMESLOT 关系、而编一份辨明交换模块2 0 1 与时分隔多路交换器1 0 之间的512条通路 TS 1 到TS 512 的可用性的途径寻找要求 (PHR) 报文。在本例中假定这些通路中有78 路可用，也就是既不忙、又没被保留。于是进

些时隙又能被途径寻找系统进程7011选为候选组的一部分。呼叫接通的余下部分按系统N的同样方式进行以建立用户话机25与23之间的通信。

作为第二个例子，设想下面是前一例的变化。当交换模块201中的途径寻找系统进程7013收到明确的包括TS14, TS99, TS349与TS410的候选组并指定TS99为FC时隙的CS报文时，查明交换模块201与时分隔多路交换器10之间的TS99变成忙的了。换句话说，在这份PHR报文传到交换模块229之后、TS99已被另一次呼叫选中了。进程7013将TS99从候选组中除去，并选TS14作为新的FO时隙。进程7013于是向途径寻找系统进程7011发一份候选组减员(CSR)报文(图56中未示出)，指明TS99应在交换模块201内解放使它被另一呼叫选入候选组，并指明TS14是新的FO时隙。但是假定虽然TS14还不是忙的，交换模块229却刚刚从某个另外的交换模块，例如交换模块208，接到一个CS报文，它已经选定TS14为一个FO时隙。这个事件称为冲突。为了建立模块229与201之间的呼叫，TS14被放在所谓的一个冲突表上。途径寻找系统进程7011选一个新的FO时隙，例如TS410，并向途径寻找系统进程7013发一份CSR报文(图中未示出)，指定FO时隙TS410，同时指明TS14应从候选组除去并加入冲突表。作为例子，设想TS410与TS349正好都在交换模块201中变为忙。由于候选组现在已经变空，进程7013就用冲突表作为新的候选表，而在此例中、冲突表中只有TS14。然后冲突表被清除。进程7013发一份包括指定的FO时隙TS14的CSR报文(图中未示出)给交换模块229中的进程7011。假定TS14在交换模块229正好变为可用。进程7011于是选定TS14作为这次呼叫的网络时隙，并发出一份包括所选的网络时隙的途径

寻找完成(PHO)报文(图中未示出)给进程7013。呼叫接通的余下部分如前所述进行。

在系统V中，顺序呼叫按以前描述的系统Ⅲ同一方式处理。途径尋找要等到最后终点口已经决定而且这次呼叫已被简化为简单呼叫后才开始。

在系统V中，查询号翻译功能和多口寻线组控制功能，均按前面描述的系统N同一方式执行。

系统V的第一个另外的实施方案

在刚才描述的系统V中，每个交换模块都能同时跟许多别的交换模块协商多个呼叫的途径。在系统V的第一个另一种实施方案中，每个交换模块在一个时间只参加一个呼叫途径寻找。在第一个另外的实施方案中，一旦终点交换模块为一次给定的呼叫发出一个标明该交换模块与时分隔多路交换器10之间的空闲通路的途径寻找要求，除非等到这次给定的呼叫的网络时隙已被选定时、不再发途径寻找要求。终点交换模块也搁下对任何进来的途径寻找要求，直到这次给定的呼叫的网络时隙已被选定。因此起点交换模块能在替这次给定的呼叫发出的途径寻找要求中选定任何空闲通路并有把握使所选的通路从终点交换模块来时仍是空闲的。一旦为这次呼叫选定了通路，起点交换模块就向终点交换模块通知这个选定的通路，而终点交换模块于是就能自由地为下次呼回答进来的途径寻找要求和发出途径寻找要求。如果终点交换模块在发出一个途径寻找要求后的一个预定的时期内没有被通知为这次呼叫选定了一条通路，就首先回答从其他交换模块送来的途径寻找要求，然后重发一个途径寻找要求，当然是经过更改反映通路状态的任何变动的。

系统V的第二个另外的实施方案

在系统V的第二个另外的实施方案中，那512条通路TS1到

T S 5 1 2 被分为四组：组1 中包括通路 T S 1 到 T S 1 2 8，组2 包括通路 T S 1 2 9 到 T S 2 5 6，组3 包括通路 T S 2 5 7 到 T S 3 8 4 而组4 包括通路 T S 3 8 5 到 T S 5 1 2。第二个另外的实施方案跟第一个另外的实施方案类似，只不过一个给定的交换模块能同时替四个呼叫寻找途径 - 每个通路组一个。在上述第二个另外的实施方案中、每个途径寻找要求只标明一个通路组内的空闲通路。一旦一个终点交换模块为一次给定的呼叫发出一个标明这个交换模块与时分隔多路交换器 1 0 之间的，例如空闲组 1 的通路的途径寻找要求，它不再发标明组 1 通路的途径寻找要求、直到对给定呼叫的网络时隙被选定为止。终点交换模块也搁下送来的标明组 1 的通路的途径寻找要求，直到对给定呼叫的网络时隙被选定为止。这样起点交换模块能在这一次给定呼叫的途径寻找要求标明的空闲组 1 通路中任选一个而有把握使所选的通道从终点交换模块来的仍然是空闲的。途径的寻找在每个其他组内以类似的方式独立地进行。

不言而喻前述的一些交换系统只不过是本发明的原理的示范，而本领域的一般技术的人员可以设想其他实施方案，但并没有离开本发明的精神与范畴。

附图简写字表

图1 页 1 / 52

RTGEN 通用路由要求

图59 页 1 / 52

RDBLK	路由数据块
CFBLK	呼叫流程块
GRPEBLK	组 块
TERMBLK	终点块
FIXEDRI	固定路由指标
SCRNING	筛选
DNTRAN	查询号翻译
ROUTING	路 由
PORTGROUP	口的组
GROUPOPORT	组的口
MHG	多口寻线组
LNSTAT	话线状态
TRKG	干线组
TKOWNER	干线主
TKQUE	干线排队

图 17 / 18 页 16 / 52

H TYPE	寻线类型
G MFLAG	组员标
NIM	下一个闲组员

图 20 页 18 / 52

CC	中央控制
CDU	中央分配单元
CM	控制存储器
DFI	数字设施接口

图 21 页 19 / 52

FIU	设施接口单元
-----	--------

图 24 页 21 / 52

SWITCH	交換
--------	----

图 32 页 29 / 52

REQTERM	要求终点
RTCONTDA	路由继续数据
ORIGGPI	起点全局口身份
TERMGPI	终点全局口身份
PR	途径要求
RTSEQ	途径顺序

图 8 页 8 / 52

FIXEDRT	固定路由
DNTRAN	查询号翻译
MLGPREHUNT	多线寻线组予寻找
MLGHUNT	多线寻线组寻线
MLGBUSV	多线寻线组忙
RTING	路由
TRKPREHUNT	干线予寻找
TRKHUNT	干线寻线
TRKBUSY	干线忙
INTEGRITY	完整性
NW-CONN	网络接线
DONE	完毕
START	开始
SCREEN	筛选

图 14 / 15 页 14 / 52

RTREQ	路由要求
PATHDES	途径描述语
RTGDATA	路由数据
DIALDATA	拨号数据
GPI	全局口身分
TREAT	处理

FARPID	远进程身分
DIGDATA	数字数据
HEADER	板头
TEXT	报文
ORIGTPI	起点端进程身分
RTGSTATE	路由程序状态
RICOUNT	路由指标计数
GRPNUM	组号
MODULE	模块
PORT	口
MEMBER	组员

图 16 页 15 / 52

RI	路由指标
DI	终点指标
SI	筛选指标
PI	前缀指标
NOC	标准局码
ROUTETYPE	路由类型
DN	查询号
TERM CLASS	终点类别
SECRI	二次路由指标

图 5 9 页 1 / 5 2

TKSTAT

干线状态

MODTRAN

模块翻译

图 2 页 2 / 5 2

CU

控制单元

LU

话线单元

TU

干线单元

图 7 页 7 / 5 2

CPCSP 呼叫处理控制系统进程

OTP 起点端进程

RSP 路由系统进程

TSP 终点系统进程

ITP 终点端进程

LNTREQ 话线终点要求

TKTREQ 干线终点要求

ANTREQ 宣告线终点要求

LTERM 话线终点

TKTERM 干线终点

图 7 页 7 / 5 2

ANTREQ 宣告线终点

SETUPCOMP 接通完成

SWREQ	交换要求
MODULE	模块
PORSTATUS	口状态
BUSY/IDLE	忙/闲
CF	呼叫转发
SC	依次接通
MODTRAN	模块翻译(见前图59,页1/52)
NO CD4D3	标准局码十千位码+百位码

图 37 页 33 / 52

RERTE	重定路由
CFTP	呼叫转发端进程

图 43 页 39 / 52

PHSP 途径寻找系统进程

图 49 页 45 / 52

MMP	模块报文处理器
FPC	基础外围控制器

图 54 页 46 / 52

PHC	途径寻找完成
PHR	途径寻找要求

图 5 1 页 4 7 / 5 2

CS 候选组

SM 交换模块

图 5 2 页 4 8 / 5 2

CSR 候选组减员

图 5 3 页 4 9 / 5 2

CONNECT 接线

说明书中简写字表

页 3 7

RAM 随机存取存储器

页 4 0

NOC 标准局码

页 3 3

DIG CNT 拨号位数

页 7 6

DN 查询号

页 7 a

D 4 千位数
D 3 百位数

页 9 0

FIFO 先进先出

页 9 8

TS1—TS512 时隙 1 至 时隙 512
T S 时隙
TIME SLOT 时隙

页 1 0 0

F C 首选

补正 8 6 1 0 1 3 2 9

文件名称	页 行	补正前	补正后
说明书	1	发明名称 用分布式数据库在分 布式控制交换系统中 的终点口确定	……在分布式控制交 换系统中终点口的确 定

说 明 书 附 图

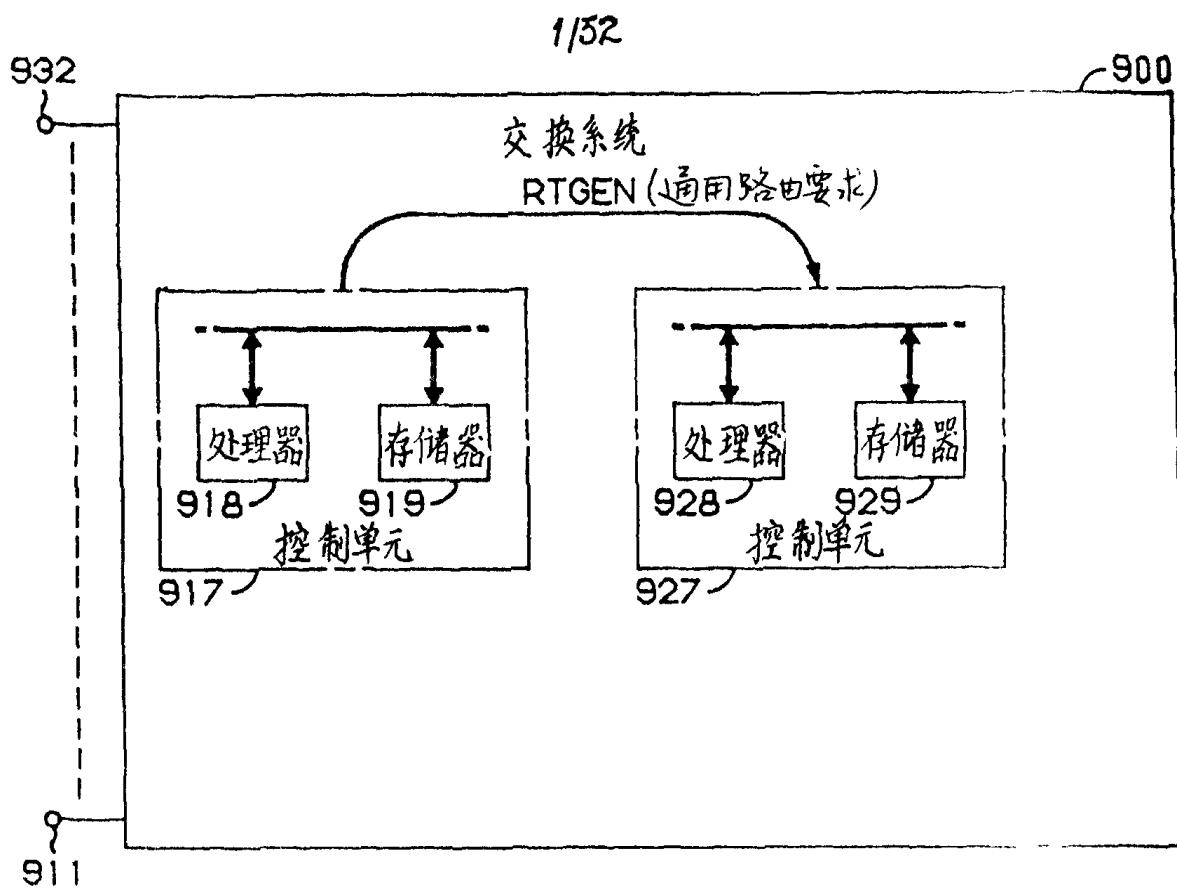


图. 1

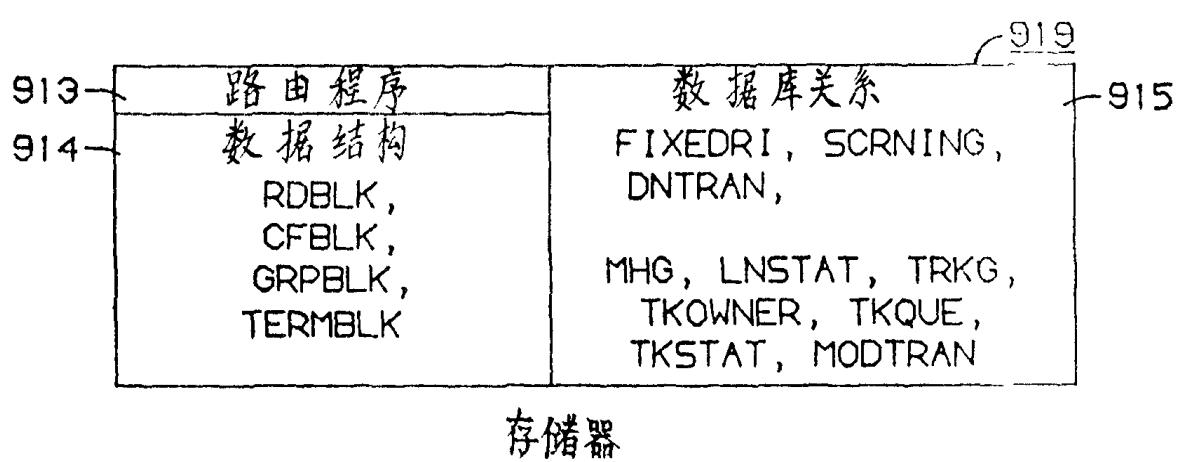
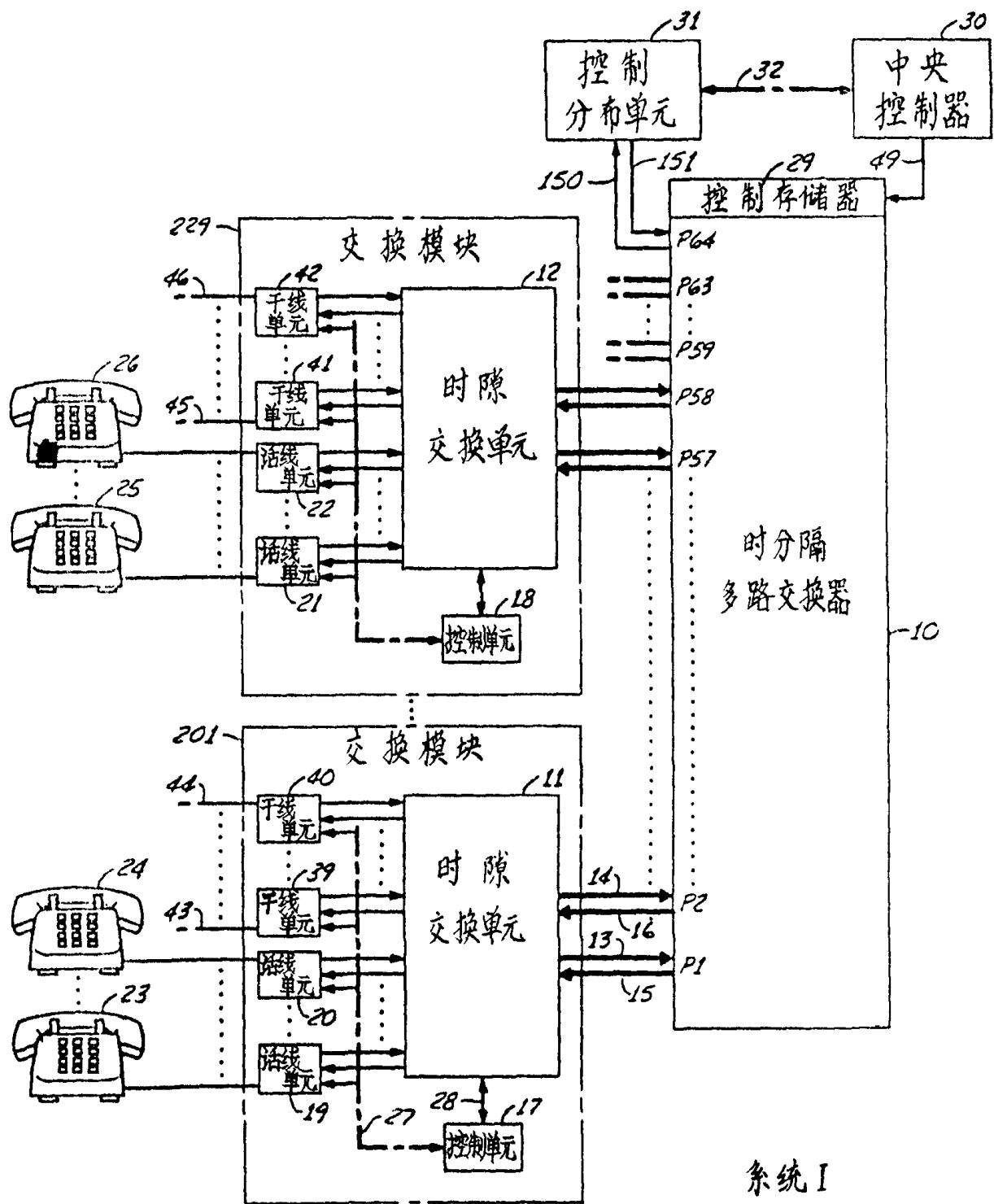


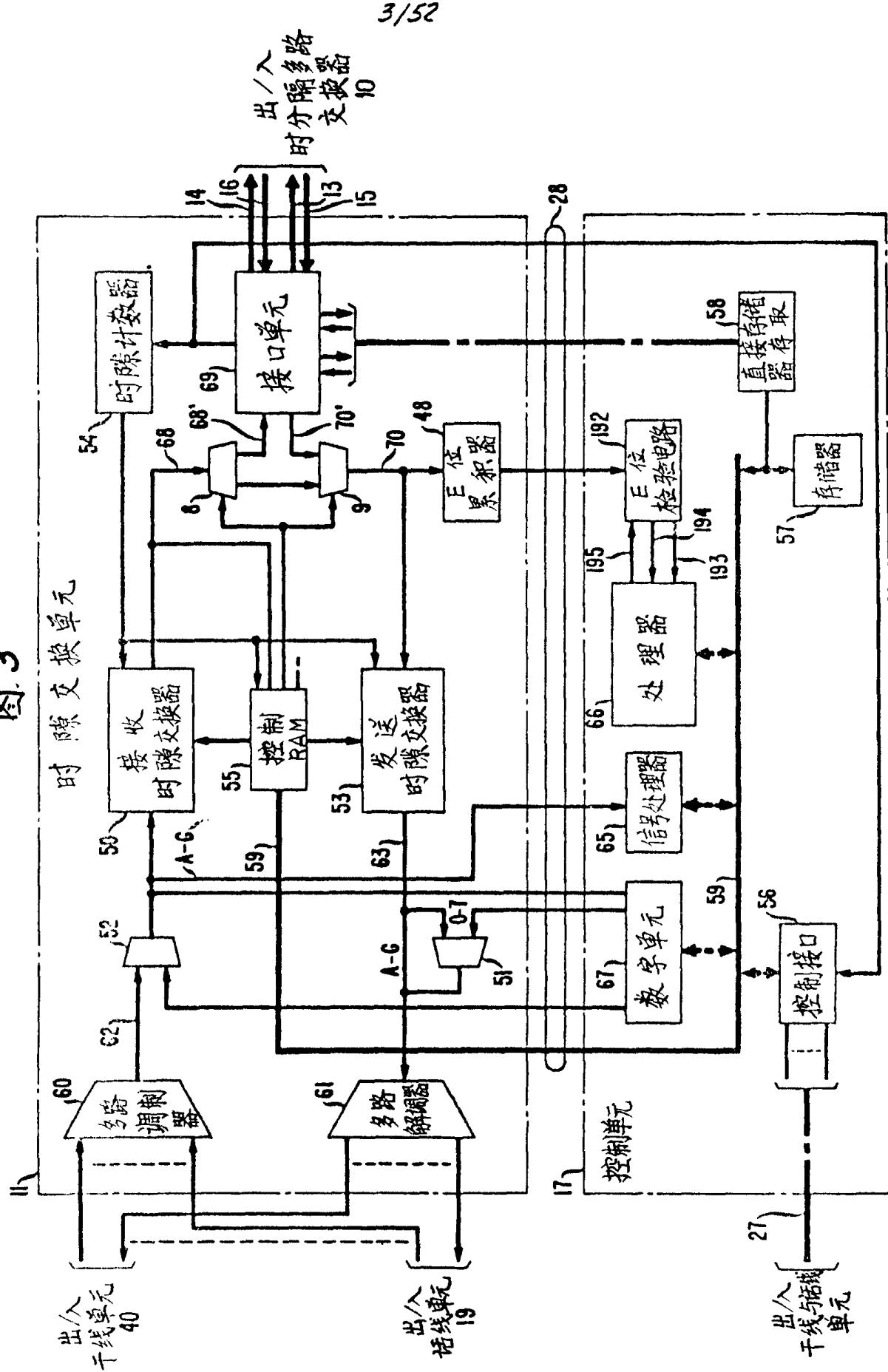
图. 59

图 2



系统 I

图 3



4

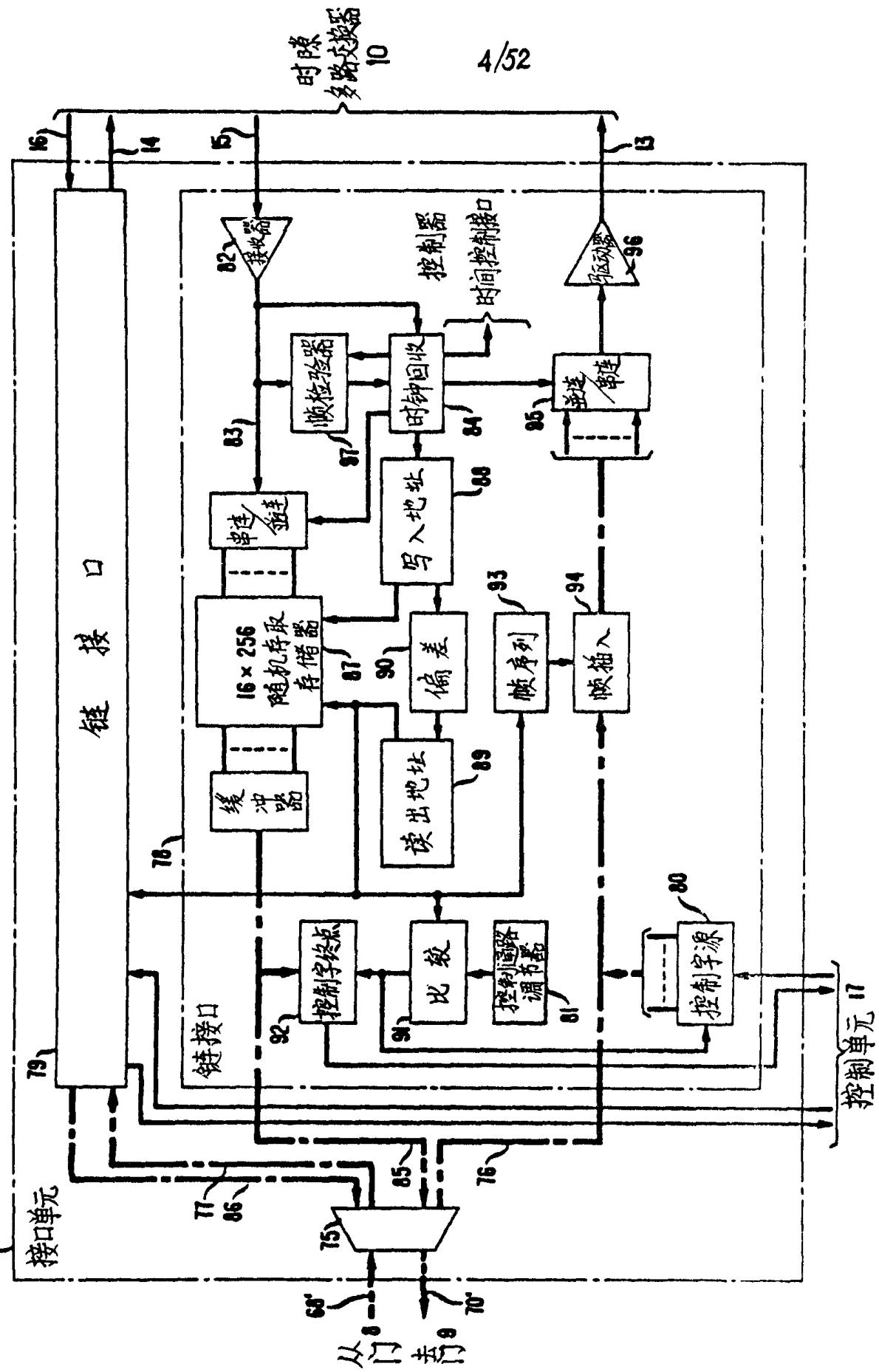


图. 5

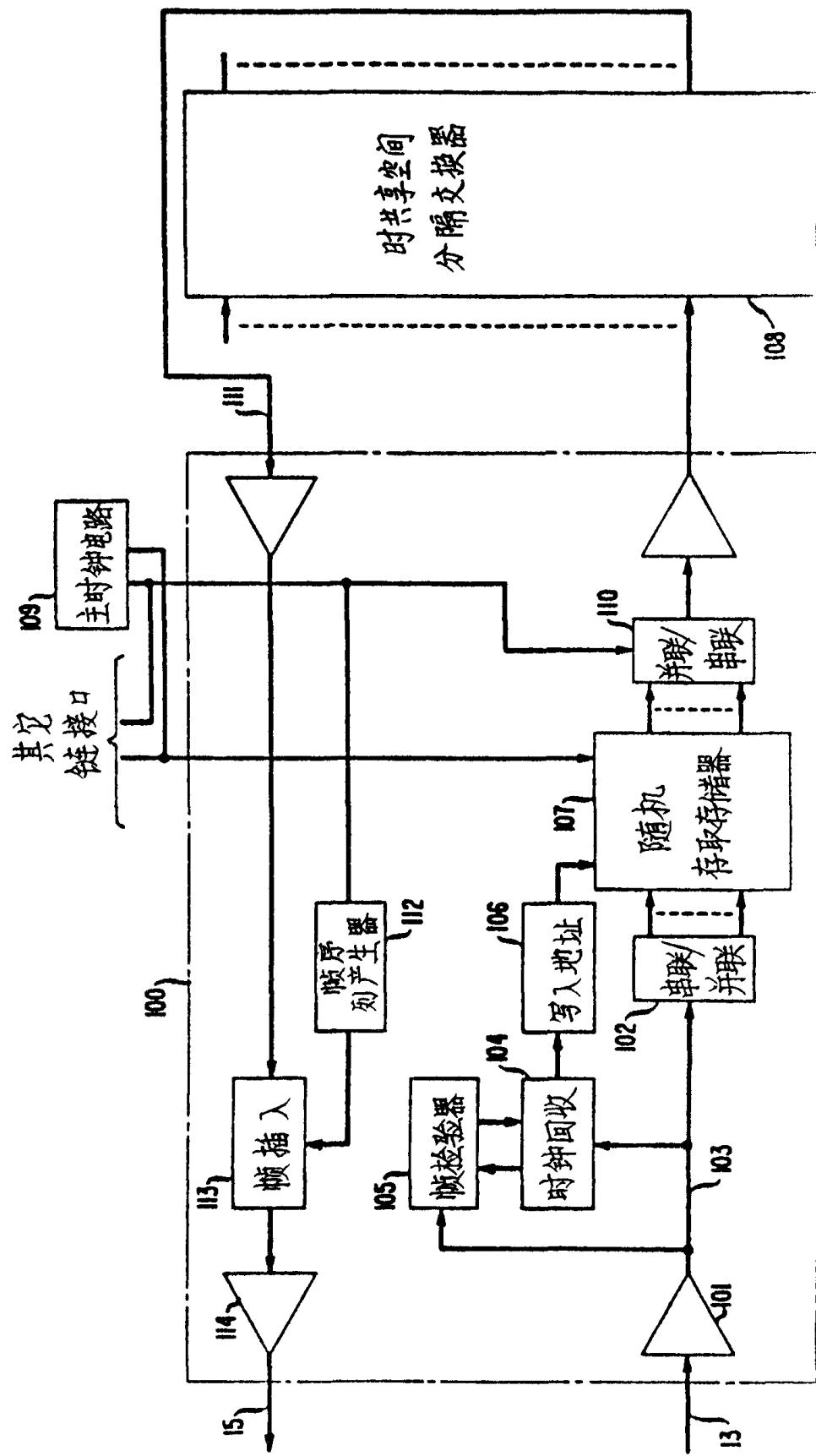
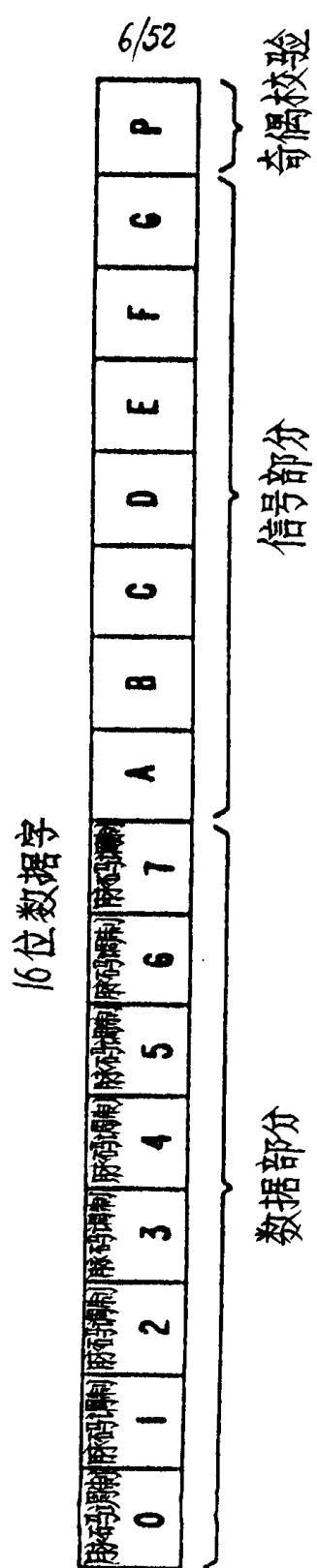
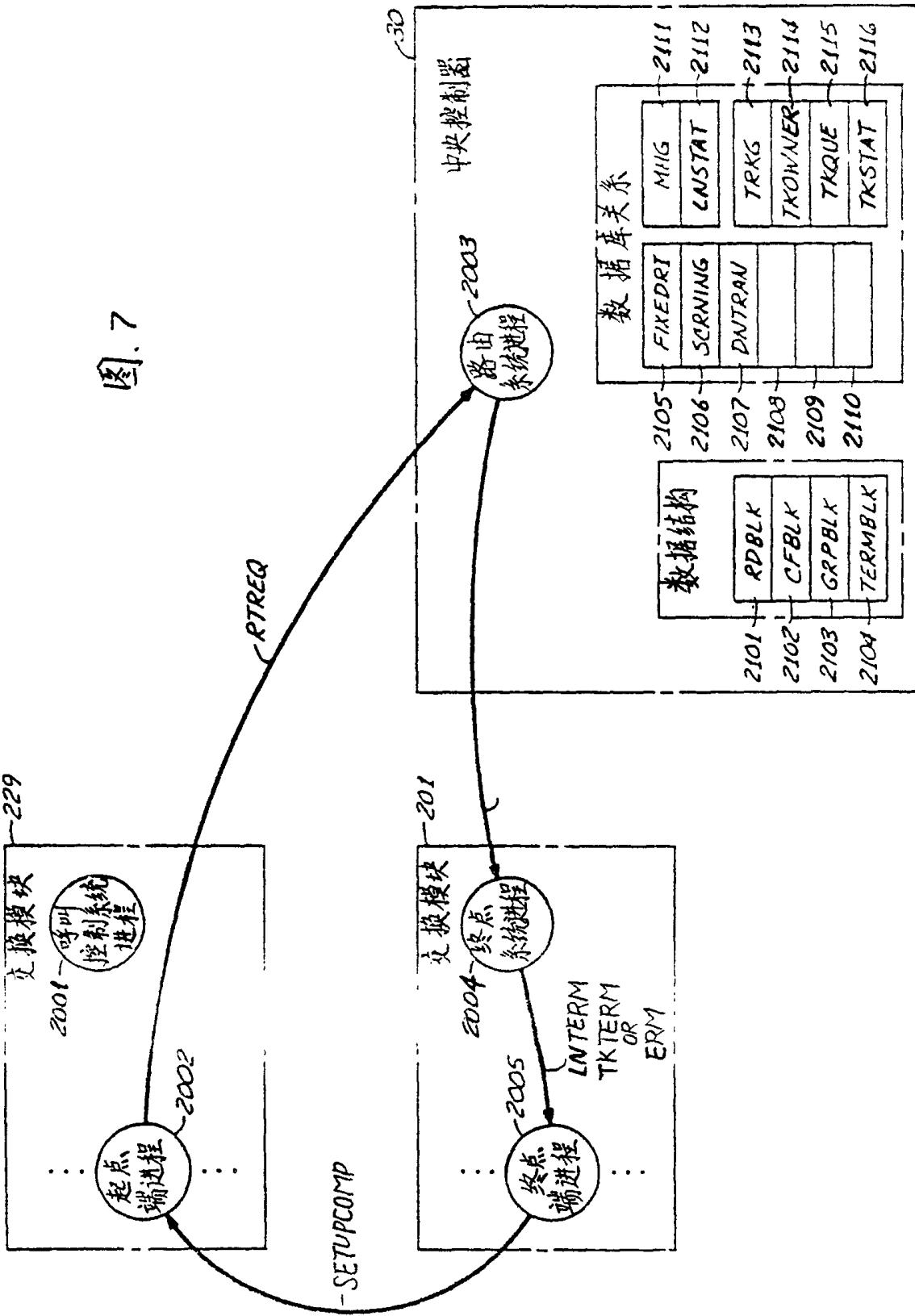


图. 6





8/52

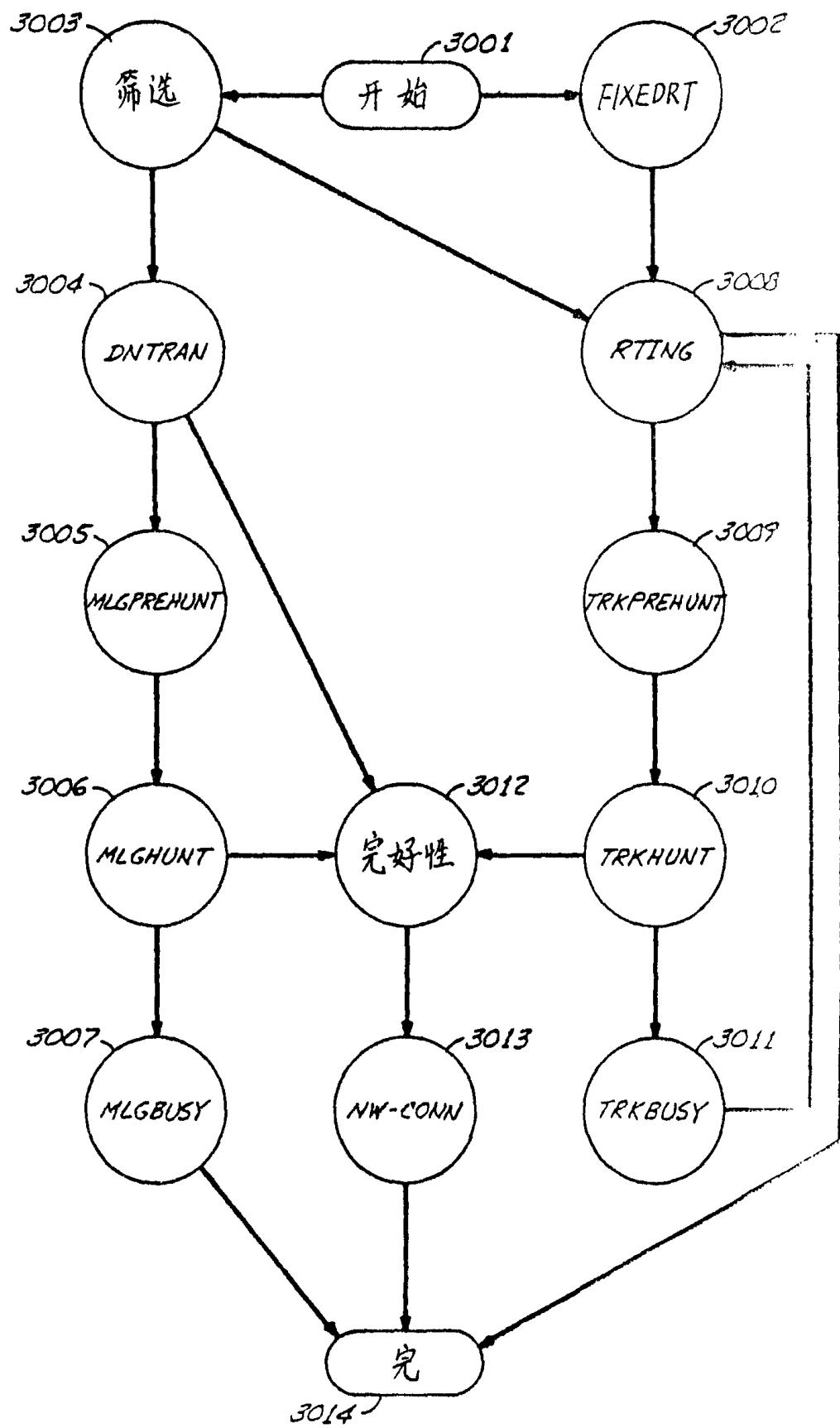
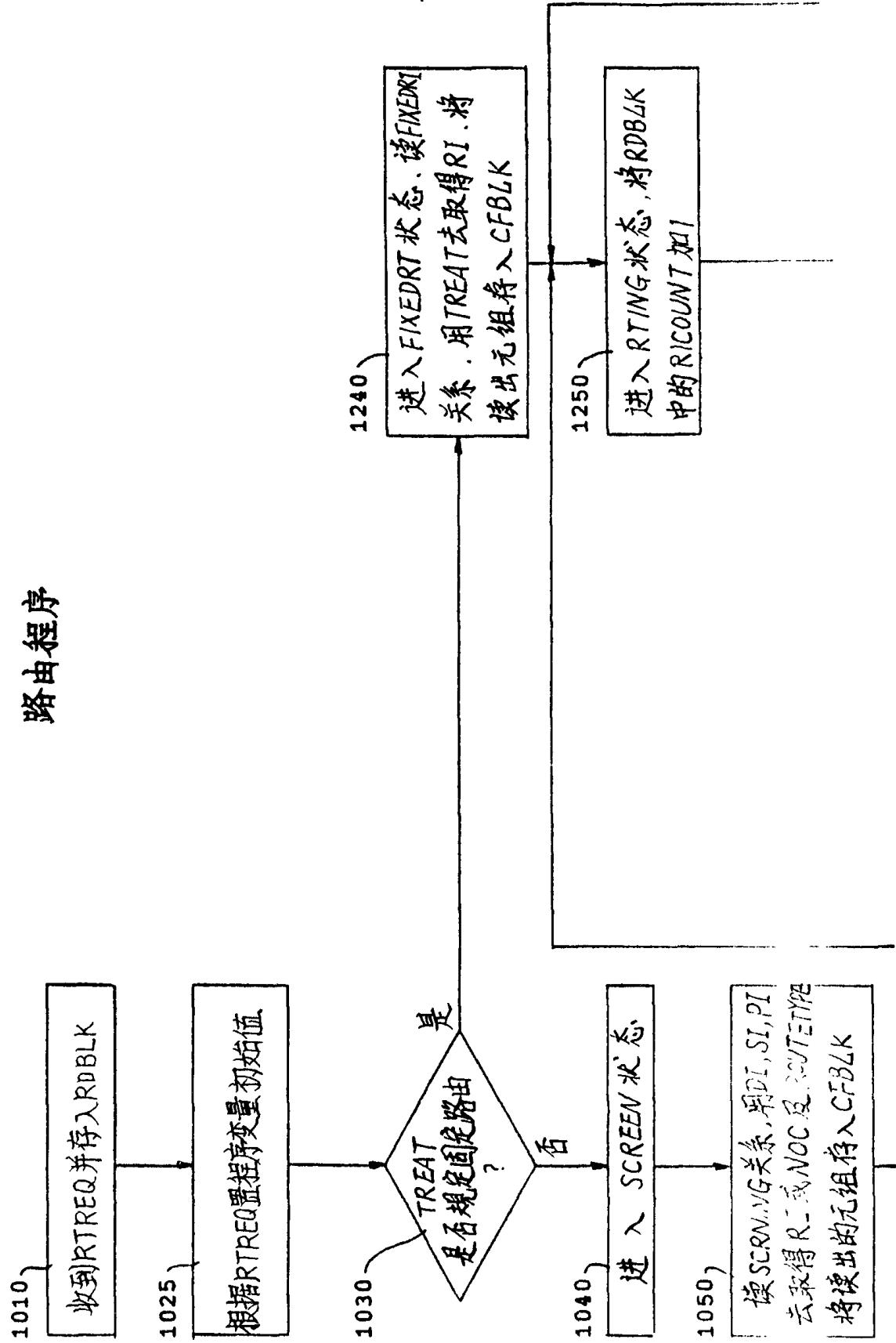
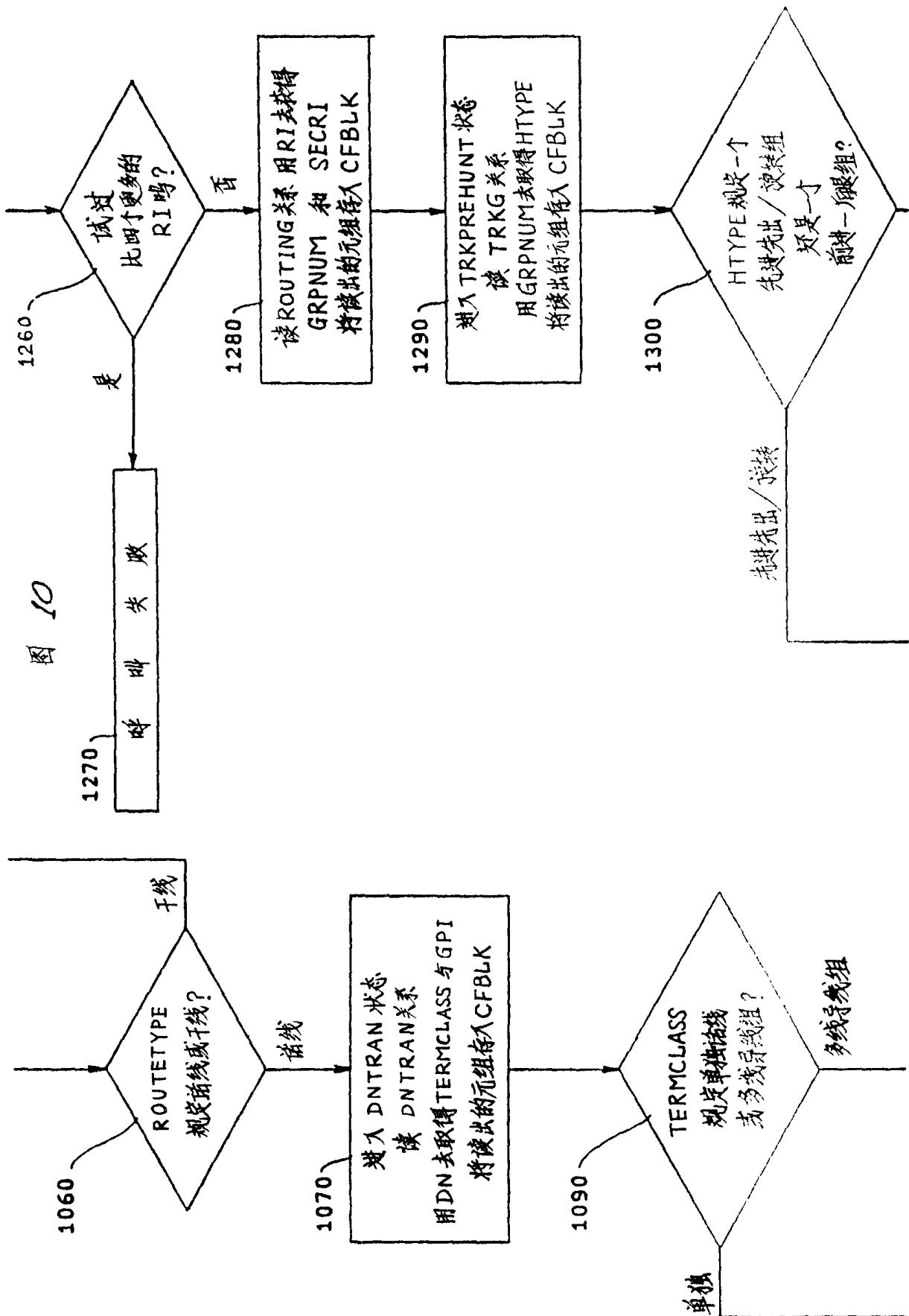


图 9

路由程序



10



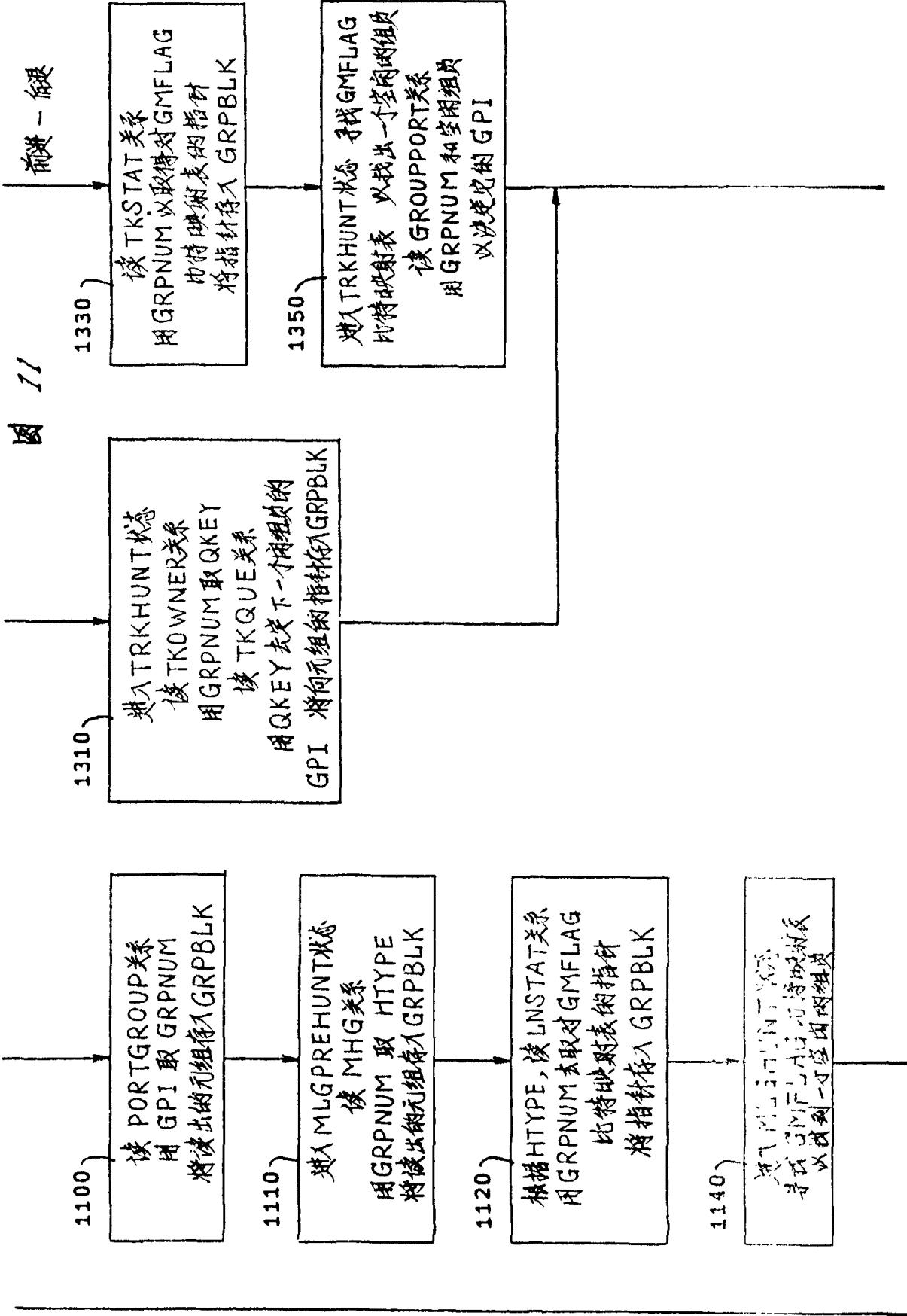


图 12

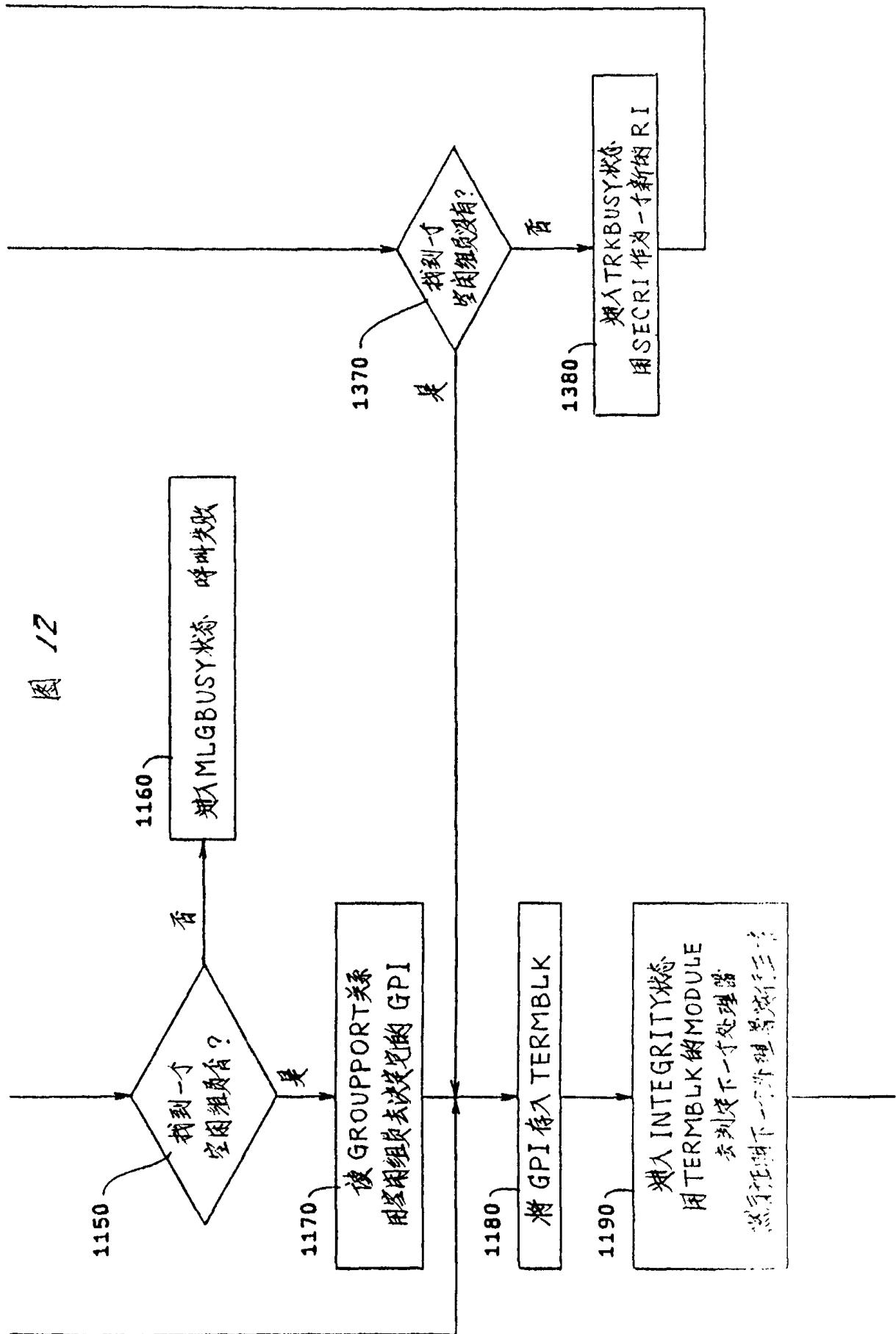
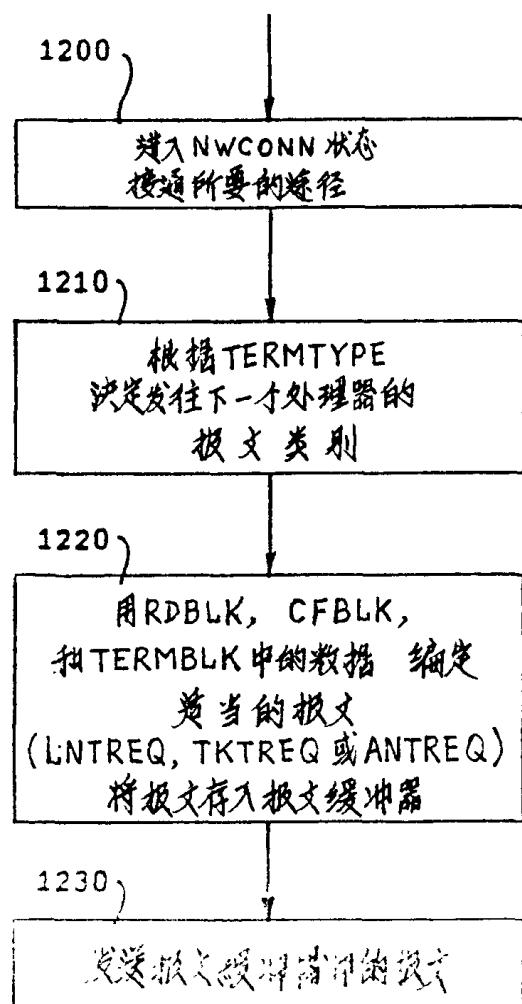


图 13



RTREQ

PATHDES, RTGDATA, DIALDATA, GPI, TREAT, ...

LNTREQ

PATHDES, RTGDATA, FARPID, GPI, ...

TKTREQ

PATHDES, RTGDATA, FARPID, GPI, DIGDATA, ...

ANTREQ

PATHDES, RTGDATA, FARPID, GPI, ...

图.14

RTREQ 报文RDBLK

HEADER, TEXT, ORIGPI, RTGSTATE, RICOUNT, ...

CFBLK

FIXEDRI, SCRNING, DNTRAN, ROUTING, ...

GRPBLK

PORTGROUP, MHG, LNSTAT, GROUPPORT, TRKG, TKOWNER, TKQUE, TKSTAT, ...

GPITERMBLK

MODULE, PORT, GRPNUM, MEMBER, ...

图.15

LS/52

FIXEDRI

TREAT, RI, ...

SCRNING

DI, SI, PI, RI, NOC, ROUTETYPE, ...

DNTRAN

DN, TERMCLASS, GPI, ...

ROUTING

RI, GRPNUM, SECRI, ...

PORTRGROUP

GPI, GRPNUM, MEMBER, ...

GROUPPORT

GRPNUM, MEMBER, GPI, ...

图. 16

16/52

MHG

GRPNUM, HTYPE ...

LNSTAT

GRPNUM, GMFLAG ...

图. 17

TRKG

GRPNUM, HTYPE ...

TKOWNER

GRPNUM, QKEY ...

TKQUE

QKEY, GPI, NIM ...

TKSTAT

GRPNUM, GMFLAG ...

图. 18

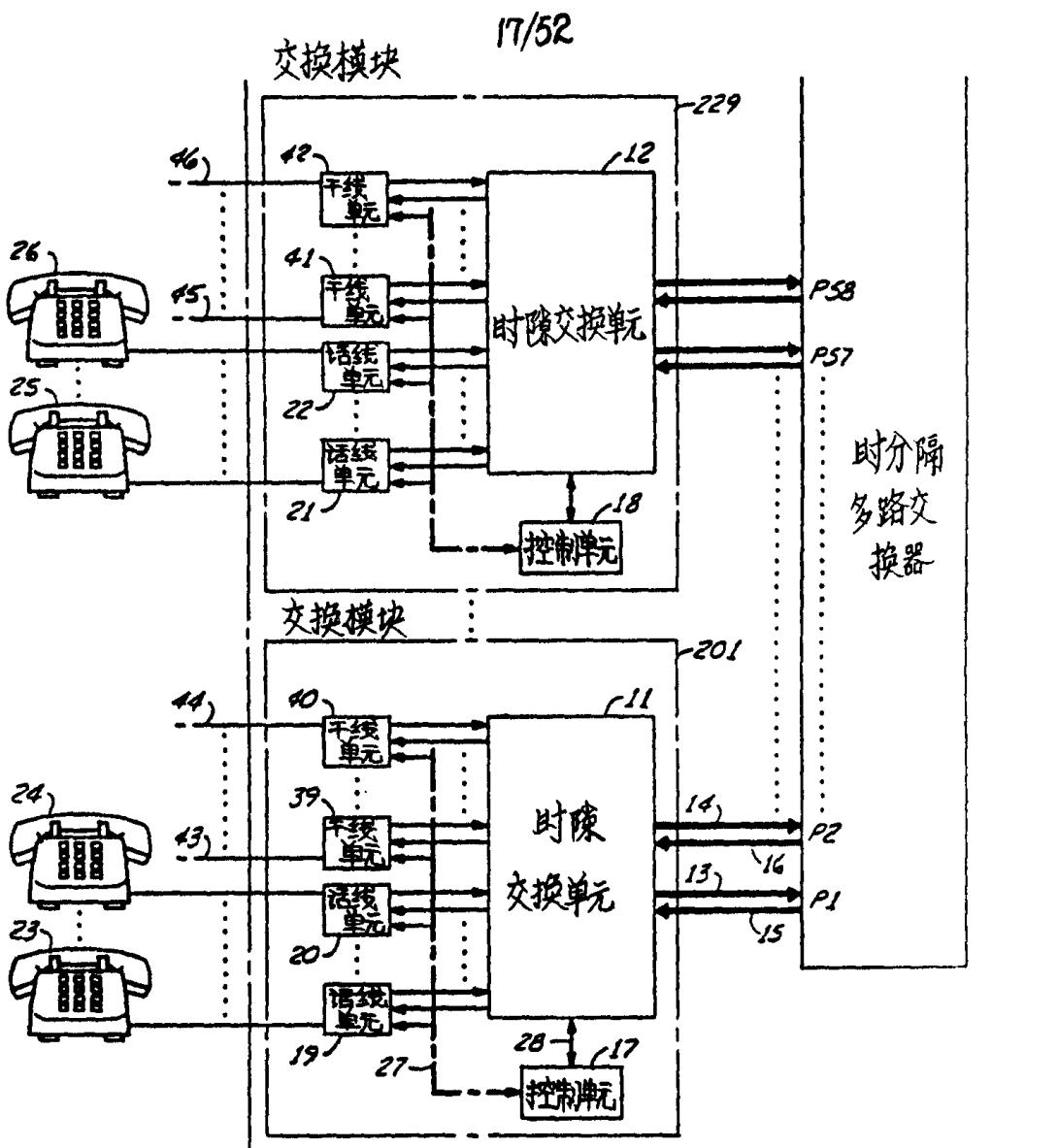


图 19

系统Ⅱ

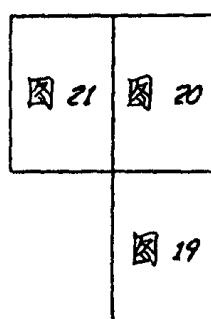


图.22

图 20

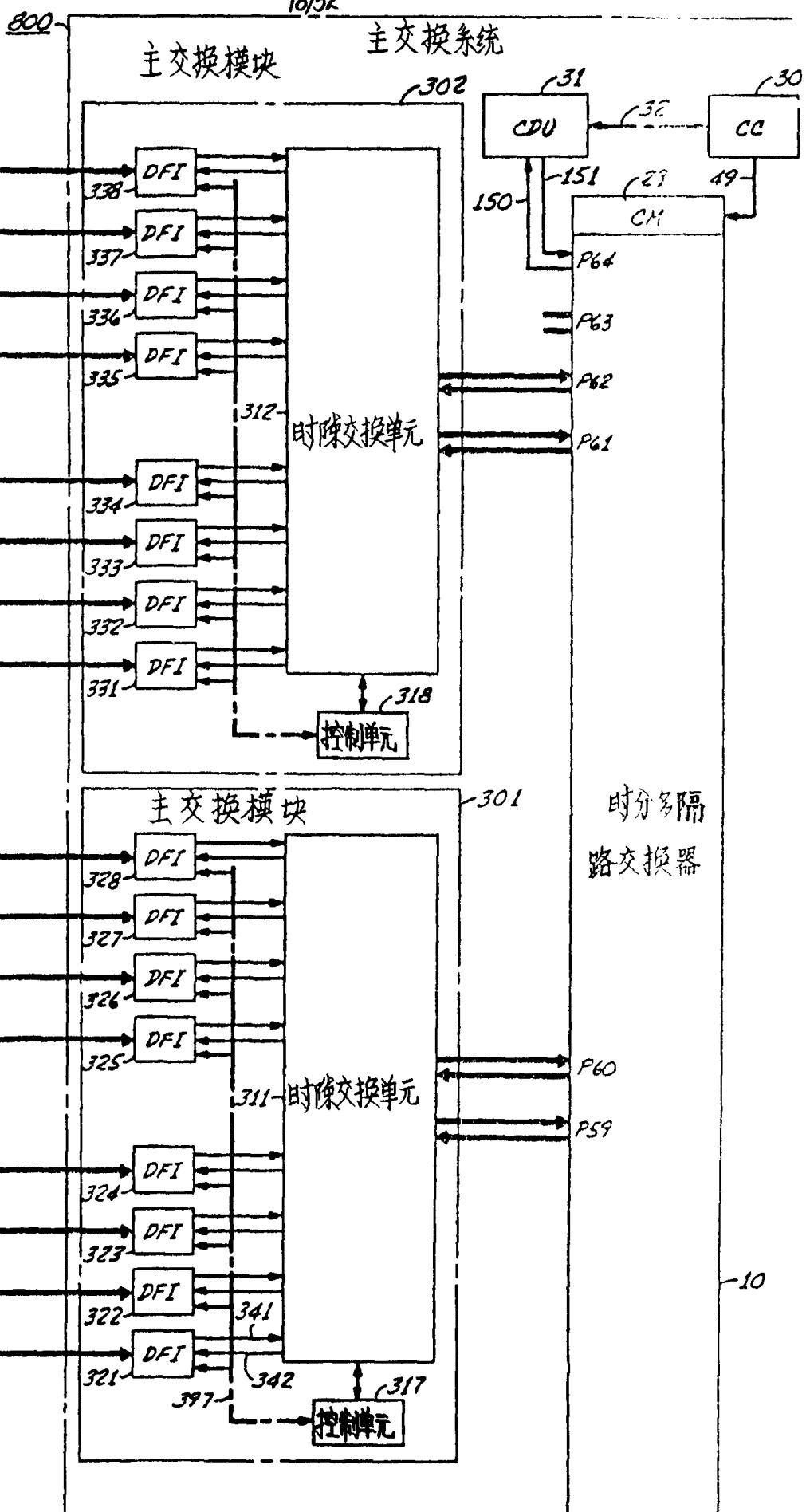
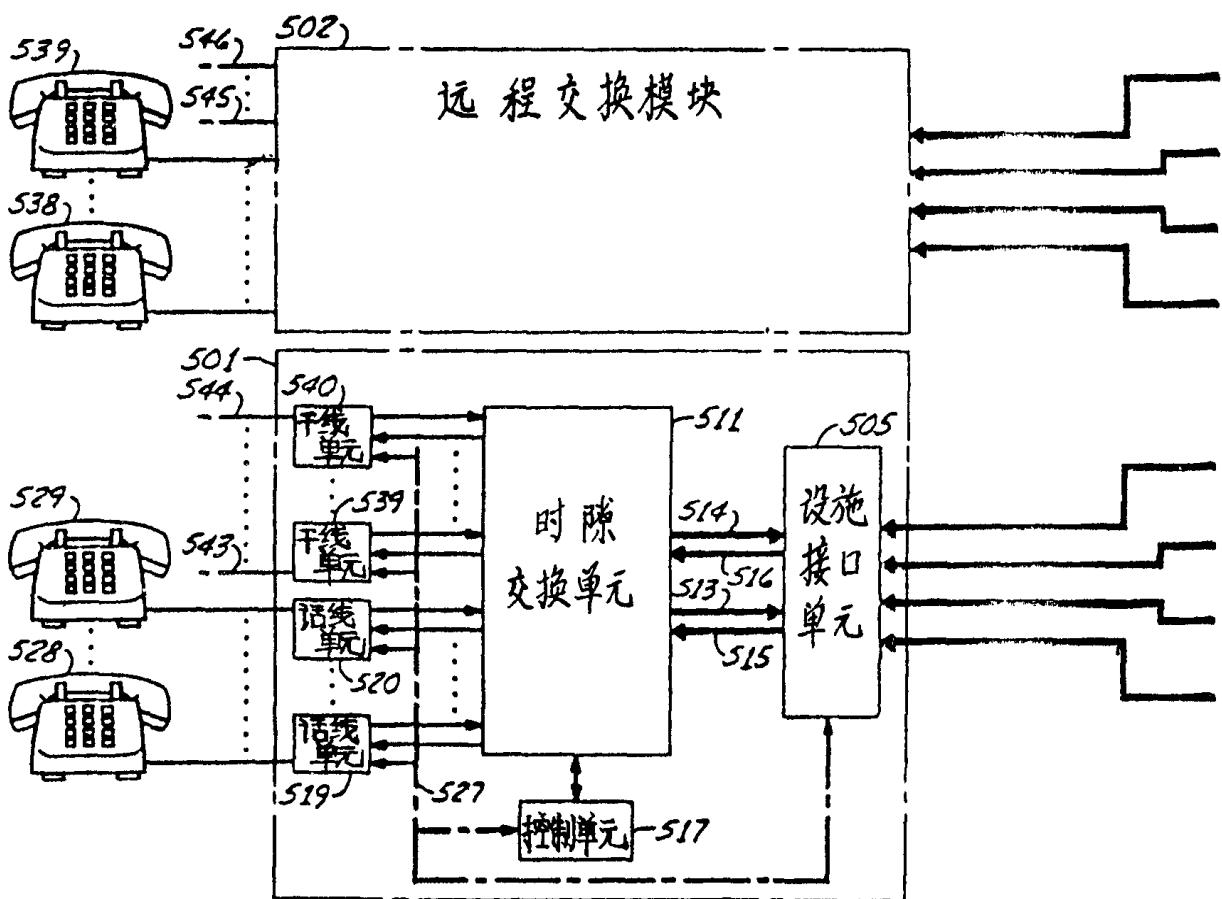
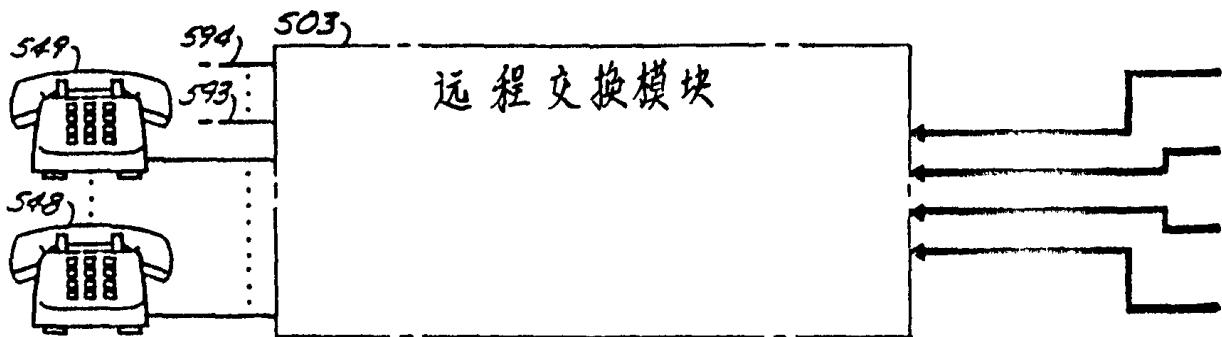
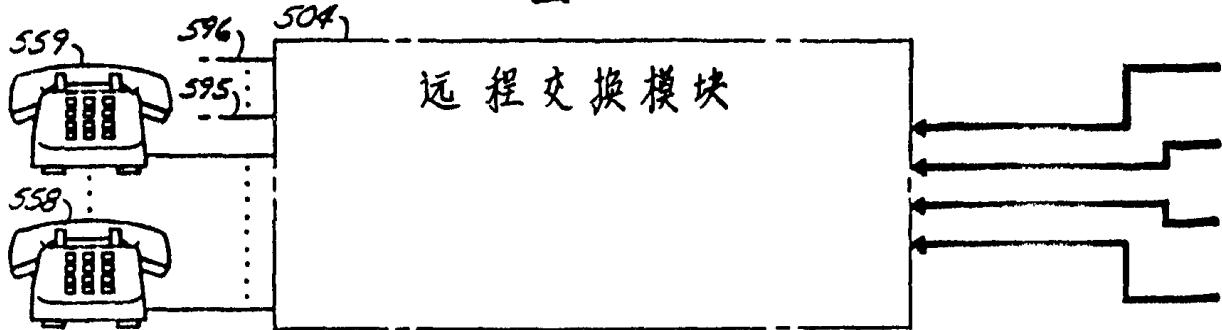


图.21



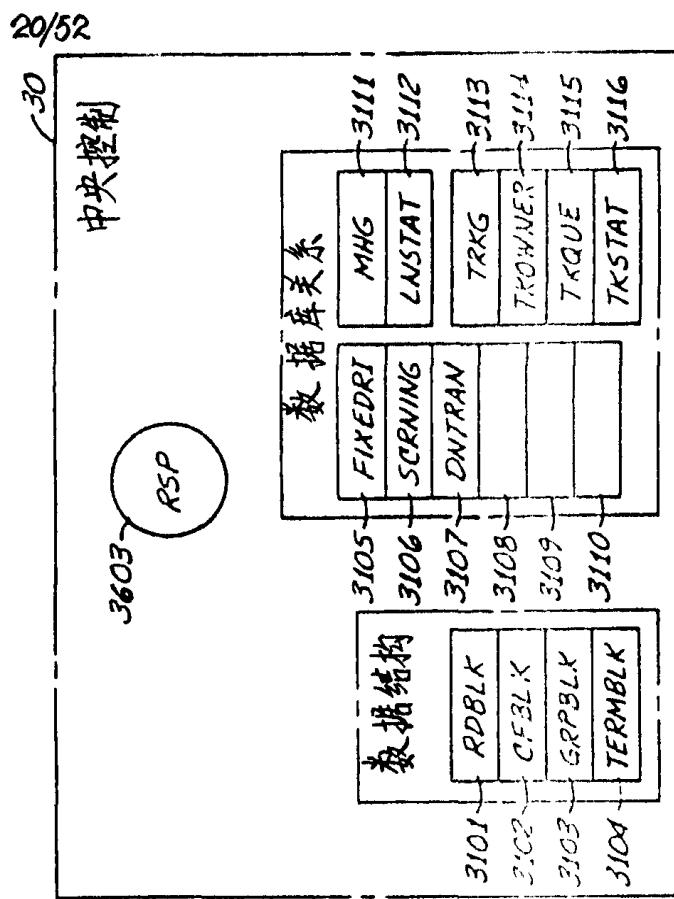
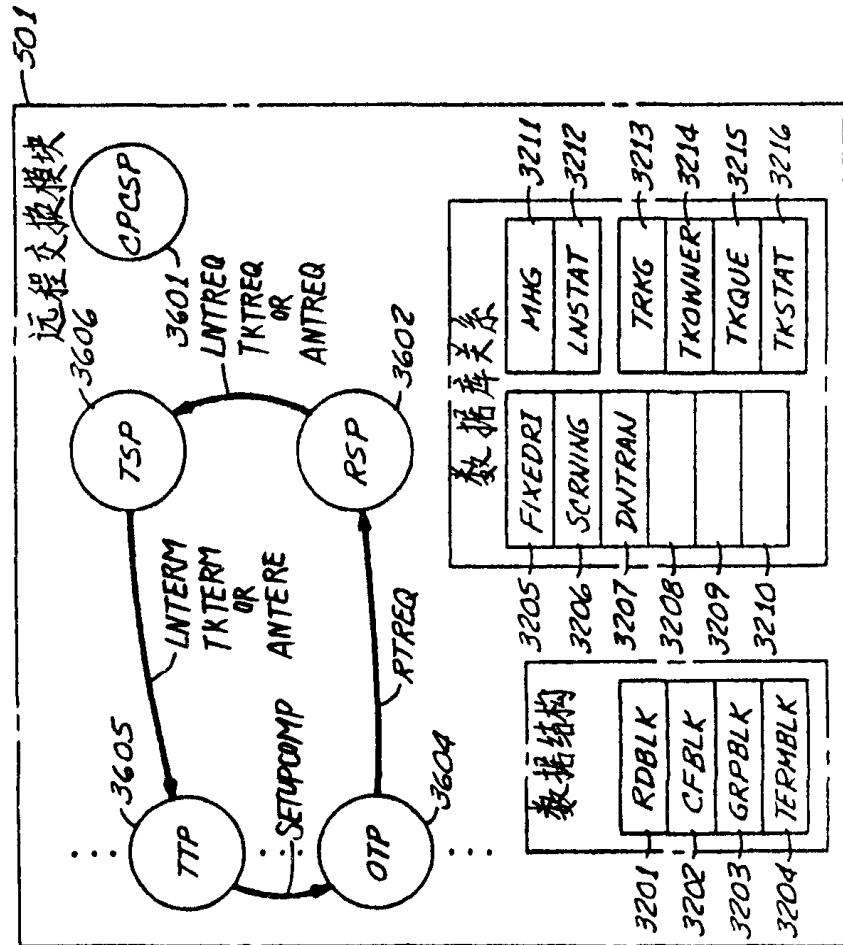


图.24

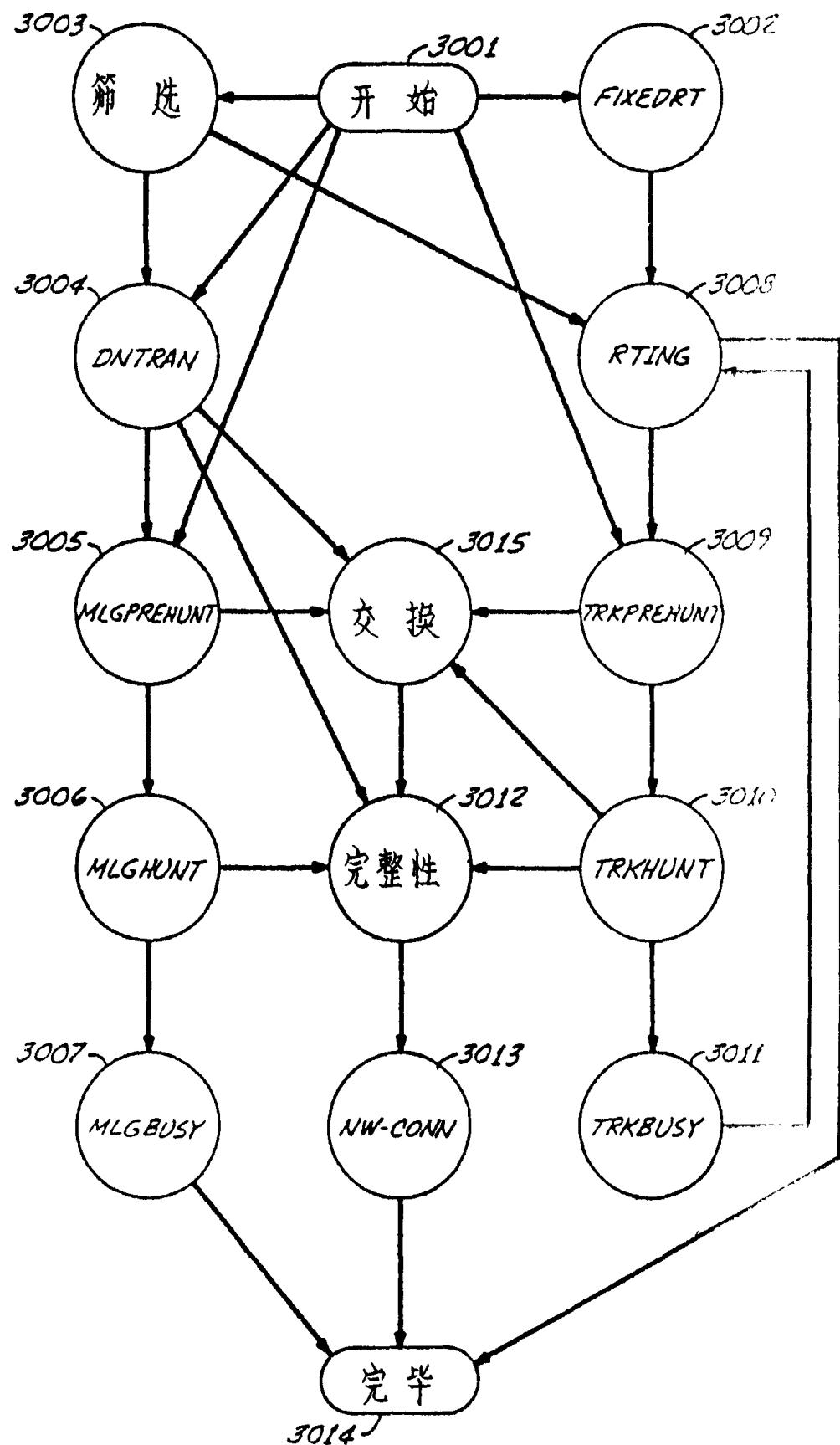
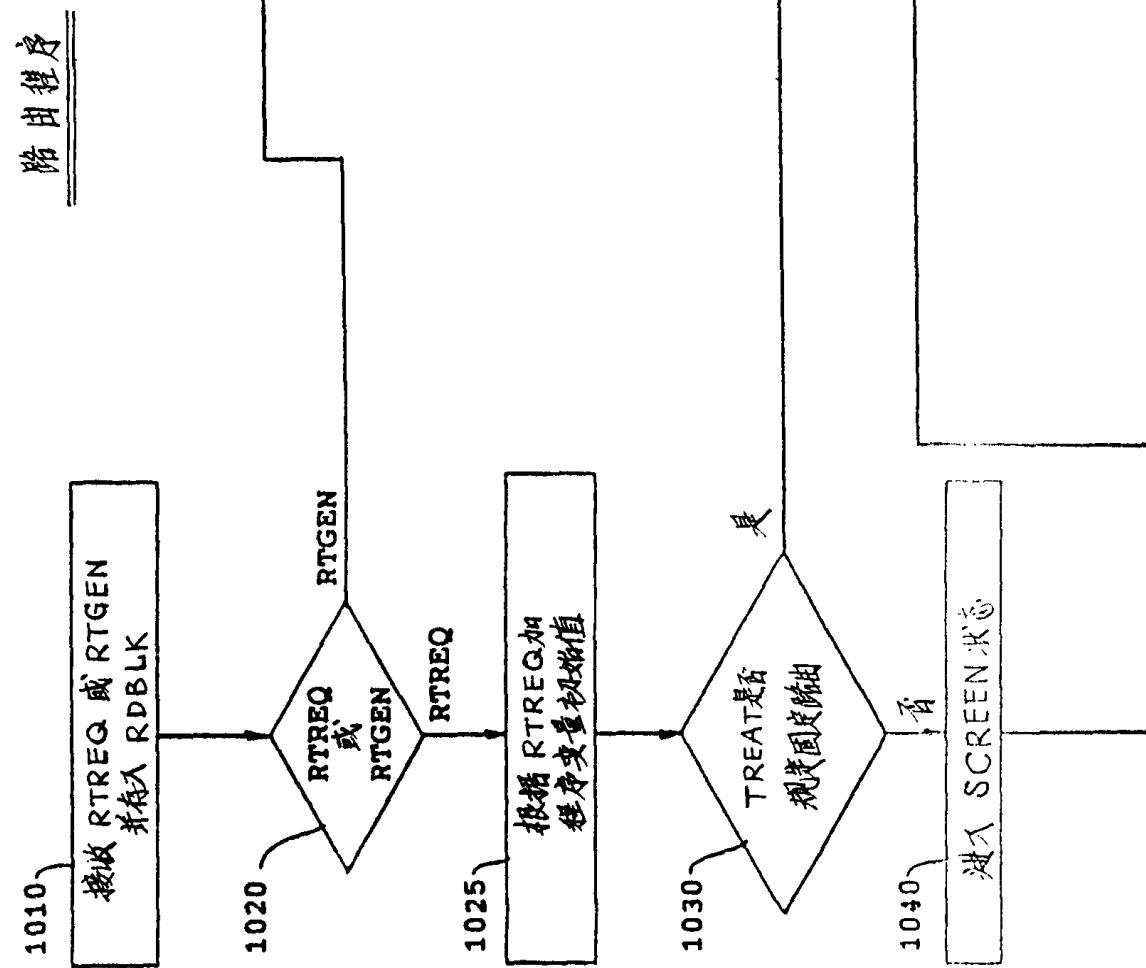


图 25



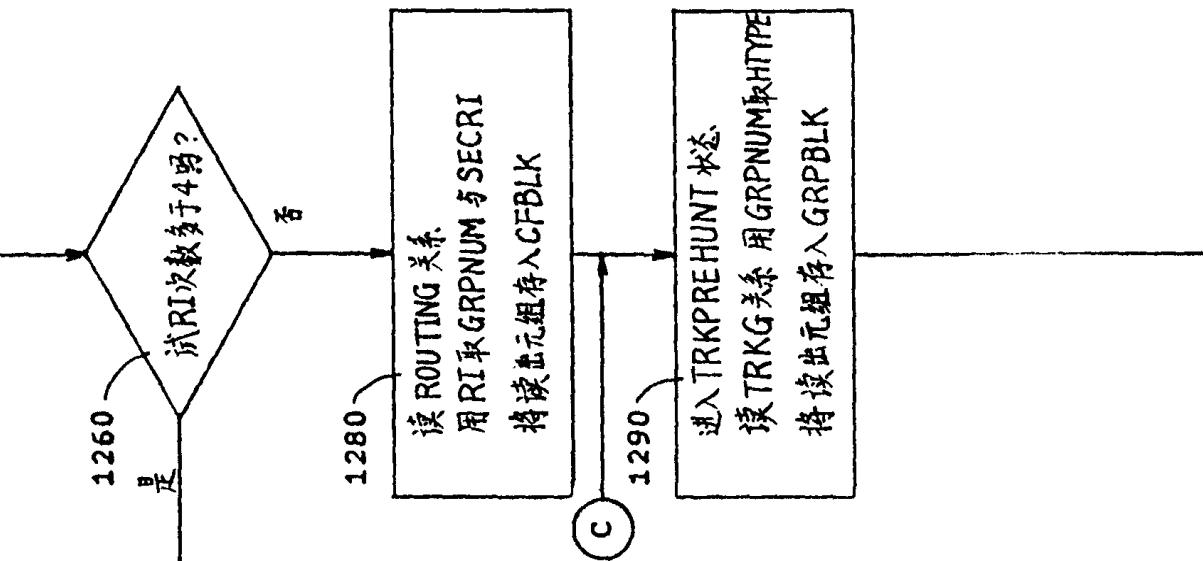
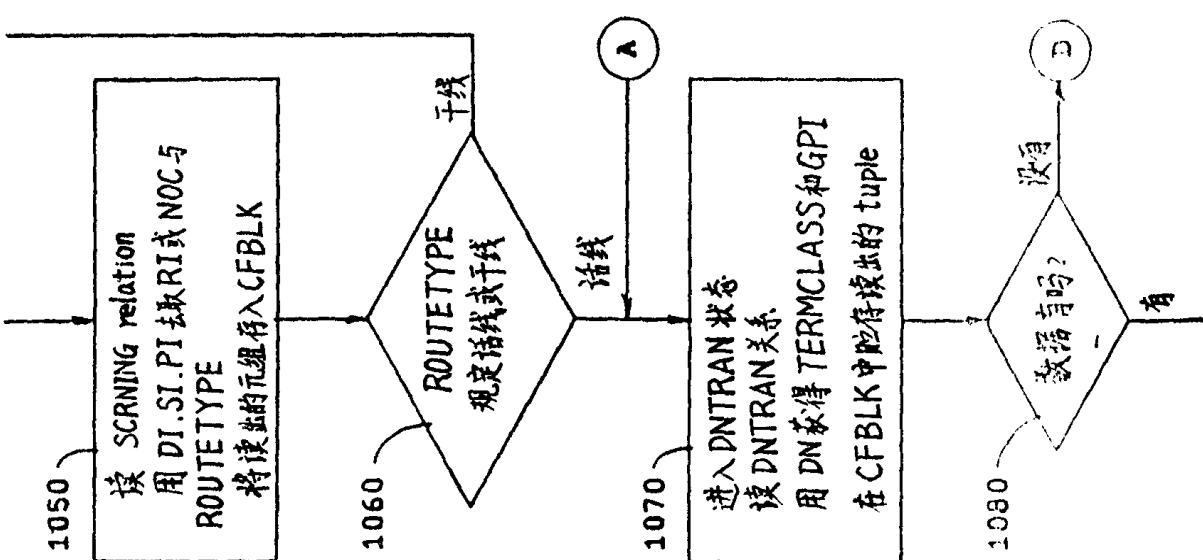
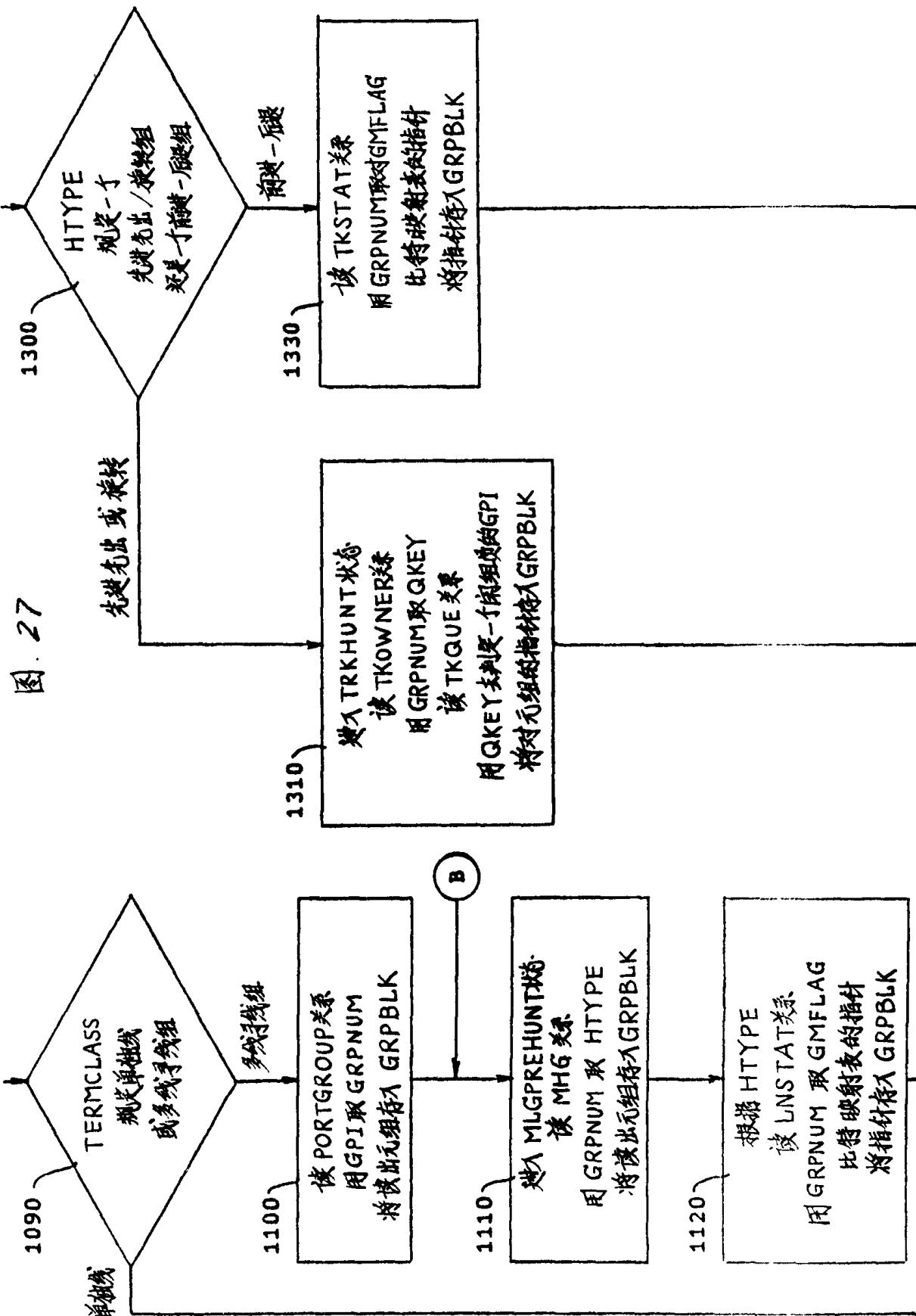


图. 26





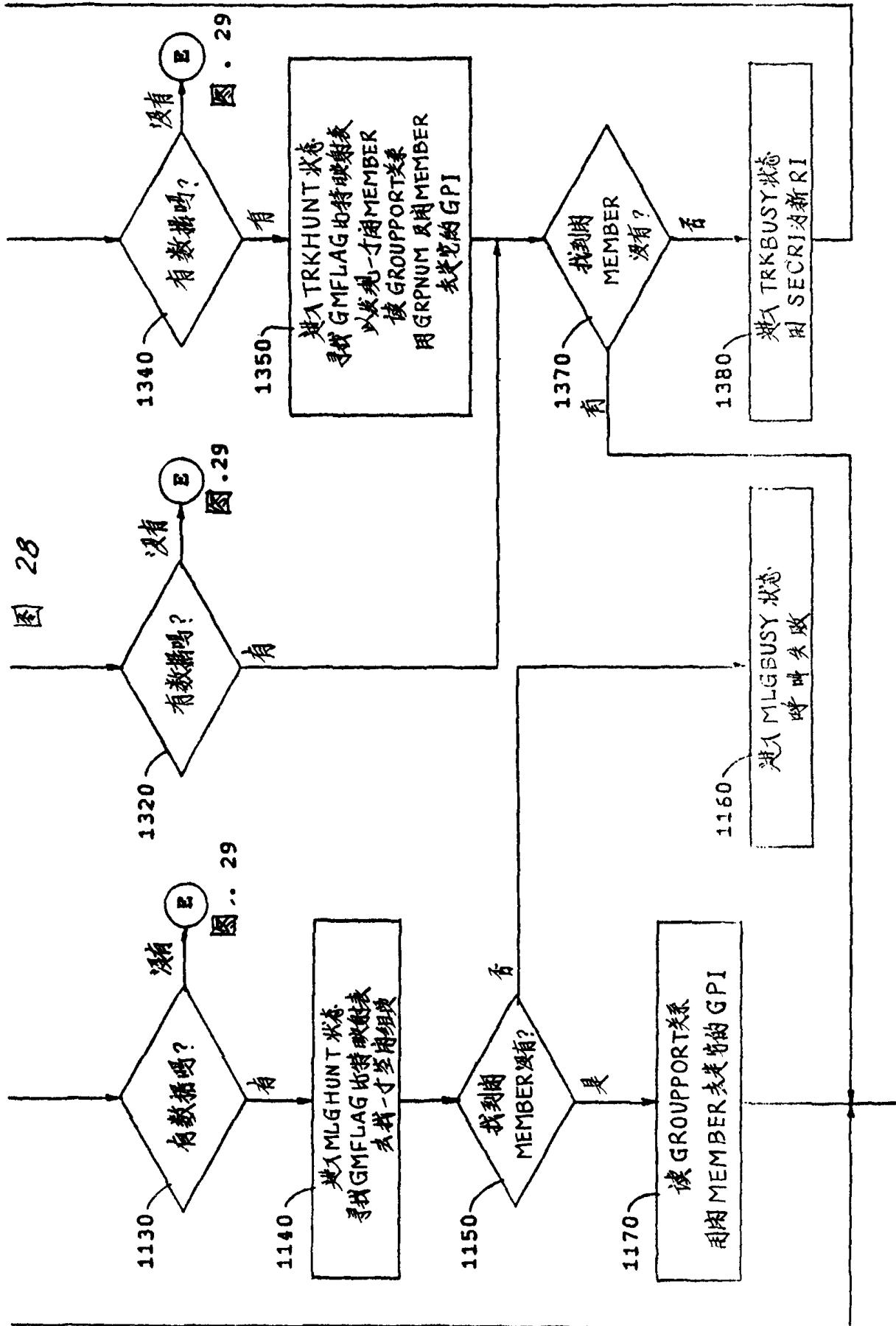
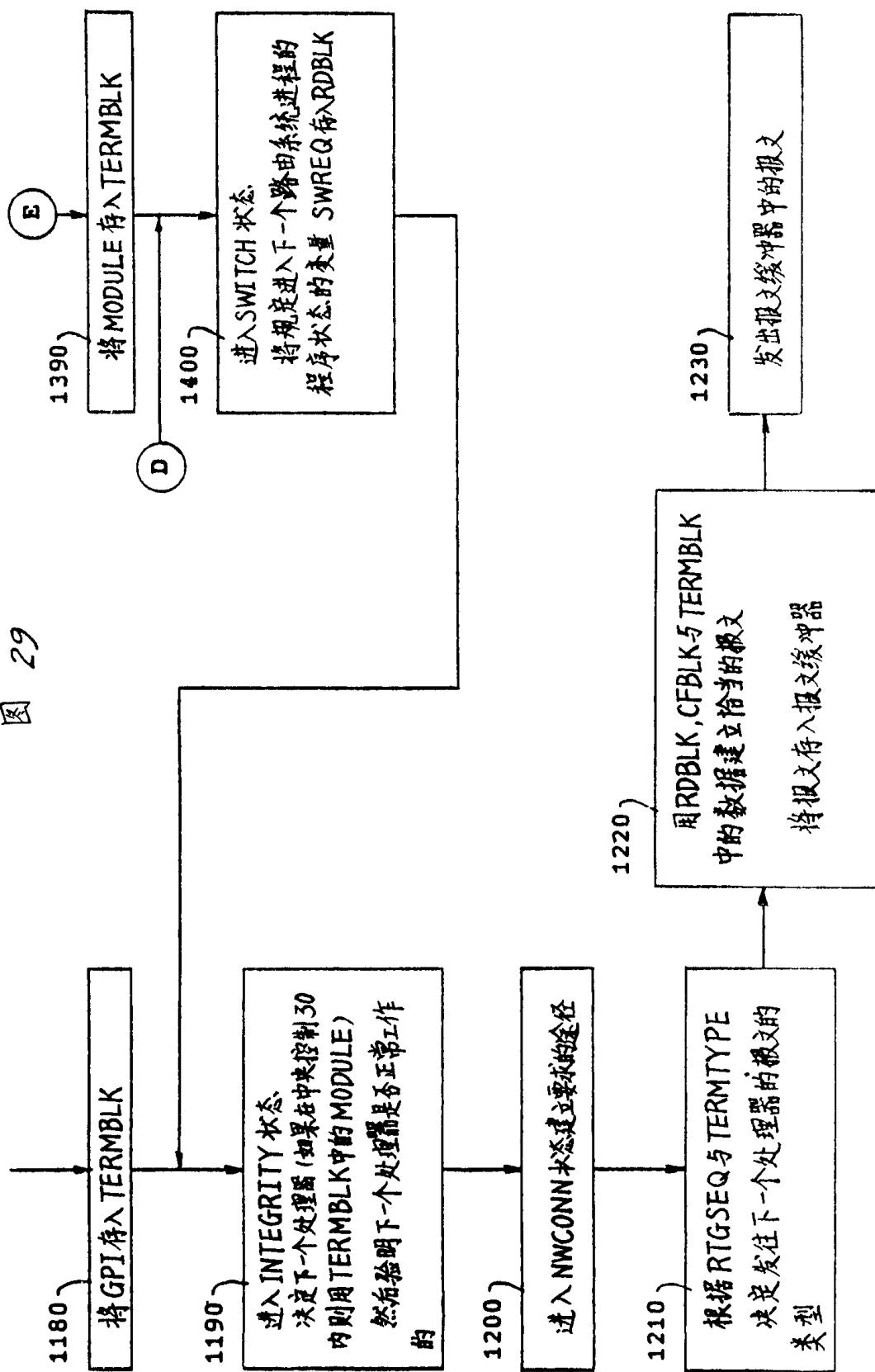
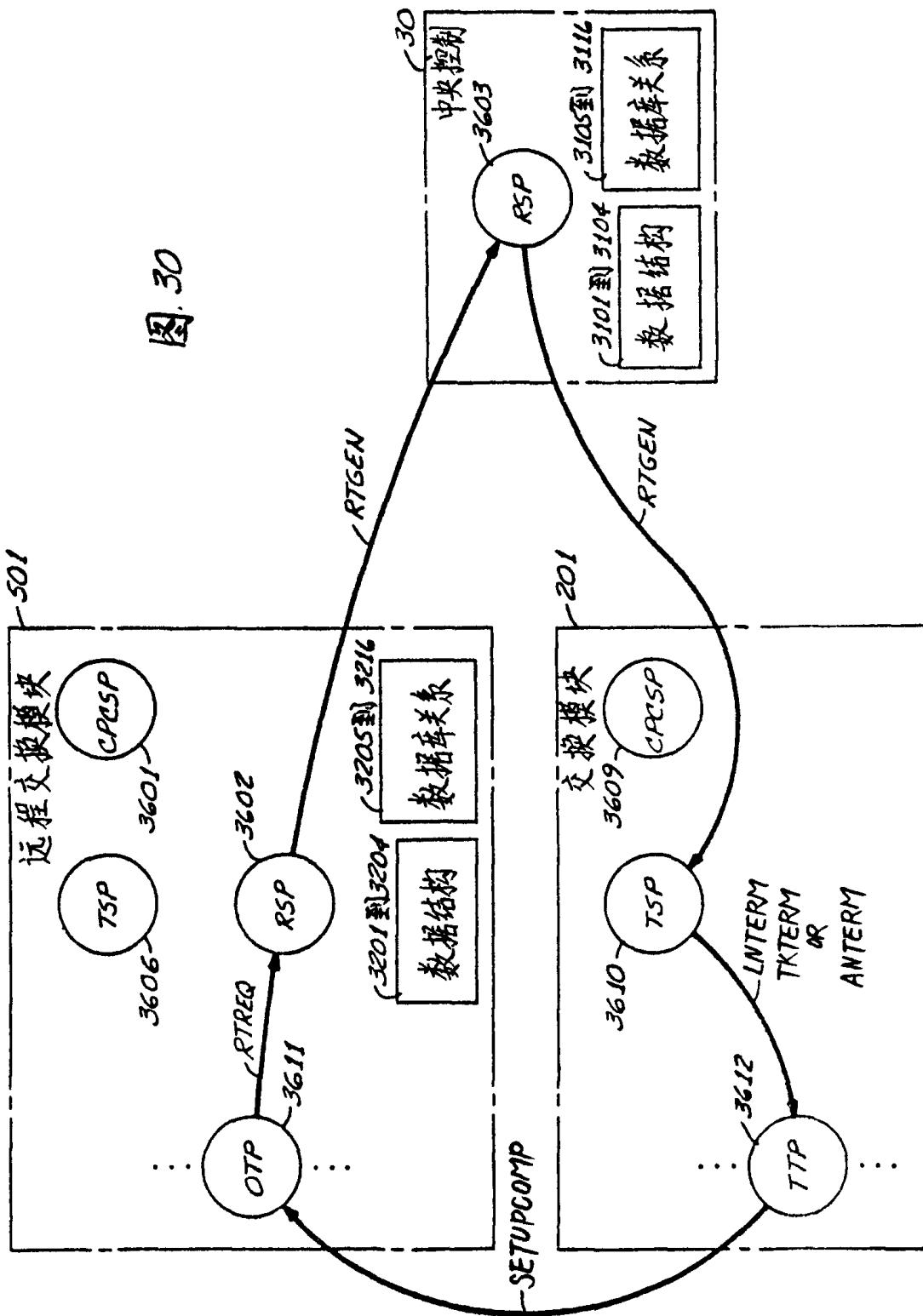
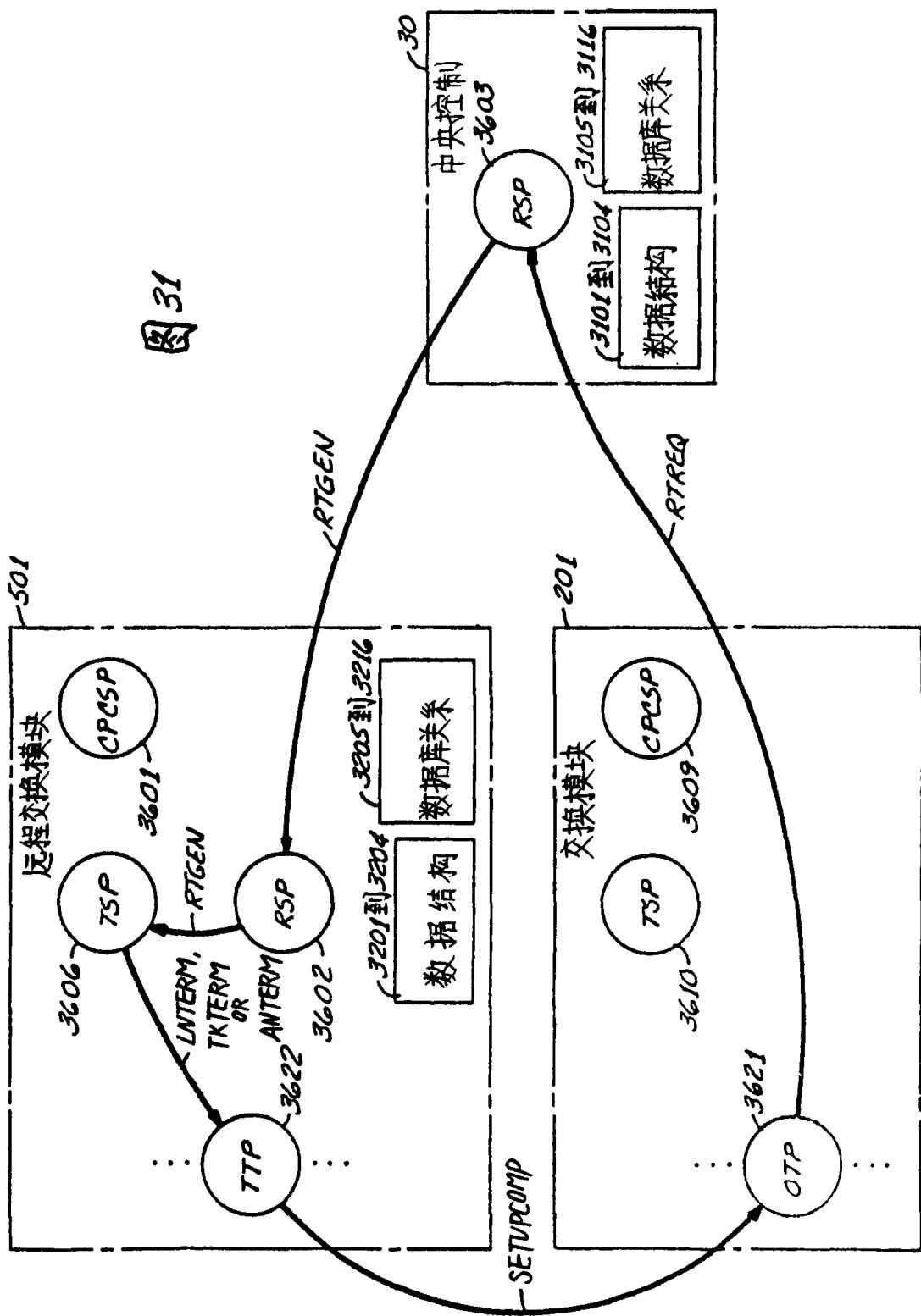


图 29







29/52

RTGEN PATHDES, RTGDATA, FARPID, REQTERM, RTCONTDA, ORIGGPI, TERMGPI, ...

PR PATHDES, RTGDATA, FARPID, ORIGGPI, TERMGPI, ...

RDBLK HEADER, TEXT, ORIGGPI, RTGSTATE, RICOUNT, RTSEQ, SWREQ, ...

MHG GRPNUM, HTYPE, MODULE, ...

TRKG GRPNUM, HTYPE, MODULE, ...

PORTSTATUS GPI, BUSY/IDLE, CF, SC, ...

MODTRAN NOCD4D3, MODULE, ...

图 . 32

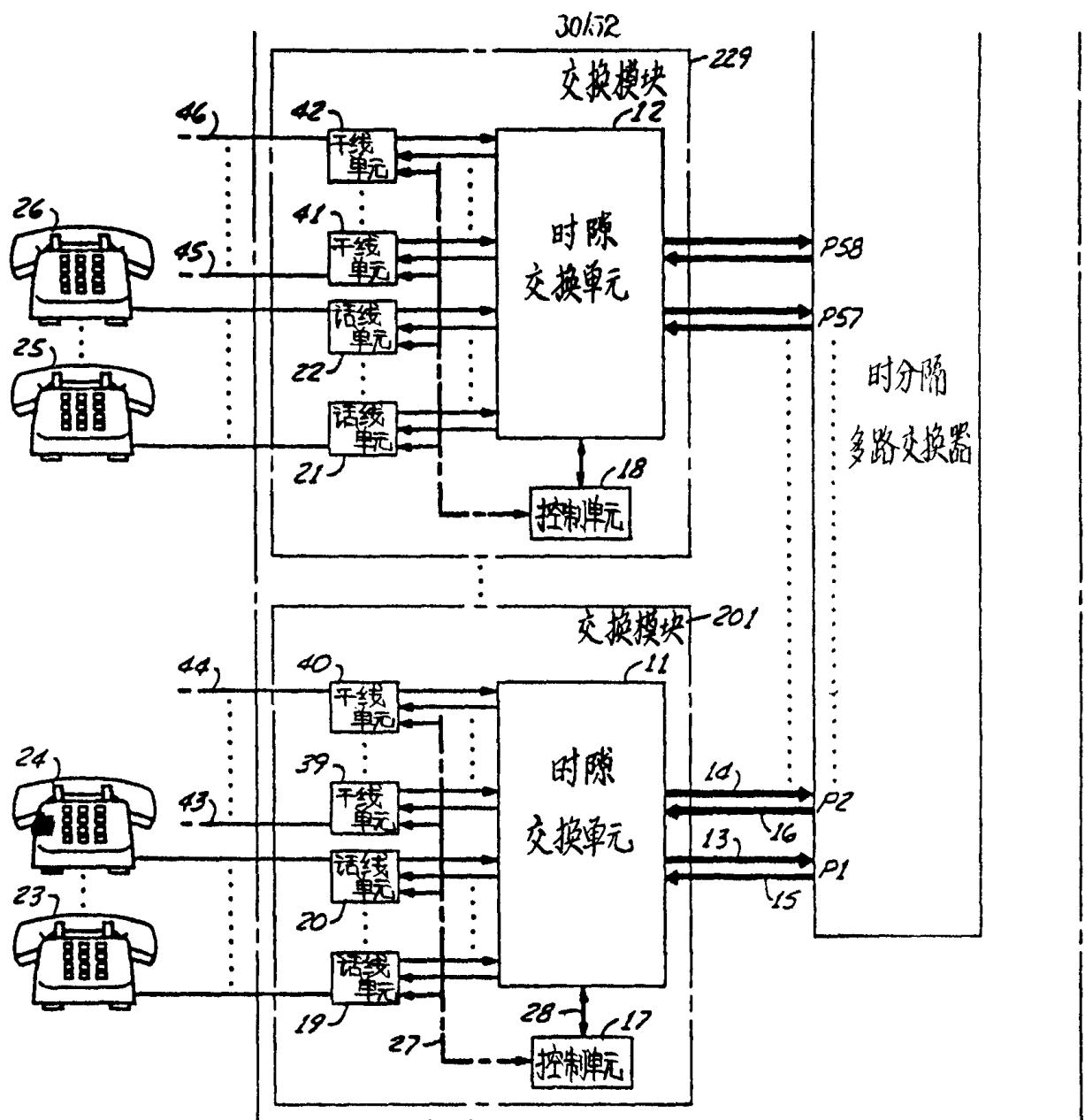
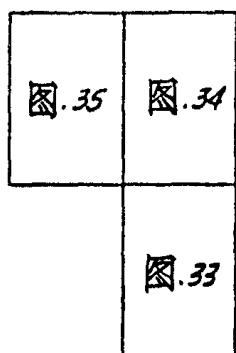


图 33



系统Ⅱ

图 36

图 34

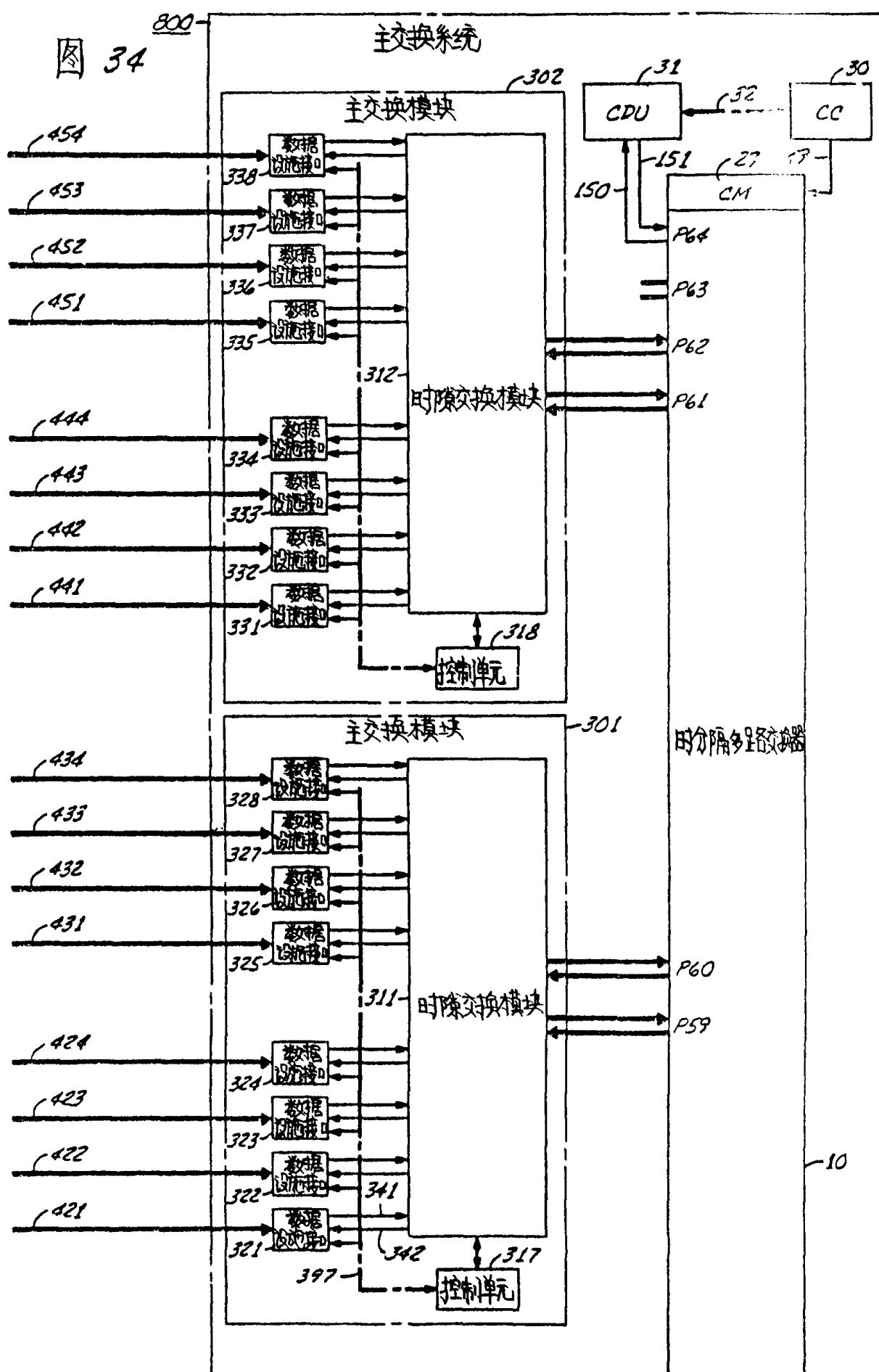
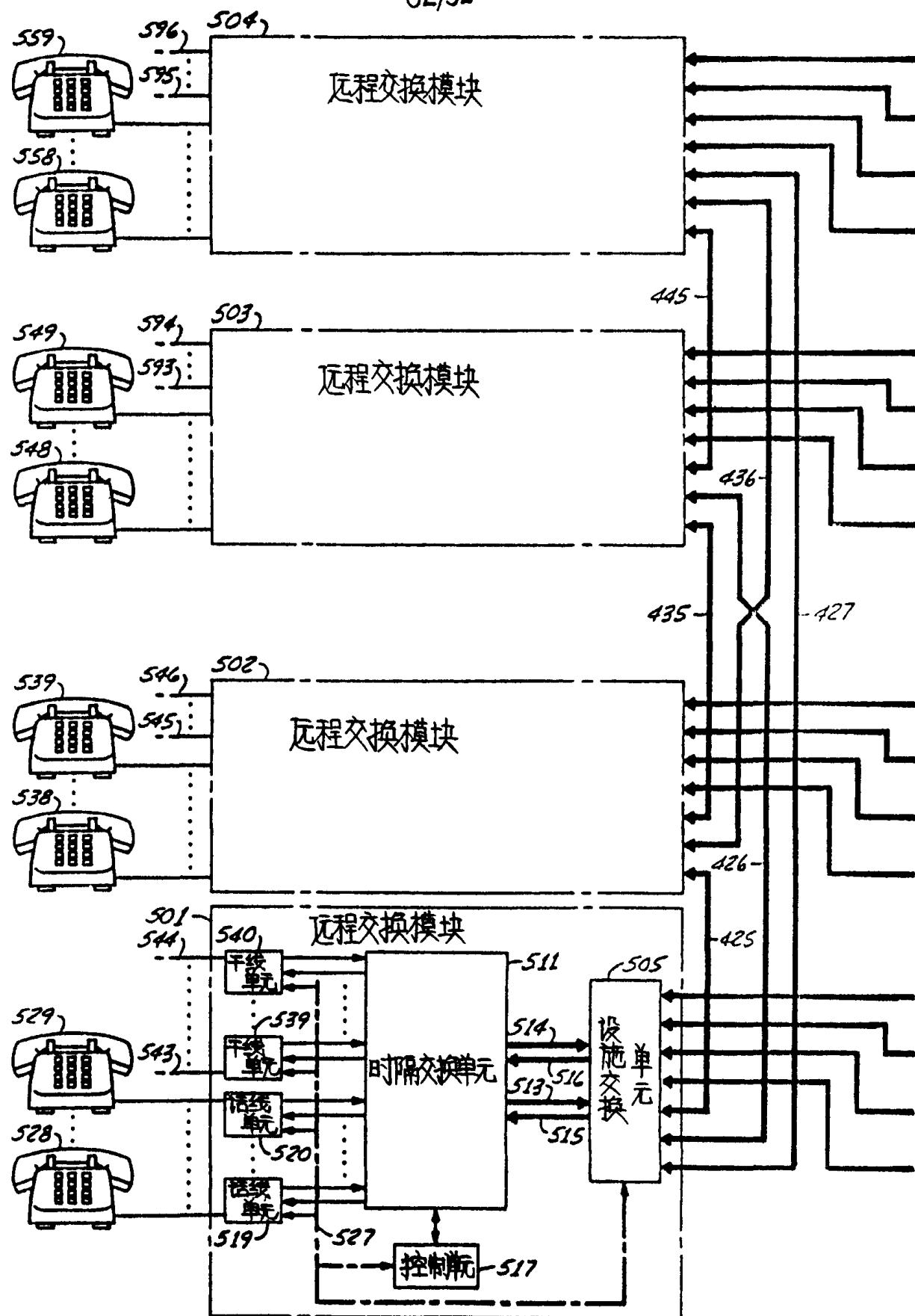


图. 35

32/52



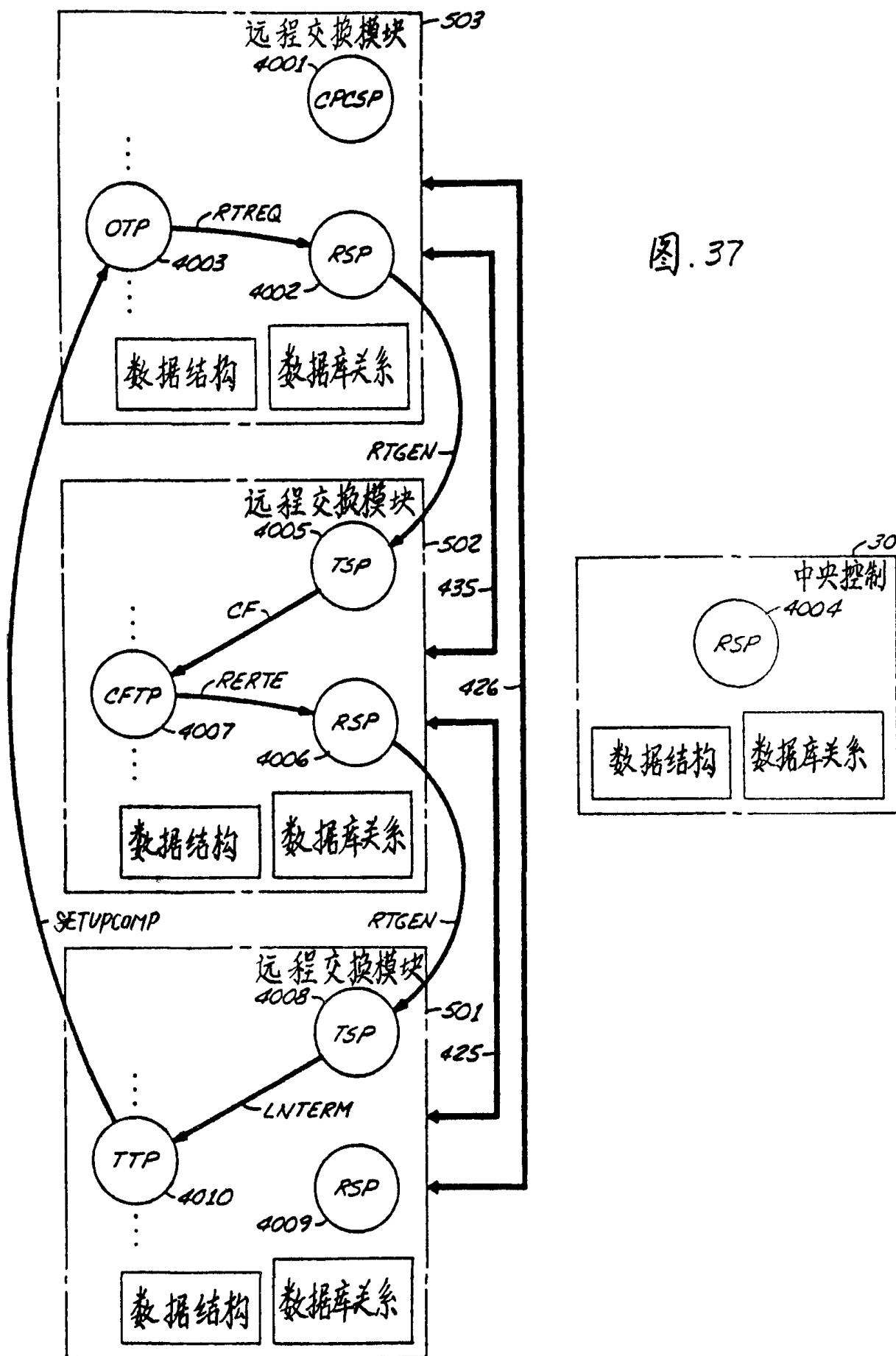
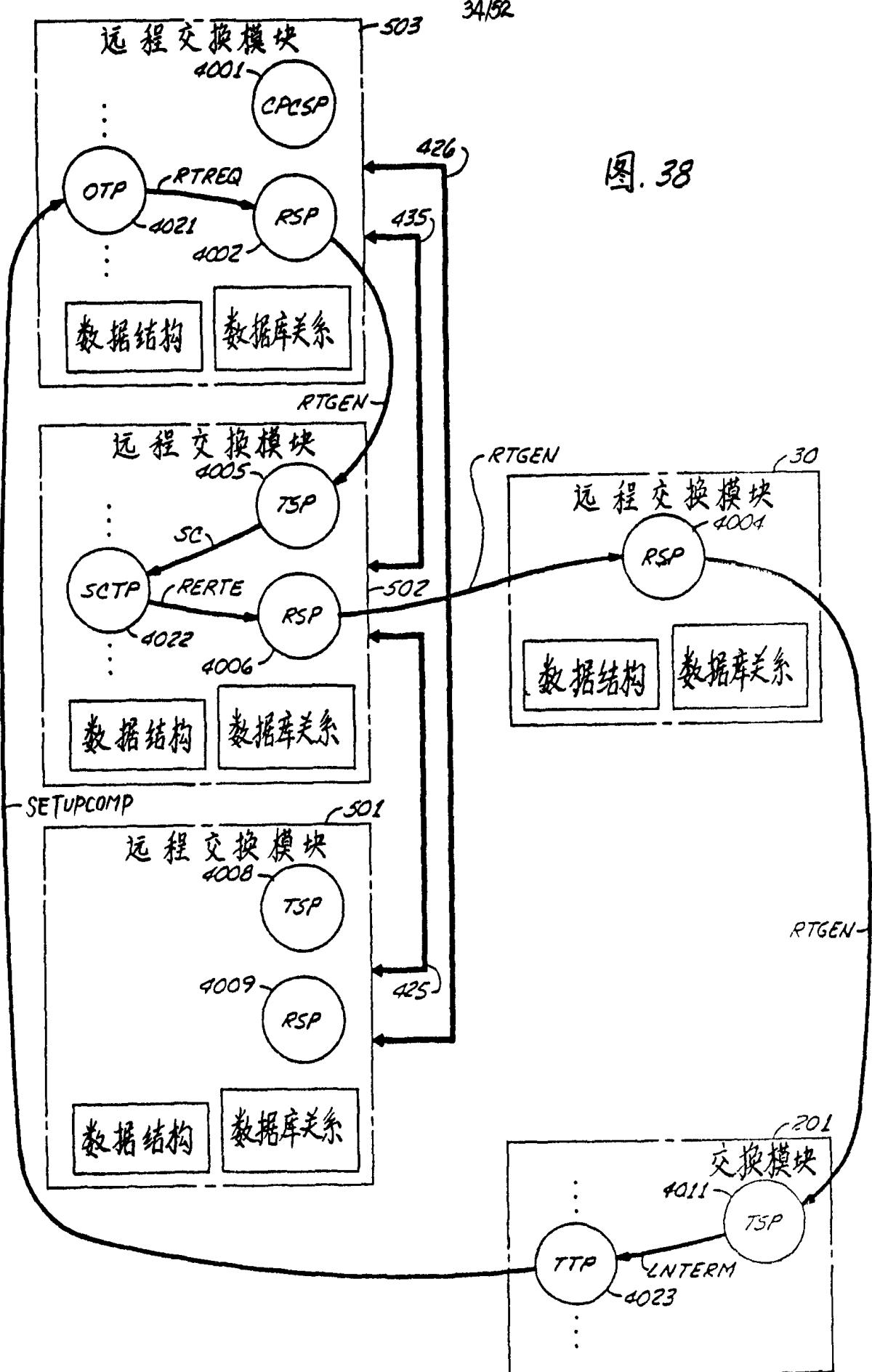


图. 37



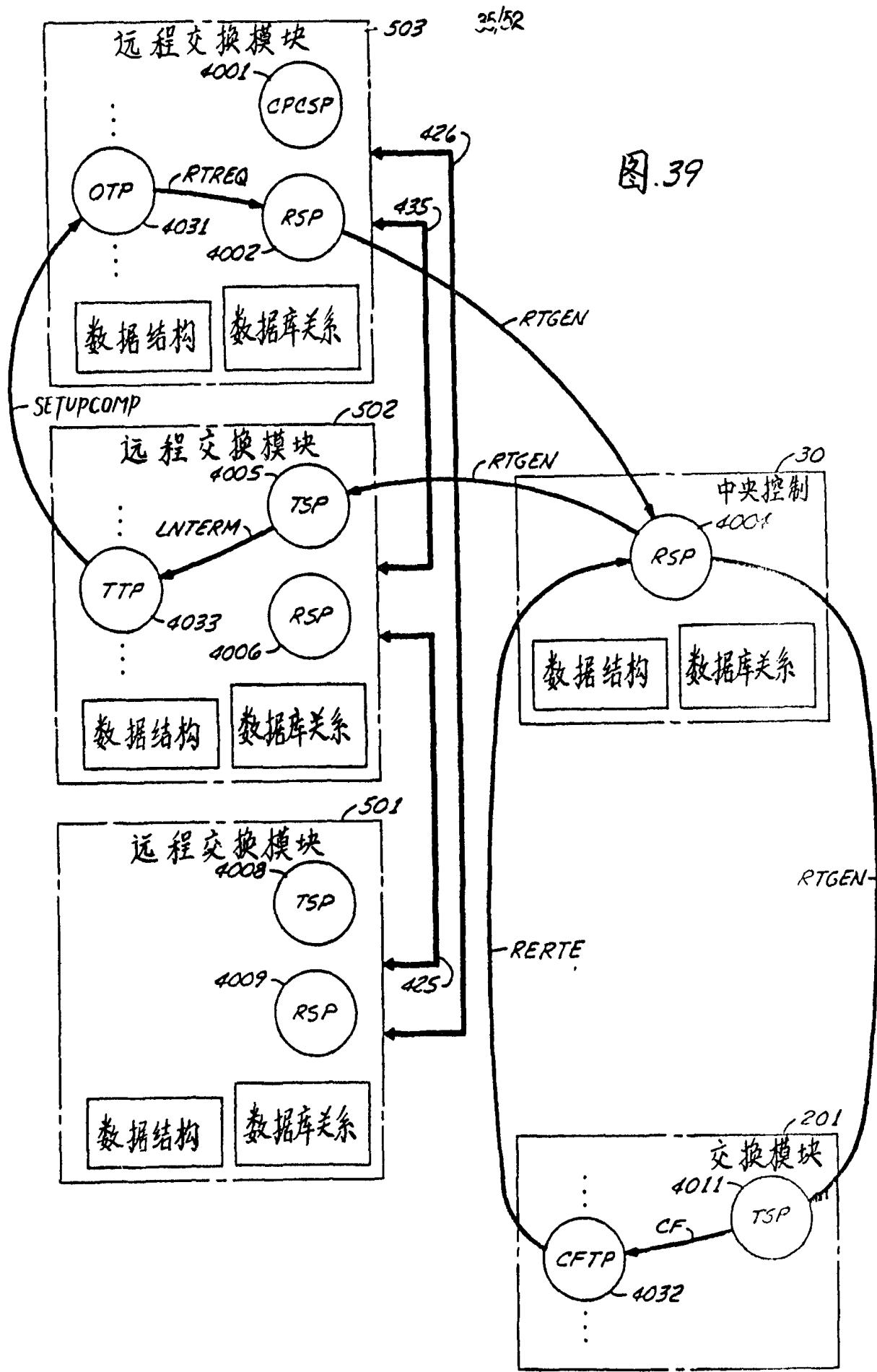


图. 40

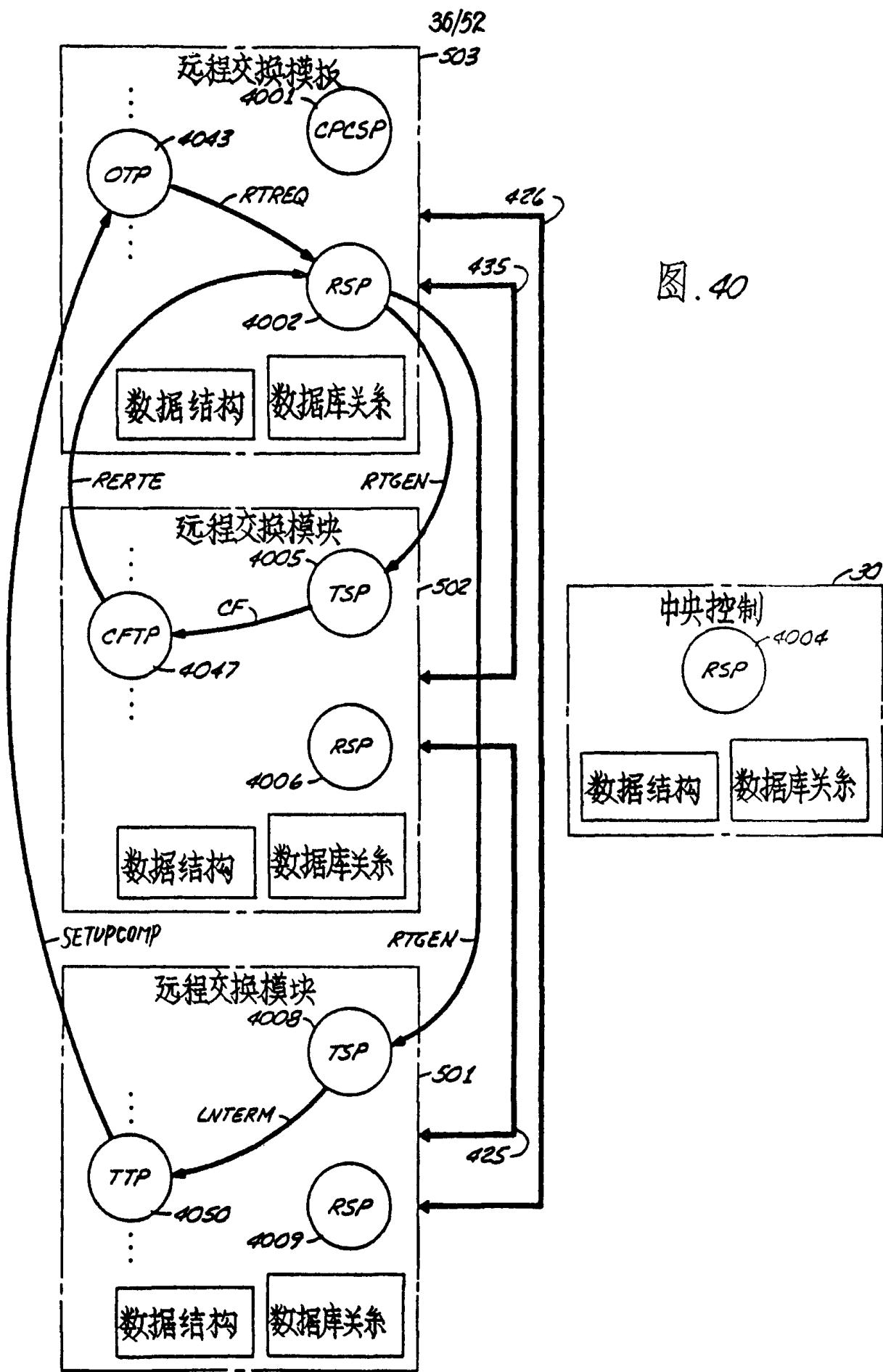
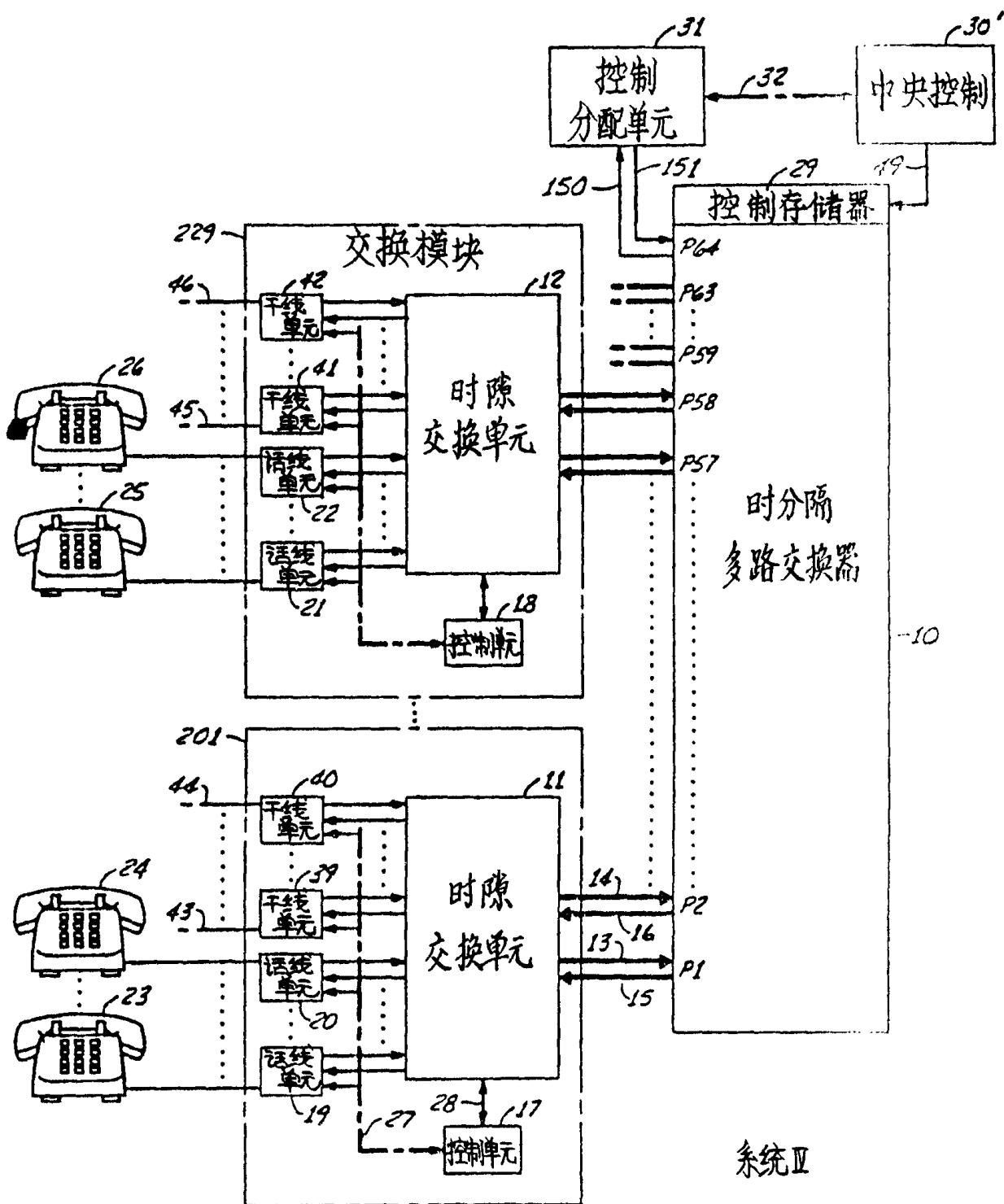
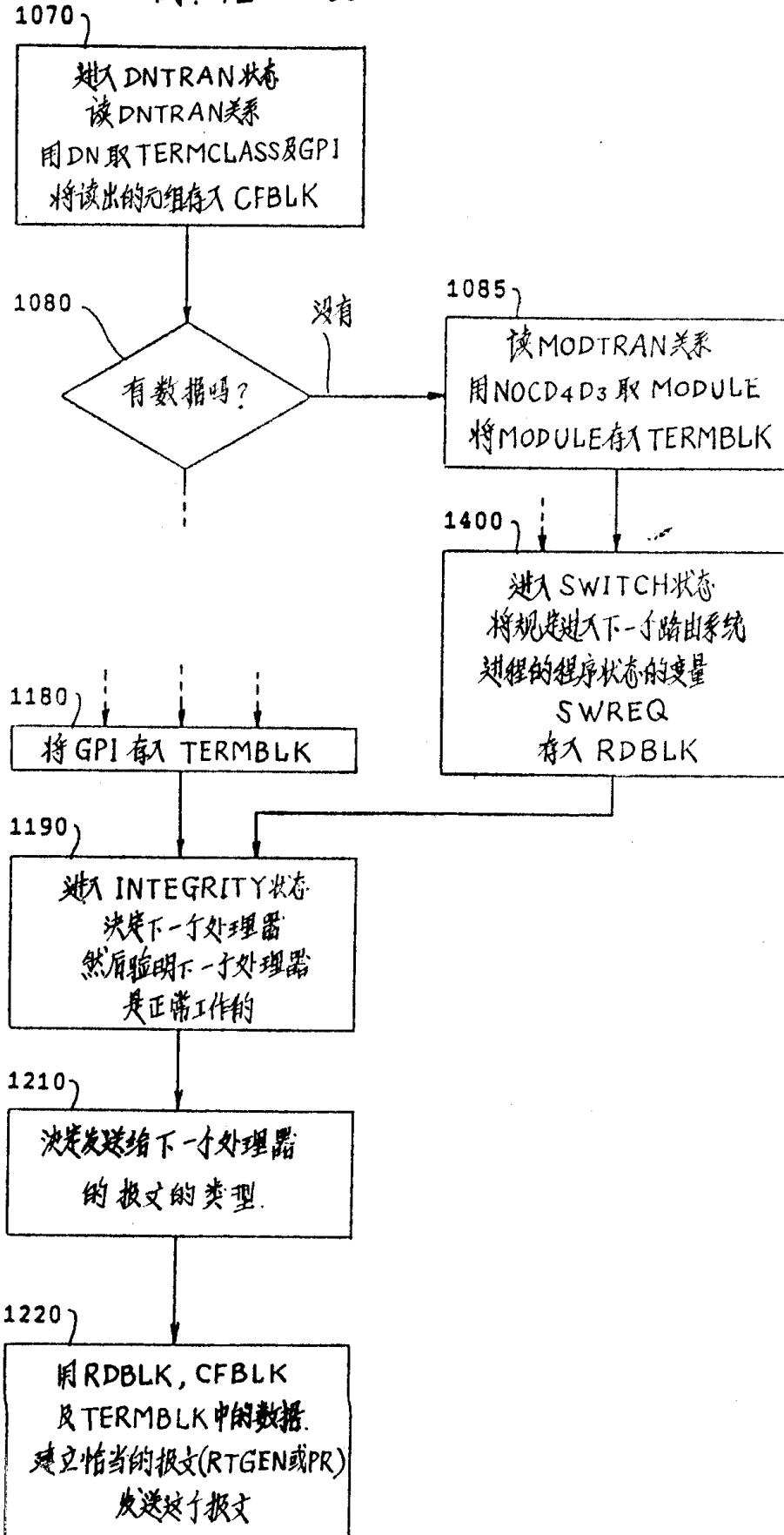


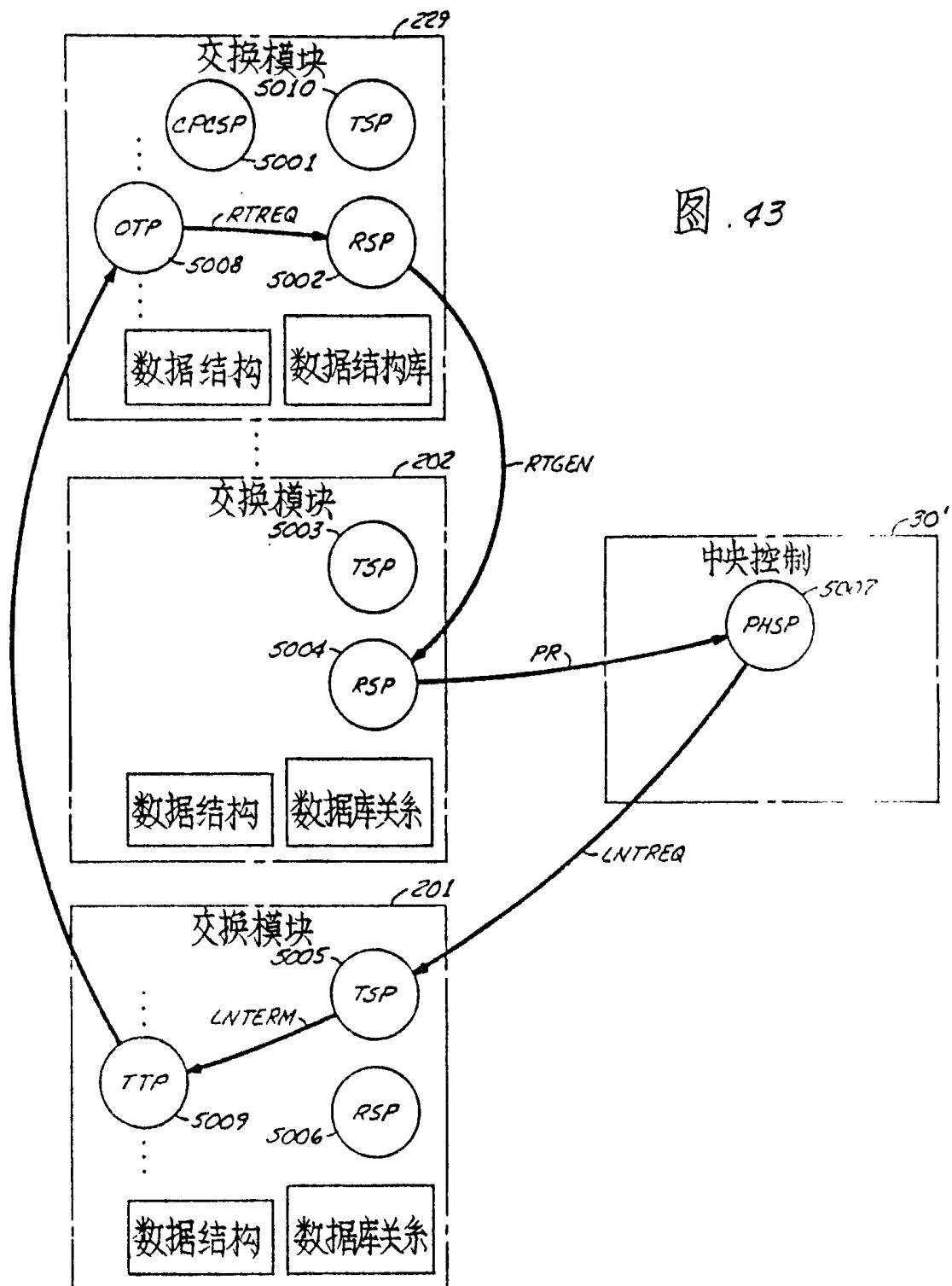
图. 41



系统II

图. 42 38/52





40/32

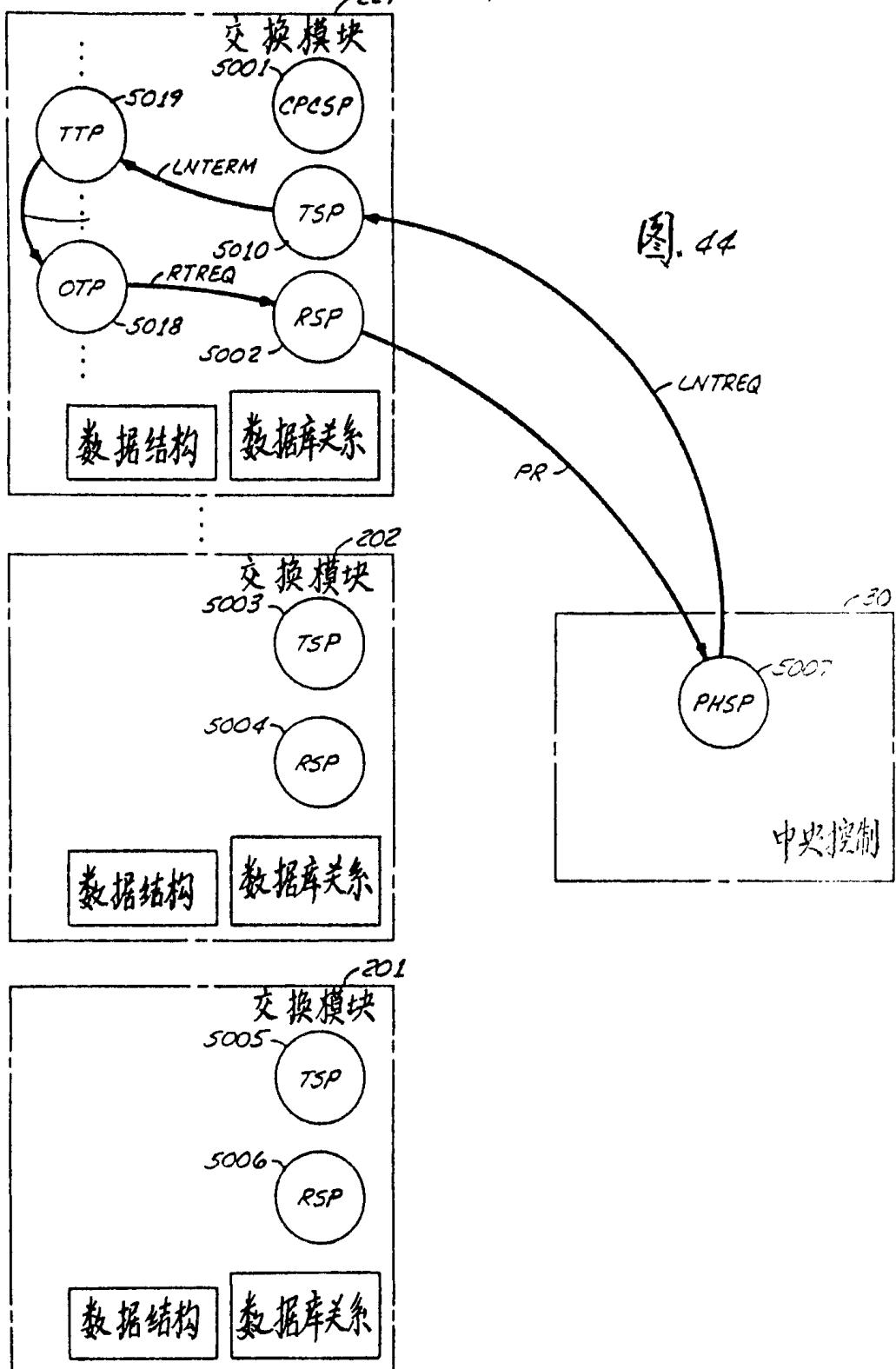
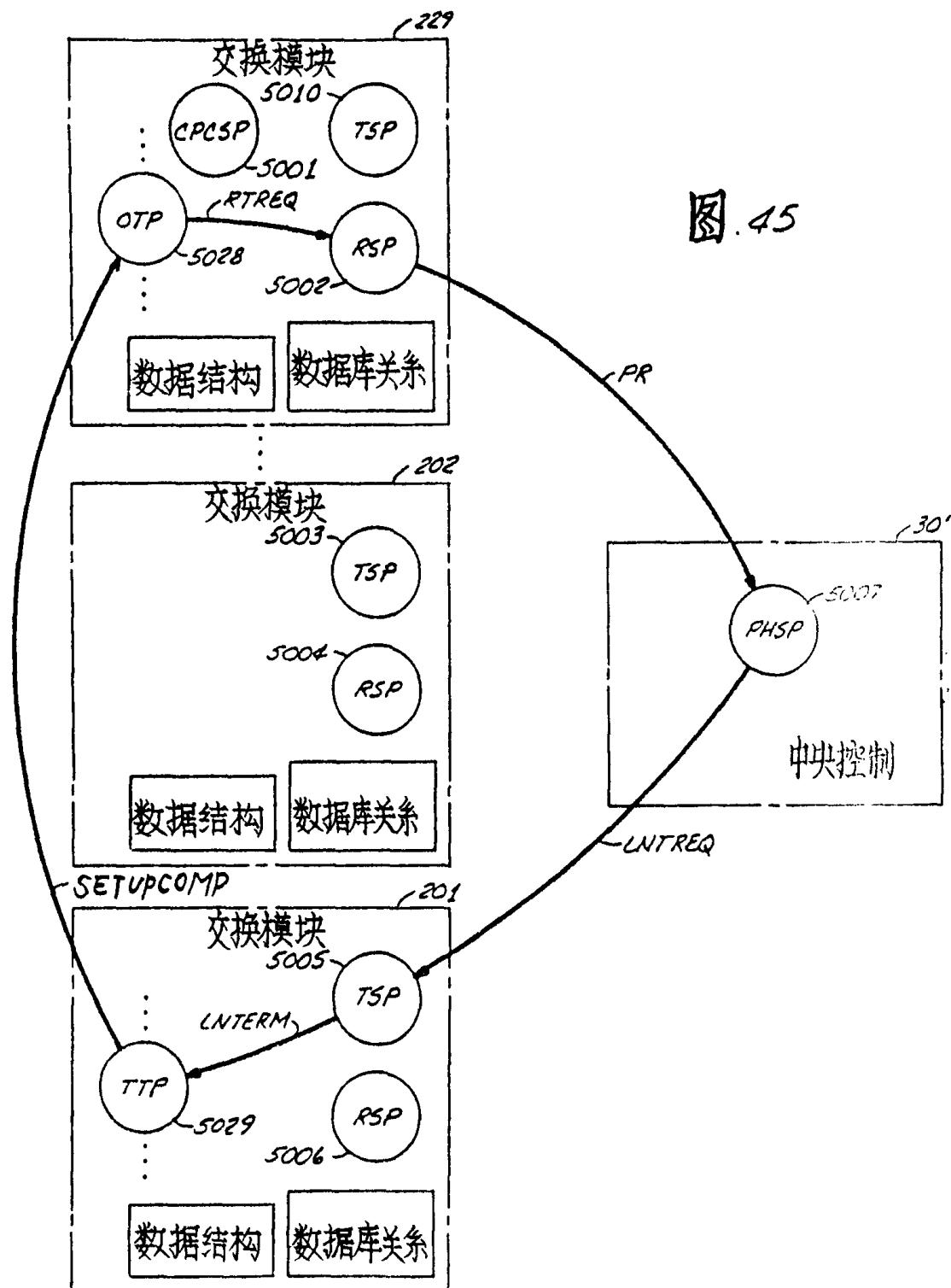
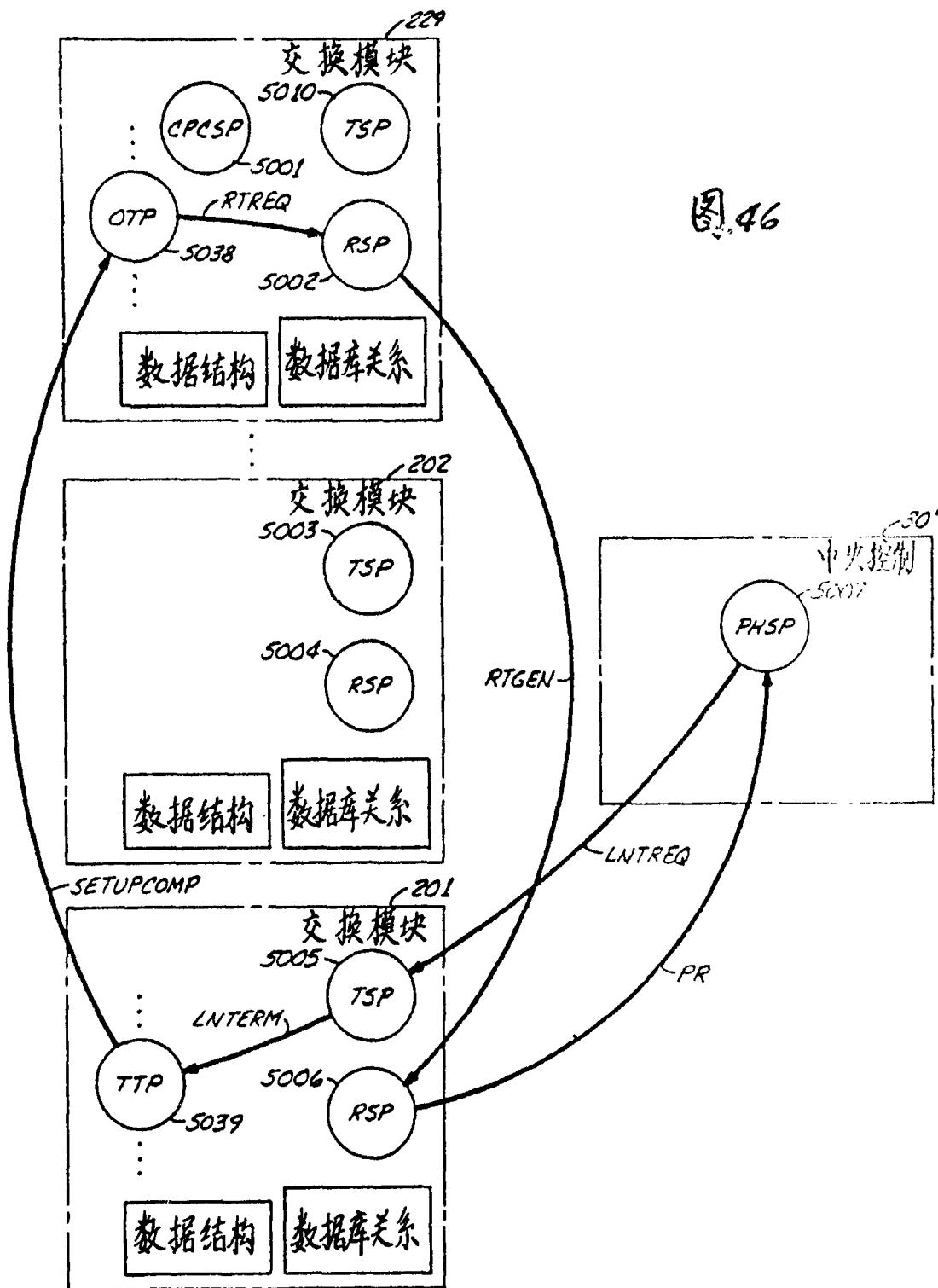


图.45





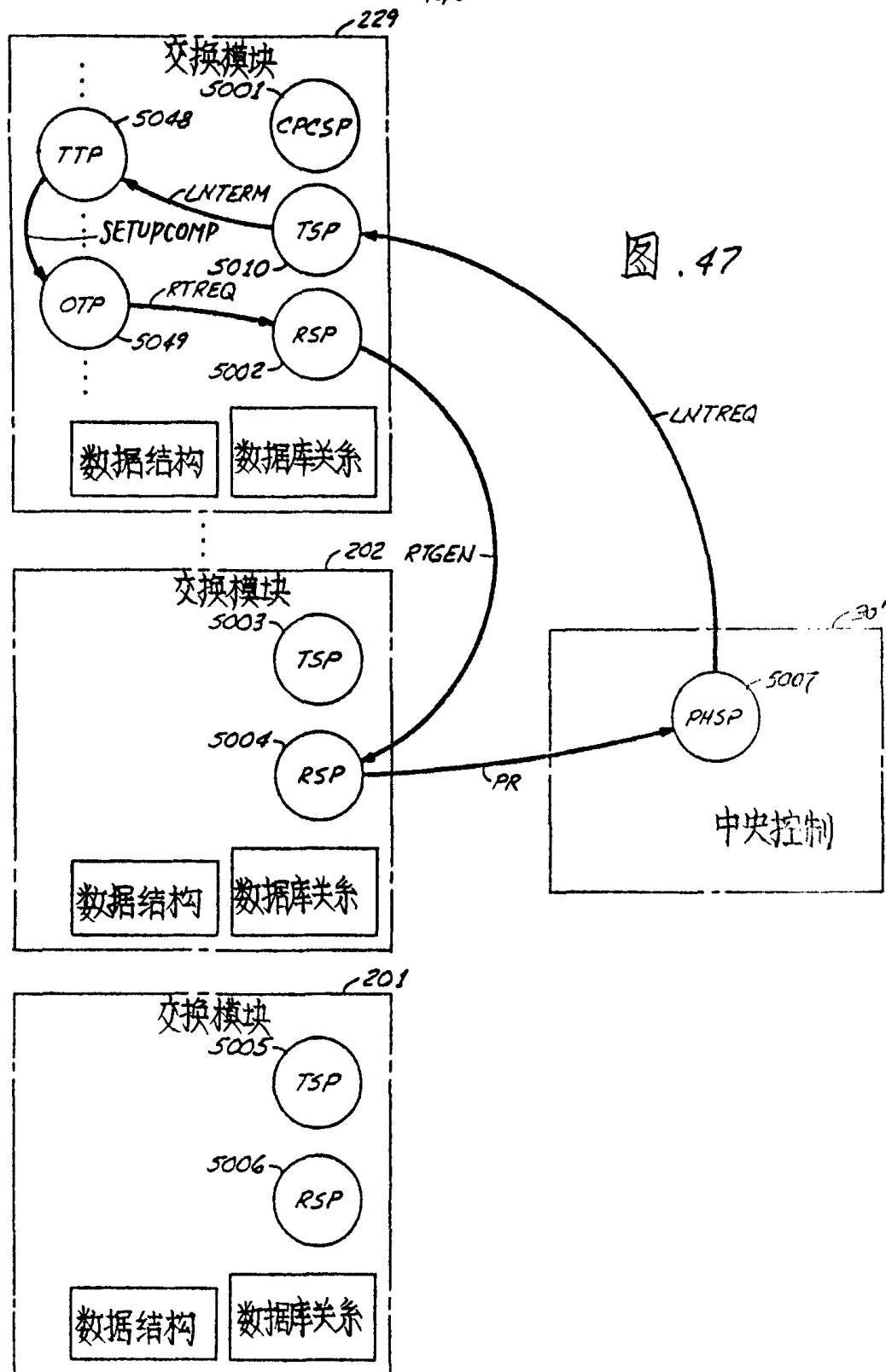
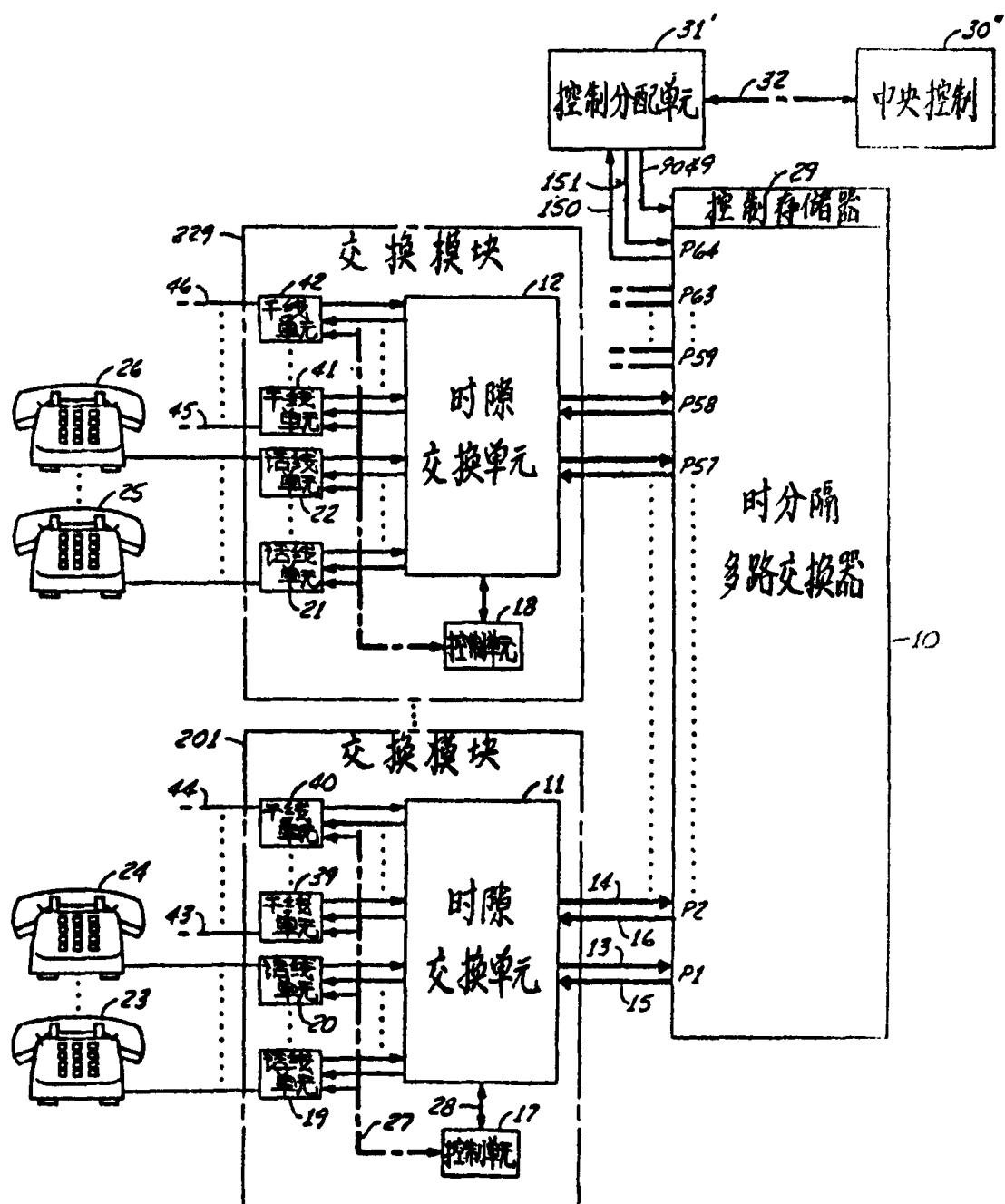


图.48



45/52

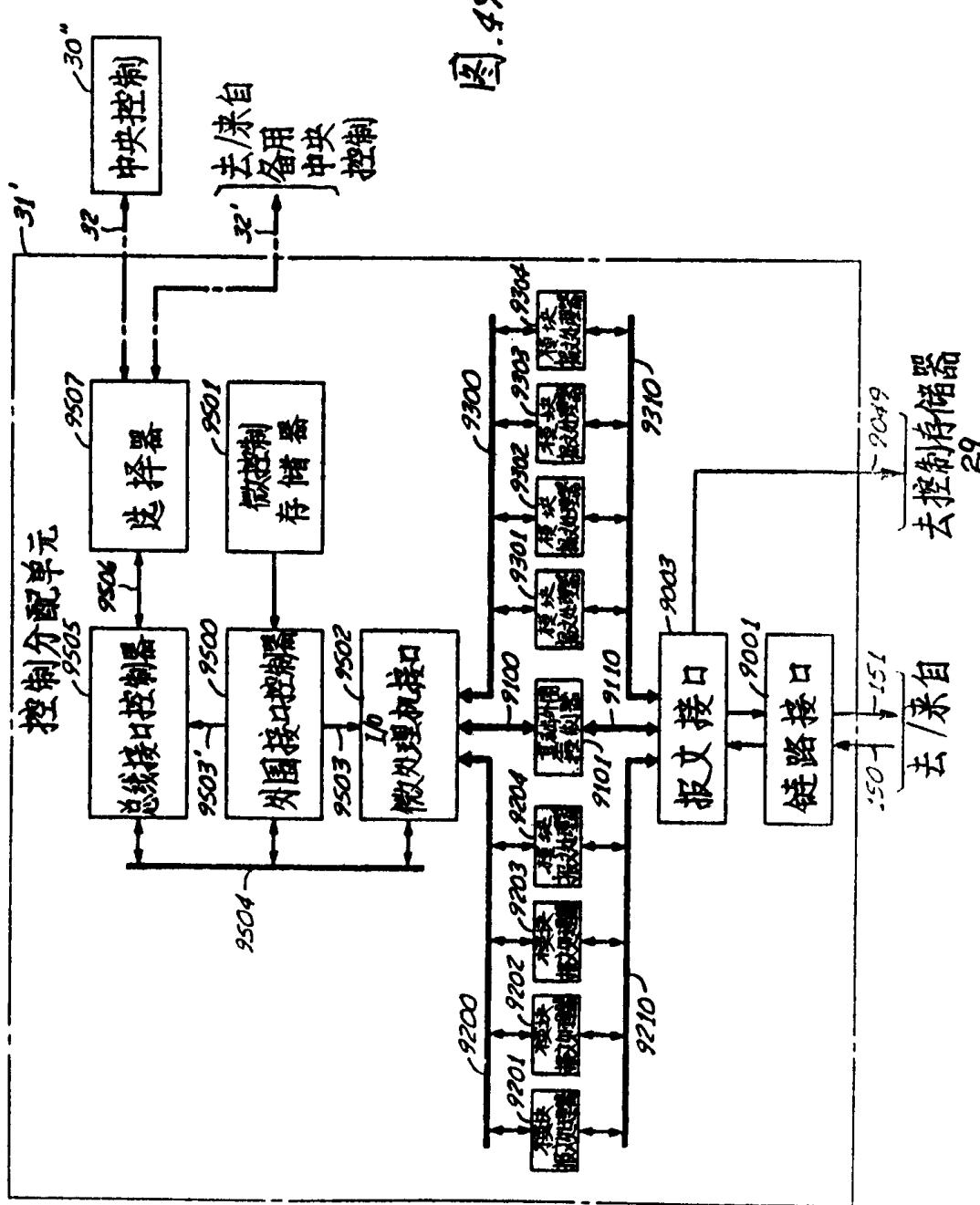
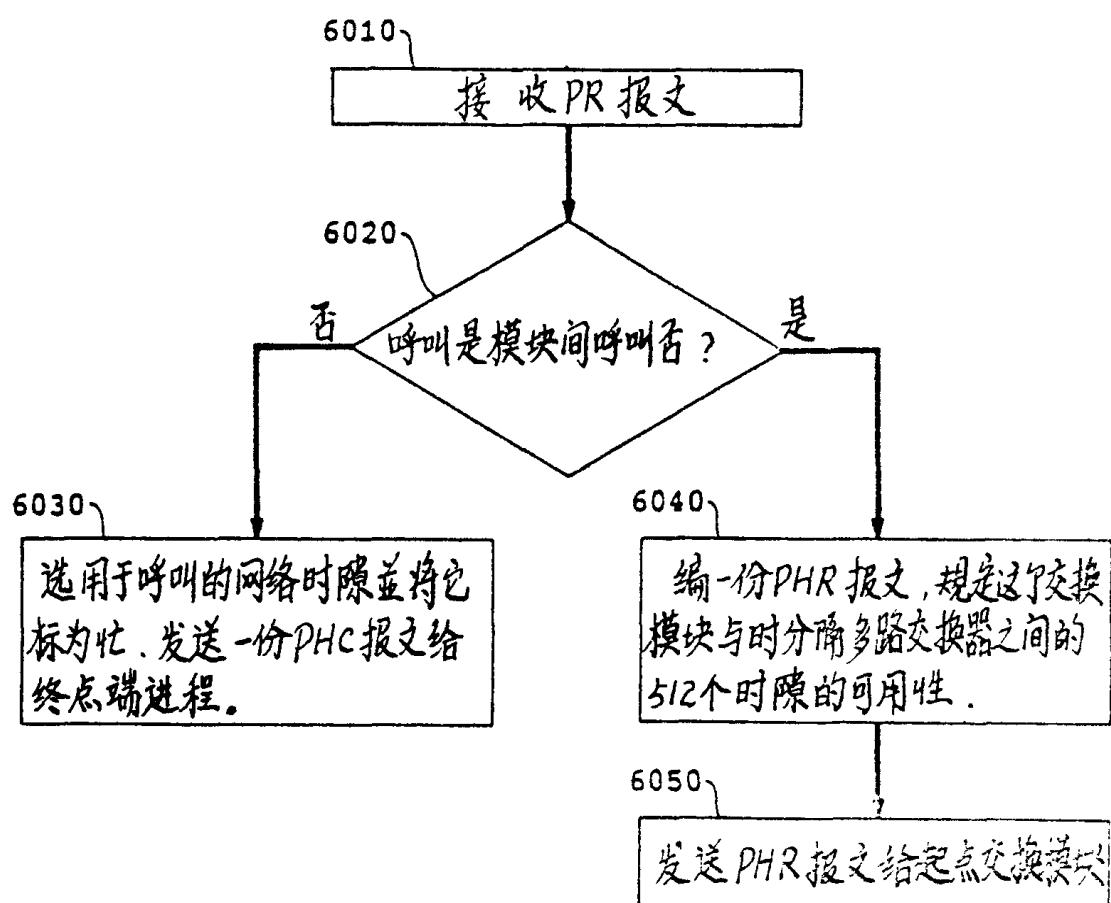
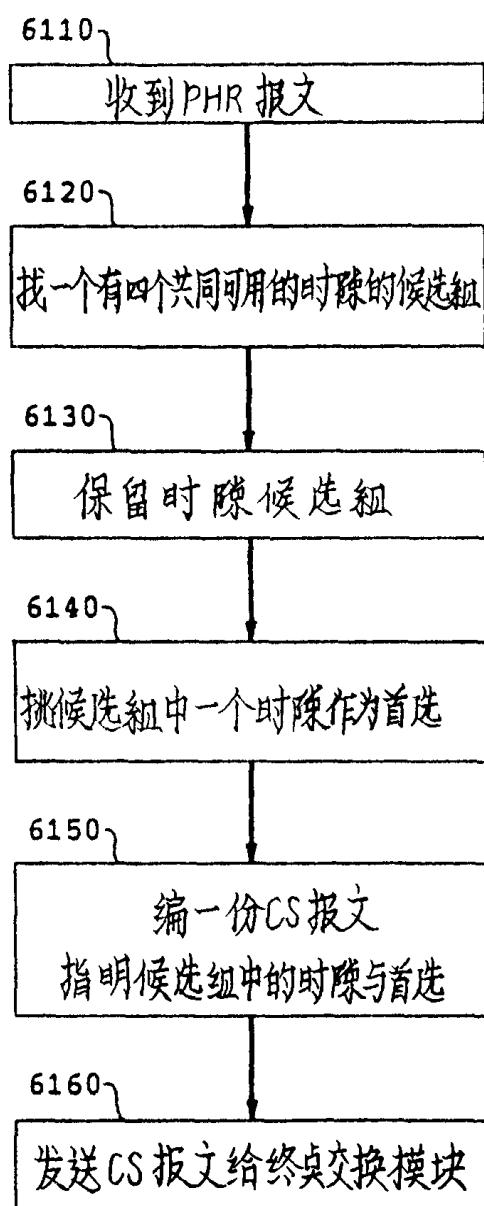


图.50



47152

图.51



48152

图.55



图.52

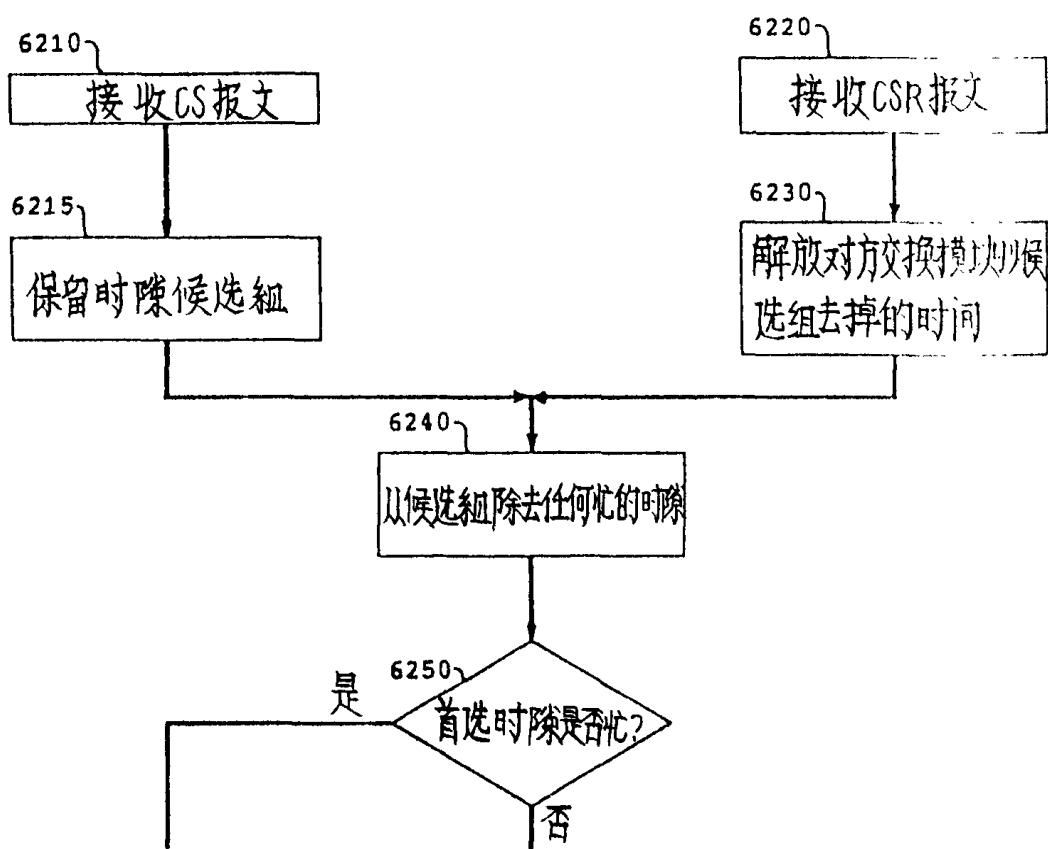
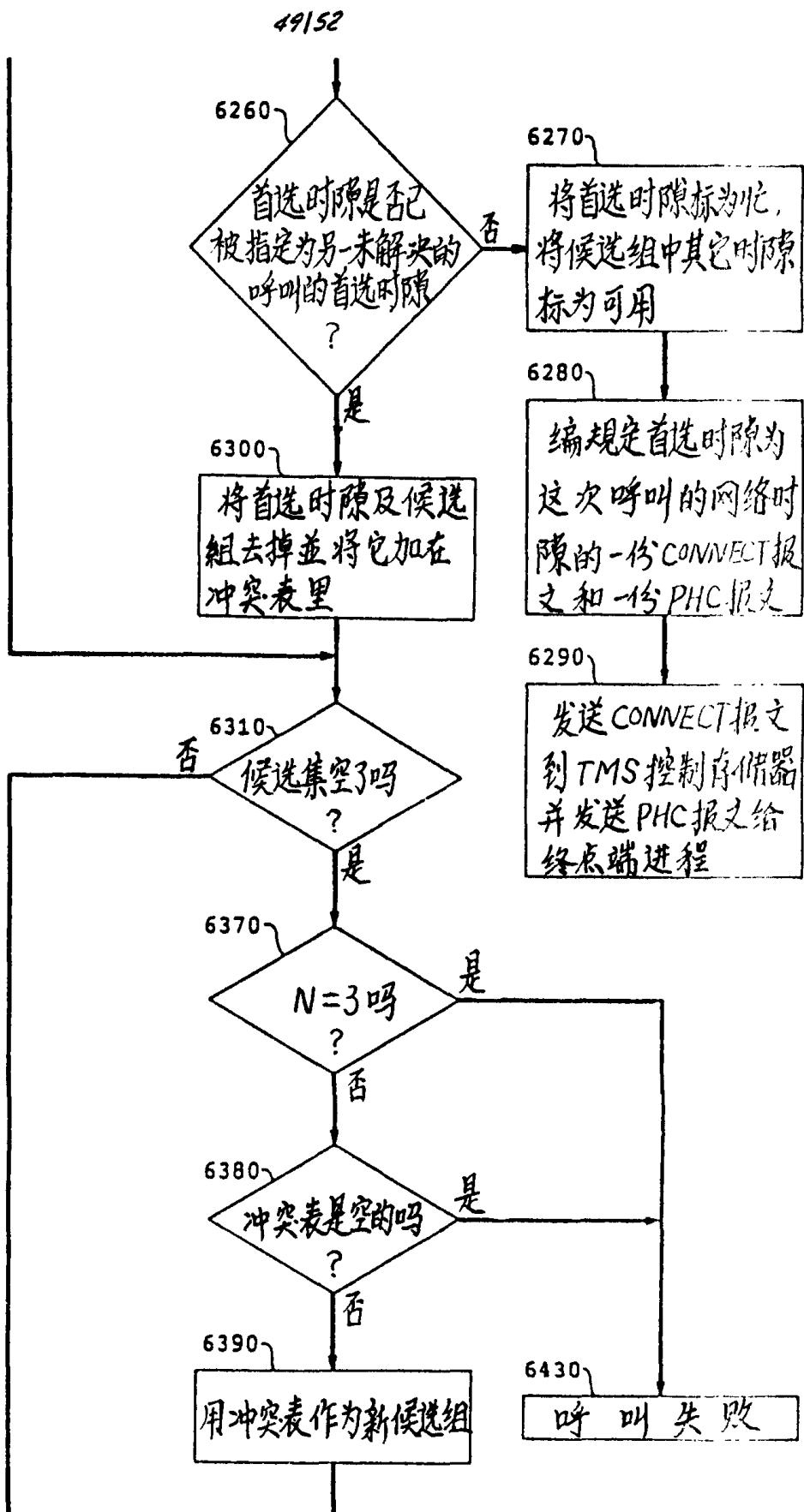
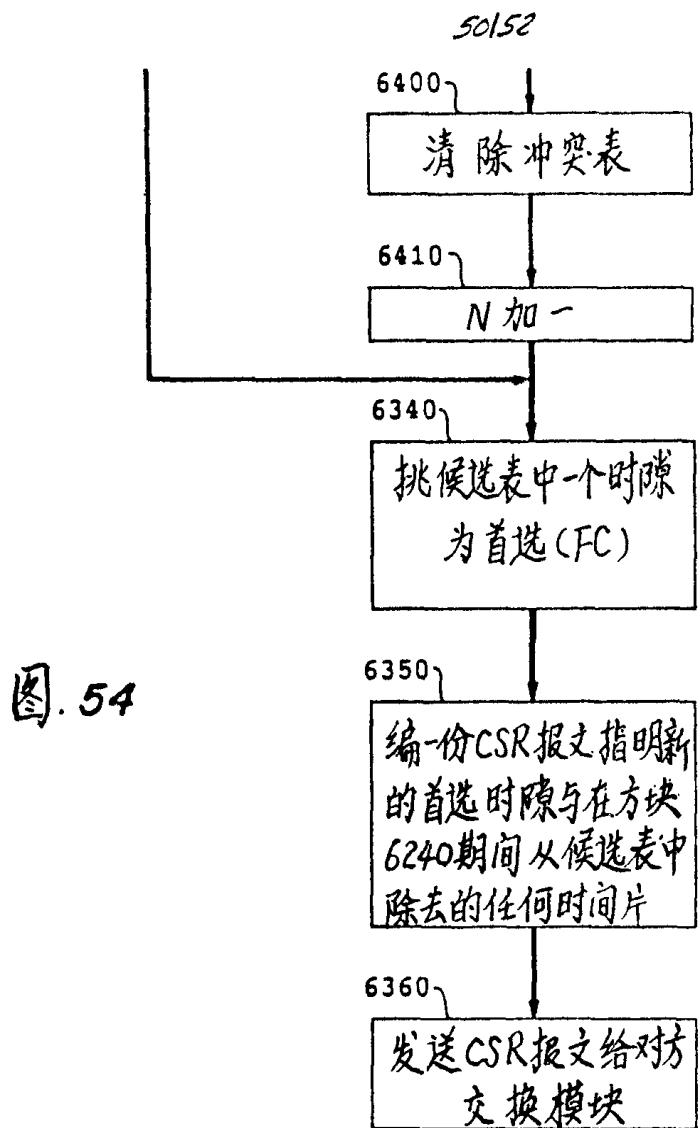


图. 53





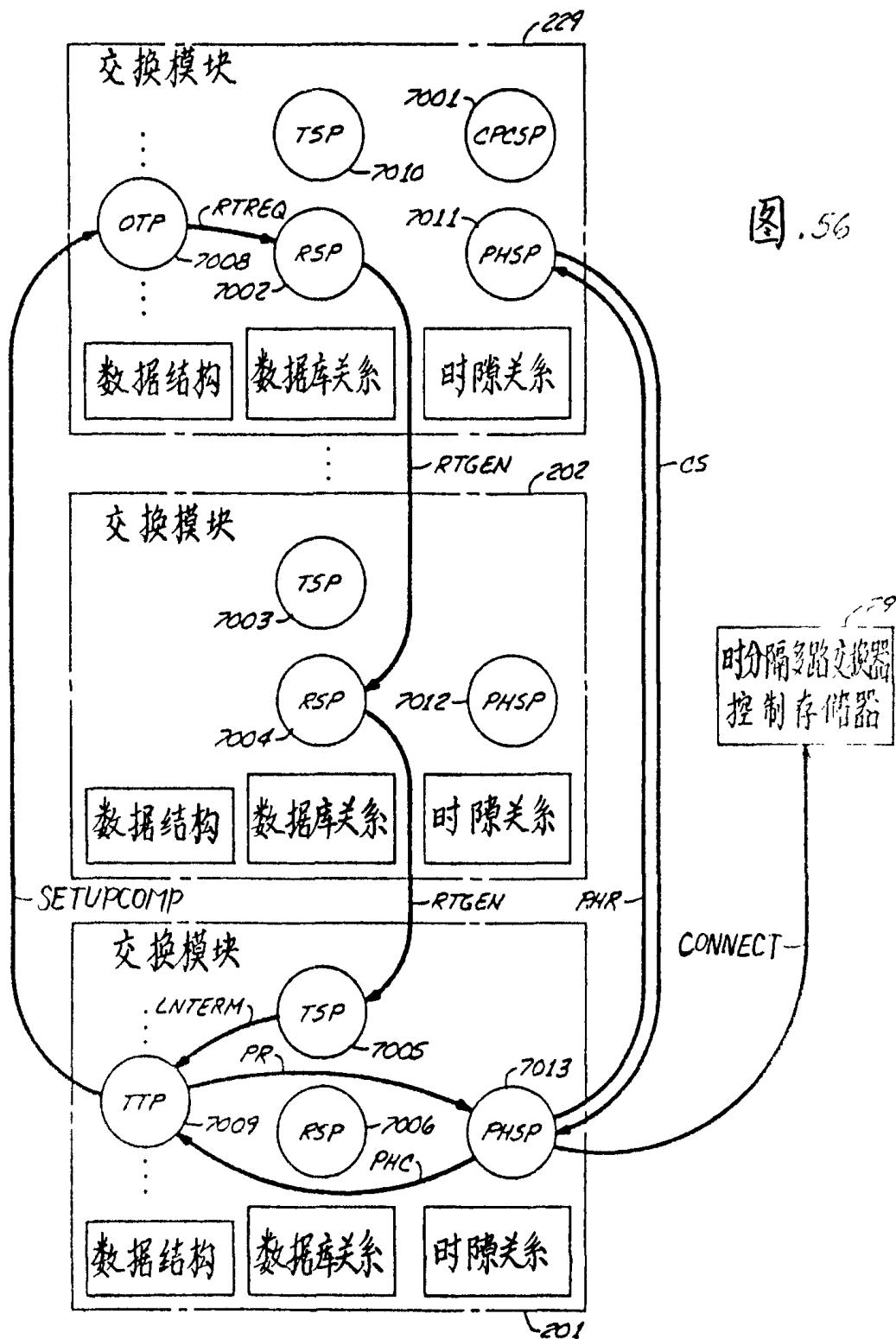


图. 9
图 10
图. 11
图. 12
图. 13

图. 57

图. 25
图. 26
图. 27
图. 28
图. 29

图. 58