

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

H04Q 11/04

H04L 12/56



[12] 发明专利说明书

[21] ZL 专利号 98805279.2

[45] 授权公告日 2003 年 10 月 29 日

[11] 授权公告号 CN 1126424C

[22] 申请日 1998.12.30 [21] 申请号 98805279.2

[30] 优先权

[32] 1998.1.20 [33] US [31] 09/009535

[86] 国际申请 PCT/SE98/02466 1998.12.30

[87] 国际公布 WO99/37117 英 1999.7.22

[85] 进入国家阶段日期 1999.11.19

[71] 专利权人 艾利森电话股份有限公司

地址 瑞典斯德哥尔摩

[72] 发明人 D·卡尔松

审查员 李振华

[74] 专利代理机构 中国专利代理(香港)有限公司

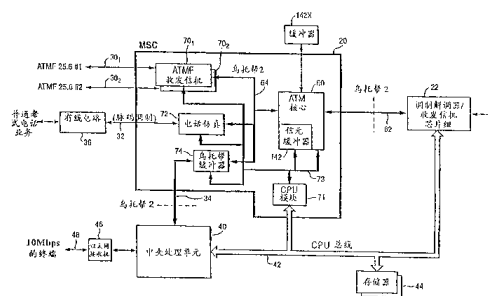
代理人 邹光新 李亚非

权利要求书 2 页 说明书 25 页 附图 18 页

[54] 发明名称 用于电信的多种业务电路

[57] 摘要

一种多种业务硬件集成电路(20)，用来在一个外部接口(62)和电路板上处理不同电信业务的多个业务装置(70₁、70₂、72 和 74)之间传输信元。该在板业务装置包括一个或多个 ATMF 收发信机(70₁、70₂)，一个乌托邦第二层装置(74)，和将 PCM 接口(30₁、30₂)之一跟一个 E1 或一个 T1 接口(32)相连的一个电路仿真器(72)。这种多种业务电路包括一个复用器/分用器核心(60)，该核心跟外部接口(62)相连，并通过一个内部接口(64)跟多个业务装置相连。在这里的实施方案里，外部接口(62)和内部接口(64)都是乌托邦第二层。



1. 一种多种业务电路, 用来从调制解调器/收发信机接收外部接口上的 ATM 信元, 用一个处理器来控制这种多种业务电路, 该多种业务电路包括:
- 5 处理不同电信业务的多个业务装置;
 连接在多个业务装置和外部接口之间的一个复用器/分用器核心, 该核心有
- 一个下行侧, 用于从外部接口向业务装置传送信元, 还有一个上行侧用来从业务装置向外部接口传送信元, 该下行侧有一个下行分用器和一个下行复用器, 上行侧有一个上行复用器和一个上行分用器,
- 10 - 一个下行回送缓冲器用于储存从下行侧往上行侧去的信元;
 - 一个上行回送缓冲器用于储存从上行侧往下行侧去的信元;
 - 其中的下行分用器用于将来自外部接口的信元传送到下行回送缓冲器、处理器和下行复用器的一个输入端这三个中的一个;
- 15 - 其中的下行复用器用于获得来自下行分用器、上行回送缓冲器和处理器的信元, 并传输给业务装置;
 - 其中的上行分用器用于将来自业务装置和处理器的信元传送给上行回送缓冲器、处理器和上行分用器和上行复用器之间的缓冲区中的一个; 和
- 20 - 其中的上行复用器用于从缓冲区和下行回送缓冲器之一获得信元, 并传递给外部接口。
2. 权利要求 1 的装置, 其中的下行分用器和下行复用器都能够同时独立地工作, 除非要将下行分用器的信元传送给下行复用器。
3. 权利要求 2 的装置, 其中至少有一个业务装置是一种 ATM 收发
- 25 信机。
4. 权利要求 2 的装置, 其中至少有一个业务装置是跟 (1) 一个 PCM 接口 (2) 一个 E1 接口 (3) 一个 T1 接口中的一个相连的一种仿真器。
5. 权利要求 4 的装置, 其中的仿真器有一个缓冲器, 用一个信道的数据或者全填充, 或者部分填充。
- 30 6. 权利要求 4 的装置, 其中的仿真器有一个缓冲器, 用所有信道的数据或者全填充, 或者部分填充。
7. 权利要求 2 的装置, 其中至少有一个业务装置是一种乌托邦第

二层装置。

8. 权利要求 2 的装置，其中的信元是 ATM 信元。

9. 权利要求 2 的装置，一个乌托邦第二层分支接口将这多个业务装置跟复用器/分用器核心相连。

5 10. 权利要求 2 的装置，其中的多种业务电路是一块集成电路芯片。

11. 权利要求 2 的装置，其中整个多种业务电路是用硬件做成的。

12. 一种多种业务电路，用来从调制解调器/收发信机接收外部接口上的 ATM 信元，用一个处理器来控制这种多种业务电路，将这一多种业务电路做成一块芯片，包括：

10 处理不同电信业务的多个业务装置；

连接在多个业务装置和外部接口之间的一个复用器/分用器核心；

将该核心跟多个外部装置相连的一个内部接口；和

其中在下行方向，该核心将来自外部接口的信元通过内部接口传送到多个业务装置之一，传送到处理器和外部接口；

15 其中在上行方向，该核心将来自多个业务装置的信元通过内部接口和处理器传送给外部接口、处理器和内部接口之一。

13. 权利要求 12 的装置，其中至少有一个外部接口和内部接口是一种乌托邦第二层接口。

20 14. 权利要求 12 的装置，其中的整个多种业务电路是用硬件做成的。

用于电信的多种业务电路

技术领域

- 5 本发明涉及电信领域，特别涉及在外部网络的物理接口上提供多种业务。

背景技术

- 10 当前非常需要利用一种电信网络提供多种业务。例如，在一种物理媒体，例如外部网络的物理接口上，同时提供象视频、话音电话、数据和其它交互式和/或多媒体业务这样的各种业务。

- 这种能够承载多种综合业务的外部网络的物理接口的例子有混合同轴光纤（HFC）网。在 HFC 网里，前端局（headend office）接收各种来源（例如模拟电视、因特网访问、数字视频点播）的信号，并将携带这些信号的光信号传递给分配中心或节点。在分配中心里，这一光
15 信号被转换并重新分配给用户楼里的网络接口单元（NIU）或者网络终端（NT）。网络接口单元用一个内部收发信机（例如调制解调器）来接收该 HFC 信号，并将适当的频道分配给电视、个人计算机和电话等等。

- 其它类型的外部物理接口也正在出现。例如，这些外部网络的物理接口包括光纤到户（FTTH）网络和兆比特高速数字用户线（xDSL）网络。
20 该 xDSL 网络采用专用电话线。

- 这种外部网络除了物理接口以外，还有网络协议接口。一种广泛使用的协议接口是异步传递模式（ATM）。ATM 是一种面向数据包的传递模式，它采用异步时分复用器技术。数据包叫做信元（cell），大小固定。一个 ATM 信元包括 53 个八位字节，其中的 5 位构成报头，48 位构成“有效负荷”或者叫做信元的信息部分。ATM 信元的报头包括两个量，用于
25 识别传递信元的 ATM 网络里的连接，具体而言就是 VPI（虚路径标识符）和 VCI（虚信道标识符）。一般而言，虚路径是在网络的两个交换节点之间定义的一条主路径；虚信道是该主路径上的专用连接。

- ATM 技术制订了许多格式标准和接口标准。例如，ATM 有几个“适配层（adaptation layers），这些是 ITU（国际电信联盟）标准的内容。另外，为叫做“乌托邦第二层（Utopia Level 2）”的 ATM 接口制订了标准，
30 见技术委员会，乌托邦第二层，1.0 版，af-phy-0039.000，1995 年 6 月。

在多种业务环境里，网络接口单元应当非常灵活，不仅能够提供现有的业务，还能够提供其它的业务，并且能够接受其它类型外部网络的物理接口。

5 一些网络接口单元使用拥有辅助硬件模块的处理器核心。例如，摩托罗拉 860SAR（分段与重组）电路有一个控制用处理器核心、用于处理 ATM SAR 功能（例如 AAL5 [ATM 适配层第 5 层]）的一个 SAR 处理器核心以及处理以太网功能的一个以太网处理器。

10 对于用户来说基于处理器的网络接口单元非常灵活，因为通过更换处理器运行的软件，用户能够很容易地修改其功能。但基于处理器的网络接口单元也有缺点。一个缺点是主要功能必须由用户装入软件来实现，这非常困难，而且设计工作量很大。第二个缺点是基于处理器的单元其数据速率有限。第三个缺点是功耗太高。

因此，需要一种灵活高效主要是基于硬件的网络接口单元，这也是本发明的一个目的。

15 发明内容

20 在外部接口和处理不同的电信业务的多个在板上（on-board）业务装置之间有一多种业务集成电路在传递信元。该在板上业务装置包括一个或多个 ATMF 收发信机，一个乌托邦第二层装置，和用来连接 PCM 接口之一的一个仿真器，一个 E1 接口，和一个 T1 接口。这一多种业务电路包括一个复用器/分用器核心，它跟外部接口连接，还通过一个内部接口跟多个业务装置连接。在本文的实施方案里，外部接口和内部接口都是乌托邦第二层接口。

25 复用器/分用器有一个下行侧（downstream side）用来从外部接口将信元传递给业务装置，还有一个上行侧（upstream side）用来将信元从业务装置传递给外部接口。下行侧有一个下行分用器和一个下行复用器；还有一个下行回送（loop-back）缓冲器，用来储存从下行侧发往上行侧的信元。上行侧有一个上行复用器和一个上行分用器，以及一个上行回送缓冲器，用来储存从上行侧发往下行侧的信元。

30 在下行侧，下行分用器将来自外部接口的信元传递到下行回送缓冲器、处理器和下行复用器的输入端中的一个。下行复用器接收下行分用器，上行回送缓冲器和处理器之一的信元，并通过内部接口将信元发往业务装置。

在上行侧，上行分用器将来自业务装置和处理器的信元，传递给上行回送缓冲器、处理器以及上行分用器和上行复用器之间的缓冲区中的一个。上行复用器接收来自缓冲区和下行回送缓冲器之一的信元，并传递给外部接口。

5 附图说明

下面通过参考附图对优选实施方案进行更具体的描述，本发明前面的以及其它的目的、特征和优点将一目了然。附图中相同的标记表示同样的部件。其中的图不一定是按比例画出来的，主要目的是说明本发明的原理。

- 10 图 1 是本发明一个优选实施方案里多种业务电路的原理图。
图 2 是图 1 中多种业务电路里包括的 ATM 核心的原理图。
图 2A 是图 2 中 ATM 核心的下行侧原理图。
图 2B 是图 2 中 ATM 核心的上行侧原理图。
图 3A 是图 2 中 ATM 核心的上行侧分用器所进行的总步骤的流程图。
15 图 3B 是图 2 中 ATM 核心的下行侧分用器所进行的总步骤的流程图。
图 4A 说明如何通过图 2 的 ATM 核心进行 VP 交叉连接。
图 4B 说明如何通过图 2 的 ATM 核心进行 VC 交叉连接。
图 5 是图 2 中 ATM 核心包括的缓冲区的原理图。
图 5A 是图 2 中 ATM 核心包括的缓冲区存储器分配的示意图。
20 图 6 说明图 1 中多种业务电路包括的电路仿真 (CE) 装置如何连接。
图 6A 是图 6 中电路仿真 (CE) 装置的原理图。
图 6B (1) 说明图 6 中电路仿真 (CE) 装置如何对信元进行打包，特别是结构化 64 kps 信道的全填充信元。
图 6B (2) 说明图 6 中电路仿真 (CE) 装置如何对信元进行打包，特别是结构化 64 kps 信道的部分填充信元。
25 图 7A 说明图 6 中电路仿真 (CE) 装置如何对信元进行打包，特别是 E1 传输的信元。
图 7B 说明图 6 中电路仿真 (CE) 装置如何对信元进行打包，特别是 T1 传输的信元。
30 图 8 说明图 6 中电路仿真 (CE) 装置里的信元延迟变化。
图 9A ~ E 说明不同操作模式中 8 位字节到图 6 中电路仿真 (CE) 装置提供的缓冲器的映射。

图 10 说明图 6 中电路仿真(CE)装置中对两个部分填充的单个 64 kbps 载体(carrier)信元进行的拆包操作。

图 11 说明图 6 中电路仿真(CE)装置如何处理丢失的和误插入的信元。

5 图 12 说明图 6 中电路仿真(CE)装置如何对下行数据率进行同步。

图 13 是图 1 中多种业务电路包括的乌托邦缓冲器的示意图。

图 14 是图 1 中多种业务电路包括的 ATMF 收发信机的示意图。

图 15 说明图 1 中多种业务电路包括的时钟模块进行的读写操作。

图 16 说明图 1 中多种业务电路包括的 CPU 模块进行的中断处理。

10 图 17A、B 分别是图 2 中 ATM 核心一个分用器和译码器, 以及一个下行复用器的 VPI/VCI 表。

具体实施方式

在下面的介绍中, 为了进行说明, 而不是进行限制, 阐述了本发明的详细结构、接口、技术等等具体细节, 目的是从整体上获得对本发明的了解。但是, 对本领域的技术人员而言显而易见, 可以用跟这里介绍的具体细节不同的许多其它的实施方案来实现本发明。在某些情况下, 一些众所周知的装置、电路和方法都不再详细介绍, 以免喧宾夺主。

1.0 多种业务电路综述

图 1 说明的是一个多种业务电路 20, 它跟一个调制解调器/收发信机芯片组 22 相连接, 并在不同的业务接口之间传递封装成 ATM 数据包的数据。这一多种业务电路 20 在其间传递数据的具体业务接口包括接口 30₁ 和 30₂ (它们都是 ATMF 25.6 接口), 接口 32 和接口 34 (一种乌托邦第二层接口[从属])。尽管接口 32 被说明成一种 PCM (脉冲编码调制) 接口[例如用于多达 4 条 64 kps 信道通过有线电路 36 支持四项 POTS (简易老式电话业务) 和一项 IDSN (综合业务数字网) 业务], 它还可以是一种 E1/T1 接口。

多种业务电路 20 可以用于, 例如, 多种业务环境里跟业务接口并分配业务的网络终端 (NT), 例如 HFC。多种业务电路 20 的用途不是仅限于 HFC, 而是还可以用于其它类型的网络, 如 xDSL (兆比特高速数字用户线) 和 FTTH (光纤到家)。

多种业务电路 20 依靠中央处理单元 (CPU) 40 来进行工作, 它们之间用 CPU 总线 42 连接。CPU 总线 42 还跟存储器单元 44 和调制解调器/收发信机 22 相连。CPU 总线 42 传递表 6 所示的信号。CPU 40 由以太网 (E/N) 收发信机 46 通过双绞线跟一个 10 Mbps 的物理接口 48 相连。

多种业务电路 20 内部包括一个叫做 ATM 核心 60 的 ATM 复用器/分用器单元, 图 2 中将进一步介绍。在“集合 (aggregate)”一侧, ATM 核心 60 由调制解调器/收发信机接口 62 连接; 在“分支 (tributary)”一侧, 则由业务接口 64 连接。调制解调器/收发信机接口 62 和业务接口 64 都是乌托邦第二层接口。调制解调器/收发信机接口 62, 其中的信号如表 2 所示, 是一个主接口, 它使得调制解调器/收发信机 22 独立。业务接口 64 是一种内部乌托邦接口, 它的定义见 ATM 论坛, 乌托邦第二层。

业务接口 64 将 ATM 核心 60 跟四个业务装置相连。ATM 核心 60 表示 ATM 层, 业务装置表示物理层。具体而言, 业务接口 64 将 ATM 核心 60 跟两个 ATMF 收发信机 70_1 、 70_2 相连, 还跟电路仿真装置 72 和乌托邦缓冲器 74 相连。ATMF 收发信机 70_1 跟 ATMF 接口 30_1 相连; ATMF 收发信机 70_2 跟 ATMF 接口 30_2 相连; ATMF 接口 30_1 和 30_2 承载的信号见表 4。电路仿真装置 72 跟接口 32 相连; 乌托邦缓冲器 74 跟接口 34 相连。接口 32 传递的信号见表 3; 接口 34 传递的信号见表 5。

多种业务电路 20 包括一个 CPU 模块 71, 通过这一模块 ATM 核心 60 跟 CPU 总线 42 并最终跟 CPU 40 相连。内部 CPU 总线 73 将 CPU 模块 71 跟 ATM 核心 60 以及业务装置 70_1 、 70_2 、72 和 74 相连。只有业务接口 64 将 ATM 核心 60 跟业务装置 70_1 、 70_2 、72 和 74 相连。

25 2.0 ATM 核心

图 2 说明 ATM 核心 60 的总体结构。在图 2 里, 内部 CPU 总线 73 将 ATM 核心 60 分成下行侧 (图 2 中总线 73 以上) 和上行侧 (图 2 中总线 73 以下)。

ATM 核心 60 的下行侧包括一个下行分用器和译码器 102, 它的一个输入端跟调制解调器/收发信机接口 62 的接收部分 62R 相连。下行分用器和译码器 102 的其它输出端口跟下行复用器 104、下行 CPU 读缓冲器 106 和下行回送缓冲器 108 相连。下行 CPU 读缓冲器 106 跟内部 CPU 总

线 73 相连。内部 CPU 总线 73 还用于为下行分用器和译码器 102 提供 VPI/VCI 配置信息，见箭头 110。

如上所述，下行复用器 104 的多个输入端跟下行分用器和译码器 102 的一个输出端相连。它还跟一组下行 CPU 写缓冲器和一个上行回送缓冲器 118 相连。下行复用器 104 的一个输出端跟业务接口 64 的一个发射部分 64T 相连。

ATM 核心 60 的上行侧包括一个上行分用器和译码器 122，以及一个上行复用器 124。上行分用器和译码器 122 的第一个输入端跟业务接口 64 的接收部分 64R 相连。上行分用器和译码器 122 的第二个输入端跟上行 CPU 写缓冲器 126 相连。上行分用器和译码器 122 的 VPI/VCI 表由 CPU 40 更新，见 120 线。上行分用器和译码器 122 的多个输出端跟上行回送缓冲器 118、一组上行 CPU 写缓冲器和(通过缓冲区 140)上行复用器 124 相连。

上行复用器 124 的第一个输入端跟下行回送缓冲器 108 相连。上行复用器 124 的第二个输入端跟缓冲区 140 的输出相连。上行复用器 124 的一个输出端跟调制解调器/收发信机接口 62 的发射部分 62T 相连。

缓冲区 140 包括多个内部队列，用于不同类型的业务质量。在这一实施方案中，缓冲区 140 包括第一到第四个缓冲器 $142_1 \sim 142_4$ 。根据例如它们的业务质量，输入缓冲区 140 的信元被传递给队列 $142_1 \sim 142_4$ 中的一个。

另外，还有一个早期数据包丢弃 (Early Packet Discard, EPD) 单元，作为上行分用器和译码器 122 的一部分。当一个信元序列构成一个 PDU (数据包单元)，要储存在一个队列或者缓冲器里时，可以决定要丢弃整个 PDU。当队列或缓冲器的空间不能容纳整个 PDU 时，必须丢弃整个 PDU。丢弃整个 PDU 而不是储存部分 PDU。图 2 对早期数据包丢弃 (EPD) 单元 144 进行了具体说明，其中本来要储存在队列 $142_1 \sim 142_4$ 中之一的一个信元被上行分用器和译码器 122 所丢弃。

在图 2 下行侧的箭头 150 指示的方向上，来自调制解调器/收发信机 22 的信元被分配给业务装置 70_1 、 70_2 、72 和 74，或者分配给下行 CPU 读缓冲器 106，下文将详细说明。还可以从下行 CPU 写缓冲器读出信元，并将它们发送给业务装置 70_1 、 70_2 、72 和 74。在图 2 上行侧的箭头 152 指示的方向上，ATM 核心 60 从业务装置 70_1 、 70_2 、72 和 74 中读出 ATM

信元，并将这些信元分配给一组上行 CPU 写缓冲器 136，或者分配给调制解调器/收发信机 22。还可以从上行 CPU 写缓冲器 126 读出信元，并将它们发送给调制解调器/收发信机 22。

3.0 信元流

5 3.1 下行信元流

调制解调器/收发信机接口 62 的 62R 部分一有信元，就将该信元读出，下行分用器和译码器 102 对输入单元的 VPI/VCI 进行检查。为此，下行分用器和译码器 102 有 VPI/VCI 表，这些表由 CPU 40 进行配置。将参考图 17A 对这些 VPI/VCI 表进行详细介绍。根据输入信元的 VPI/VCI，对下行分用器和译码器 102 里的 VPI/VCI 表进行查询，以确
10 定该信元的物理目的地，同时确定它离开 ATM 核心 60 时应该使用的新的 VPI/VCI。

图 2A 更详细地说明 ATM 核心 60 的下行侧，特别是下行 CPU 写缓冲器组 116。如图 2A 所示，这组 116 下行 CPU 写缓冲器包括缓冲器 116₁、
15 116₂ 和 116₃，其中的每一个都有跟内部 CPU 总线 73 相连的输入端和跟下行复用器 104 的输入端相连的输出端。这每一个缓冲器都跟三个业务装置 70₁、70₂、72 和 74 之一相连，从而使 ATM 核心 60 能够从 CPU 40 将信元发送给拥有缓冲器 116₁、116₂ 和 116₃ 之一的三个业务装置。

图 3A 结合图 2A 说明下行分用器 102 处理来自调制解调器/收发信机 22 的一个信元的总步骤。在步骤 3A-1 里，分用器 102 判断接口 62
20 上是否有来自调制解调器/收发信机 22 的信元。接口 62 上来自调制解调器/收发信机 22 的信元叫做“集合”信元。分用器 102 检查是否有集合信元，如同步骤 3A-1 中做出否判决的分支所示。如果有集合单元，就在步骤 3A-2 里检查它的 VPI/VCI(如上所述)，并暂停下行复用器 104。
25 下行分用器和译码器 102 由 VPI/VCI 就可知道数据包要往哪里去，例如要去业务装置 70₁、70₂、72 和 74 之一，或者 CPU 40(例如下行 CPU 读缓冲器 106)，或者要去下行回送缓冲器 108。这种情况见 10.1 部分和图 17A。

对下行输入信元的 VPI/VCI 完成译码以后，首先在步骤 3A-3 里检查该信元是否要送往 CPU 40，以及下行 CPU 写缓冲器 106 是否准备好接收信元。如果步骤 3A-3 的检查结果是肯定的，就在步骤 3A-4 里将信元写入下行 CPU 读缓冲器 106。

如果步骤 3A-3 的判断结果是否定的,就在步骤 3A-5 里检查是否要将该信元送往下行回送缓冲器 108,并且检查下行回送缓冲器 108 是否准备好。如果步骤 3A-5 里的检查结果是肯定的,就在步骤 3A-6 里将信元发送给下行回送缓冲器 108。

- 5 如果步骤 3A-5 的判断结果是否定的,就在步骤 3A-7 里判断该信元是否乌托邦信元(例如,要发往业务装置 70₁、70₂、72 和 74 中的一个),同时判断该信元要去的具体装置是否准备好接收信元。如果步骤 3A-7 的检查结果是肯定的,就在步骤 3A-8 里请求下行复用器 104 做好准备。不断地向下行复用器 104 发出请求,直到肯定(步骤 3A-9)下行复用器
- 10 104 已经做好准备。如果下行复用器 104 已准备好,就在步骤 3A-10 里将信元发往下行复用器 104,这样就可以通过业务接口 64 的发送部分 64T 将该信元发送倒它要去的具体装置。

- 15 在步骤 3A-7 的检查时,ATM 核心 60 连续更新业务装置 70₁、70₂、72 和 74 中每一个的状态。此时,利用 Tx_Clav 信号对接口 64 进行查询就能够知道每一个装置里的缓冲器是否有足够的空间来储存一个完整的信元。

如果步骤 3A-3、3A-5 和 3A-7 里的判断结果全是否定的,就在步骤 3A-11 里丢弃该信元。完成步骤 3A-4、3A-6、3A-10 和 3A-11 以后,就回到步骤 3A-1 里等待处理新的下行信元。

- 20 跟图 2A 相关联的图 3B 说明下行复用器 104 所经历的总的步骤。在步骤 3B-1 里复用器 104 判断它是否已经被分用器 102 所暂停(见图 3A 中的步骤 3A-2)。如果复用器 104 已经被暂停过,就在步骤 3B-2 里判断分用器 102 是否有信元。当复用器 104 请求分用器 102 做好准备时(见图 3A 的步骤 3A-8),复用器 104 就知道分用器 102 已经有了信元。如
- 25 果分用器 102 没有信元,复用器就回到步骤 3B-1。如果分用器 102 有信元,复用器 104 就在步骤 3B-3 里接收分用器 102 发送给它的信元(见图 3A 中的步骤 3A-10)。然后在步骤 3B-4 里,复用器 104 根据该信元的 VPI/VCI,将该信元传递给乌托邦装置(业务装置 70₁、70₂、72 和 74)中合适的一个。然后,回到步骤 3B-1。

- 30 如果分用器 102 没有暂停复用器 104,复用器 104 就在步骤 3B-5 里检查上行回送缓冲器 118 是否有回送信元。如果上行回送缓冲器 118 有一个信元,就在步骤 3B-6 里检查该单元要去的乌托邦装置(业务装

置 70₁、70₂、72 和 74 中的一个) 是否准备好接收该信元。如果该业务装置没有准备好, 就回到步骤 3B-1。如果业务装置准备好了, 复用器 104 就在步骤 3B-7 里从上行回送缓冲器 118 读出该信元, 在步骤 3B-8 里将该信元通过复用器 104 传递给合适的业务装置。将该信元通过复用器 104 传递给合适的业务装置以后, 程序回到步骤 3B-1。

如果上行回送缓冲器 118 没有回送信元, 就在步骤 3B-9 中判断下行 CPU 写缓冲器 116 之一是否有一个 CPU 信元。如果所有的缓冲器 116 都没有信元, 就回到步骤 3B-1。如果其中一个缓冲器 116 有信元, 就在步骤 3B-10 里判断该 CPU 信元要去的乌托邦装置 (业务装置 70₁、70₂、72 和 74 中的一个) 是否准备好。如果该业务装置没有准备好, 就回到步骤 3B-1。如果有一个 CPU 信元, 就在步骤 3B-11 中从准备好的 CPU 缓冲器 116 里读出该 CPU 信元。在步骤 3B-12 里, 复用器 104 将 CPU 信元传递到合适的业务装置, 然后回到步骤 3B-1。

图 3A 所描述的分用器 102 的工作过程以及图 3B 所描述的复用器 104 的工作过程使得这些装置可以同时独立处理信元。例如, 如果分用器 102 忙于从接口 62 读取信元和将读来的信元存入下行回送缓冲器 108, 复用器就可以从上行回送缓冲器 118 读取信元, 并将这些信元发送到业务装置 70₁、70₂、72 和 74 中的一个。只有当信元是从接口 62 发往接口 64 时, 才需要分用器 102 和复用器 104 一起工作。

3.2 上行信元流

图 2B 更详细地说明 ATM 核心 60 的上行侧, 特别是上行 CPU 写缓冲器组 136。具体地说, 这组 136 上行 CPU 写缓冲器包括缓冲器 136₁、136₂ 和 136₃。这些缓冲器中的每一个都跟三个业务装置 70₁、70₂、72 和 74 中的一个相连, 这使得 ATM 核心 60 能够将信元从拥有三个缓冲器 136₁、136₂ 和 136₃ 之一的三个业务装置发送给 CPU 40。

在 ATM 核心 60 的上行侧, 只要业务接口 64 上的业务装置 70₁、70₂、72 和 74 有一个有信元, 就要立即将其读出来。是否有数据是由接口 64 上的 Rx_C1v 信号来标志的。除了这些业务接口 64 以外, 当包括一个完整的信元时, 也要读上行 CPU 写缓冲器 126。业务装置和上行 CPU 写缓冲器 126 拥有相同的优先级。

在上行方向, 进来的上行信元有八个可能的目的地——上行 CPU 写缓冲器组 136 中的三个缓冲器, 缓冲区 140 中的四个缓冲器, 和上行回

送缓冲器 118。不能只依靠 VPI/VCI 来判断进来的上行信元的目的地。来自不同 ATMF 信道的两个数据单元有相同的 VPI/VCI 这一事实说明还必须利用物理源（例如 ATMF 接口 70_1 、 70_2 ）来判断目的地。跟下行分用器和译码器 102 一样，上行分用器和译码器 122 的 VPI/VCI 表也包括来源信息。上行分用器和译码器 122 的 VPI/VCI 表由 CPU 40 来更新，见 120 线。在 10.2 节和图 17B 里对上行分用器和译码器 122 的 VPI/VCI 表进行更详细的介绍。

因为 ATM 核心 60 非常迅速地读取信元，并将信元存入目标缓冲器，因此 ATM 核心 60 总是可以保证按正确的顺序从不同的业务装置里读取数据。在这一点上，ATM 核心 60 工作得足够快，哪怕是用最快的速度接收所有业务装置的数据，业务装置 70_1 、 70_2 、72 和 74 根本也不可能出现堵塞。

可以在所有的 ATM 连接和缓冲区 140 的所有缓冲器里进行早期数据包丢弃。对于每一个 VPI/VCI 都有信息说明是否进行早期数据包丢弃 (EPD) (根据建立连接时的 CPU 40 的结构)，以及当前的 EPD 状态 (内部变量)。

在上行复用器 124 里将缓冲区 140 的信元和来自下行回送缓冲器 108 的信元进行复用组合，然后交给调制解调器/收发信机接口 62。

在 ATM 核心 60 的上行侧，只要调制解调器/收发信机接口 62 给出信号说明它已经准备好接收一个完整的信元，就可以立即获得一个信元。上行复用器 124 收到这样的信号后，就给下行回送缓冲器 108 中的信元以最高的优先级，并能跟调制解调器/收发信机接口 62 上的四个信道的任意一个相连。缓冲区 140 里缓冲器的处理依赖于 ATM 核心 60 的模式。ATM 核心 60 有三种不同的模式。

在 ATM 核心 60 的第一种模式里，缓冲区 140 的所有四个缓冲器 $142_1 \sim 142_4$ 都跟调制解调器/收发信机接口 62 中的一个信道相连。在这第一种模式里，四个缓冲器 $142_1 \sim 142_4$ 各有不同的优先级。

在 ATM 核心 60 的第二种模式里，两个缓冲器 $142_1 \sim 142_2$ 跟调制解调器/收发信机接口 62 的一个信道相连，另两个缓冲器 $142_3 \sim 142_4$ 跟调制解调器/收发信机接口 62 的另一个信道相连。在这第二种模式里，跟同一个信道相连的两个缓冲器优先级不同，但跟另一个信道相连的另两个缓冲器则有相同的优先级。

在 ATM 核心 60 的第三种模式里，四个缓冲器 $142_1 \sim 142_4$ 中的每一个都跟调制解调器/收发信机接口 62 中单独（例如不同）的一个信道相连。在这第三种模式中，所有四个缓冲器 $142_1 \sim 142_4$ 都有相同的优先级。

4.0 VPI/VCI 处理

- 5 每一个 ATM 连接都有两个 VPI/VCI——一个用于跟调制解调器/收发信机接口 62 连接（例如在集合侧），另一个用于跟业务接口 64 连接（例如分支侧）。因为电路仿真器 72 生成上行信元时 VPI/VCI 是固定不变的，因此，必须将该 VPI/VCI 值用于分支 VPI/VCI。关于如何利用 VPI/VCI 表对 VPI/VCI 值进行译码，将在 10.0 节和图 17A、图 17B 里介绍。ATM 连接的实例见表 1。

- 10 ATM 核心 60 可以同时处理总共 128 个 VPC 和 VCC ATM 连接。在集合侧，利用了 VPI 的所有 12 位，但只有 16 种组合同时有效。高八位用于过滤信元，这在 HFC 应用里是必须的，其中的每一个 NT 都有它自己的 VPI。低四位（16 种组合）将决定 VPC/VCC（跟 VCC 的 VCI 一起）在 ADSL（非对称数字用户环线）应用里，可以将高八位复位。只利用 VCI 的低八位（LSB）。所有 256 种 VCI 组合和 16 种类 VPI 组合都能混合，但只有 128 种组合同时有效。

在分支侧，只利用了 VPI 的四位（低四位），VCI 也只利用了八位（低八位）。所有的组合都能混合，但只有 128 种组合同时有效。

- 20 图 4A 说明如何通过 ATM 核心 60 建立 VP 交叉连接。图 4A 包括一个分用器和译码表 400，它们储存在 ATM 核心 60 中的一组内部随机存取存储器里，并由 CPU 40 来维护。通过 ATM 核心 60 的 128 个连接中的任意一个都能配置成 VP 交叉连接（VPC），其中 16 个连接跟 VPC 一起配置。在这种情况下，将 VPI 的 4 个 LSB 译码。集合侧的 8 个 MSB 必须跟 VP 滤波器对应，而在分支侧则将它们复位，也就是说，不支持任何一般流控（GFC）处理。除非对 OAM（操作管理和维护）而言，对 VPC 而言所有的 VC 都透明。将所有的分段和端到端 F4 流选择出来，并发送给 CPU 40。

- 25 图 4B 说明如何通过 ATM 核心 60 建立 VC 交叉连接。通过 ATM 核心 60 的所有 128 个同时连接都被设置成 VC 交叉连接（VCC）。利用 VCC 处理意味着只将分用和译码表里规定的 VC 通过 ATM 核心 60 分配，包括预先规定的信令 VC（对于 ITU，VC=5；对于 ATM，VC=16。见图 4B）。VCI 的 8 个 MSB 必须是 0。VPI 的 4 个 LSB 和 VCI 的 8 个 LSB 都被译码。VPI

的 8 个 MSB 都用于 VP 交叉连接。将分段和端到端的 F4 OAM 信元按每一个 VP 挑选出来，就象 VPC 一样。但是，分段 F5 信元则是按每一个 VC 来分类。将这些信元发送给 CPU 40。

在 HFC 应用里，要能够广播信元。这是用单独的 VPI 寄存器 402 来实现的，如图 4A 和 4B 所示。对应于这一寄存器的下行信元和 VPI 将被发送给 CPU。

5.0 缓冲

只对上行流进行业务质量 (QoS) 处理。从业务接口 64 (以及从上行 CPU 写缓冲器组 136) 读出来要发往上行 (图 2 中的箭头 152 指示的方向) 的所有信元都储存在缓冲区 140。

图 5 更详细地说明多种业务电路 20 的缓冲区 140。缓冲区 140 实际上包括一个缓冲器控制器 140C，它连接在上行分用器和译码器 122 (分支复用) 和上行复用器 124 (集合复用) 之间。缓冲器控制器 140C 检测内部存储器 (例如 RAM 142) 或者外部存储器 (例如 SRAM 142X，如图 1 所示) 的数据存取。例如，图 2 和图 2B 中的缓冲器 $142_1 \sim 142_4$ 可以包括在内部存储器 (例如 RAM 142) 里。缓冲器 $142_1 \sim 142_4$ 是包括在内部存储器还是外部存储器里，是由 CPU 40 在启动的时候规定的。

因此，多种业务电路 20 的内部缓冲容量是有限的，用 4 个队列 $142_1 \sim 142_4$ 表示。在这一实施方案里，内部存储器的大小是 2048×8 。外部 SRAM 142X 的容量大得多，例如 $128k \times 8$ 。

如图 5A 所示，或者是将内部存储器 142 或者是将外部存储器 142X 分成 4 个区域。如上所述，这 4 个区域在本发明的某些模式下可以对应于不同的信元级别。第一个区域 (区域 1) 总是从地址 $0x0000$ 开始，区域 2~4 则随后。所有区域的大小都是可以用程序控制的，包括所有区域的 EPD 门限值。如上所述，从不同的工作模式的角度来看，4 个缓冲器区域 1~4 不必跟 4 个不同的业务质量级别对应。可以将两个恒定位速率 (CBR) 信元储存在不同的缓冲器区域里，如果其中的一个对时序要求更严。对于每一个 ATM 连接，这在建立的时候就确定好了。

6.0 电路仿真装置

电路仿真 (CE) 装置 72 在 ATM 和同步电话通信 (Synchronous Telephony Traffic) 之间进行转换。在上行方向 (图 2 中的箭头 152)，电路仿真 (CE) 装置 72 从 PCM 接口将同步时隙通信 (Synchronous

Timeslot Traffic) 打包通过 AAL1 (适配层 1) 交给 ATM 信元。将信元交给业务接口 64, 通过上行信道输送。在下行方向(图 2 中的箭头 150), 将来自业务接口 64 的 ATM 信元拆包, 并重建时隙通信 (timeslot traffic), 这也是通过 AAL1 完成的。

5 图 6 说明电路仿真 (CE) 装置 72 的环境, 其中的电路仿真 (CE) 装置 72 连接在业务接口 64 和 PCM 接口 32 之间。有线电路 36 (见图 1) 包括一条双用户线音频电路 (DSLAC), 它跟几个用户线接口电路相连 (SLIC)。SLAC 是一种进行 PCM 调制的电路, 也叫做 CODEC (Coder/DECoder, 编码器/解码器)。一个 DSLAC 电路里有两个 SLAC。

10 SLIC 处理用户线上的高电压和大电流。

图 6A 是电路仿真 (CE) 装置 72 的结构实例。电路仿真 (CE) 装置 72, 就象所有多种业务电路 20 一样, 是一个纯硬件电路。电路仿真 (CE) 装置 72 有一组配置和状态寄存器 72-10, 它们用内部 CPU 总线 73 跟 CPU 40 相连。下面介绍在 72-10 组里利用寄存器。电路仿真 (CE) 装置 72
15 有一个 PCM E1/T1 接口 72-20, 它跟接口 32 相连; 一个 nAAL1 重组单元 72-30, 它跟接口 64 里的发送线相连; 以及一个 AAL1 分段单元 72-40, 它跟接口 64 里的接收线相连。在 PCM 接口 72-20 跟 AAL1 重组单元 72-30 之间是一个下行双口 RAM 72-50。在 PCM 接口 72-20 和 AAL1 分段单元 72-40 之间是一个上行双口 RAM 72-60。双口 RAM 72-50 和 72-60 中的
20 的每一个都分成不同的区域, 下文中将进行介绍。

6.1 打包

电话数据可以打包通过结构化 64 kbps 信道或者无结构 2048/1544 kbps 信道传送。这两种方法都在下文中进行描述。

6.1.1 结构化 64 kbps 信道

25 对于结构化 64 kbps 信道, ATM 信元总是只包括一个信道的数据。这些信元可以是全填充的 (47 个八位字节), 如图 6B (1) 所示, 或者是部分填充的 (22 或者 11 个八位字节), 如图 B (2) 所示。全填充信元的好处是有很高的频带利用率 (100%), 缺点是很长的组装延迟 (47 x 125 ms = 5.9 ms)。部分填充信元的频带利用率较低 (对于 22 个八位
30 字节为 47%, 对于 11 个八位字节位 23%), 但组装延迟也较小 (分别是 2.8 ms 和 1.4ms)。

6.1.2 无结构 2048/1544 kbps 信道

使用无结构 2048/1544 kbps 时, ATM 信元包括或者是 E1 接口 (2048

kbps) (见图 7A), 或者是 T1 接口 (1544 kbps) (见图 7B) 的所有信道的数据。

6.2 信元延迟变化

下行数据 (例如图 2 中箭头 150 的方向) 必须进行额外的延迟, 以
5 处理信元延迟变化 (CDV)。如图 8 所示, 在源和目的地之间有一个标
称传输延迟。如果这一延迟是一个常数, 目的地就可以在信元到达的那
一时刻以后立即开始读数据。但是, 某些信元可能比其它的延迟时间长
得多, 因此当一个信元到得非常晚时, 目的地必须有一个额外的缓冲器。
10 否则, 就会出现缓冲器下溢出。由于组装延迟很长, CVD 处理不会有太
大的额外延迟。在这一实施方案里, 电路仿真 (CE) 装置 72 处理 CDV
的时间长达 3.9 ms。

6.3 丢失的和误插入的信元

通过检查 SAR-PDU 报头里的序列号, 可以检测到丢失和误插入的信
元。当前信元的序列号跟前一个的序列号不连贯时, 它就可能是一个误
15 插入的信元, 但也可能是这一信元和前一信元之间丢失了许多信元。这
可以通过查看下一信元的序列号来确定。如果它跟当前信元的序列号是
连贯的, 就认为丢失了一些信元。如果它跟前一个的连贯, 那么当前的
信元就是误插入的。不连贯的信元将不予储存。(在下一信元到达以后)
20 如果认为信元是误插入的, 不储存它不会带来任何弊端。如果丢失了许
多信元, 就已经造成了危害。

6.4 同步

因为 POTS 是一种同步业务, 所以有必要提取与源有关的时钟信号,
否则就会出现缓冲器溢出或下溢出。在同步系统里, 业务时钟信号是直
25 接从网络时钟信号里提取出来的 (来自调制解调器的下行数据时钟)。
在异步系统里, 通常使用自适应时钟信号提取方法。但这种方法不适合
于结构化的电路仿真, 原因是存在延迟。这样改为由调制解调器提供
一个参考时钟信号, 用于产生本地时钟信号。

6.5 分段

每一个 POTS 信道的数据通常都是恒定的八位字节流, 周期为 125
30 ms。这些八位字节连续地储存在缓冲器 50 里。每一个 POTS 信道的缓冲
器都有 94 个八位字节位置, 对于全填充模式 (图 9A) 这相当于两个信
元, 对于 22 个八位字节部分填充模式 (图 9B), 这相当于 4 个信元,
而对于 11 个八位字节部分填充模式 (图 9C), 这相当于 8 个信元。

6.5.1 分段: 64 kbps 的载体信元 (全填充)

35 当电路仿真 (CE) 装置 72 工作在 64 kbps 载体单元模式时, 缓冲

器被设置成两个区域，每个区域有 47 个八位字节（图 9A）。每一个区域代表一个信元——第一个区域（八位字节 0~46）代表编号为偶数的信元，其它的区域（八位字节 47+）代表编号为奇数的信元。CPU 40 通过在设置寄存器里设定一个专用位来控制每一个信道开始工作。使用单个 64 kbps 载体的信道都是独立初始化的。只要一个区域填满了来自 PCM 接口的数据，就产生一个信元并往上行方向发送。由于没有任何额外的信元缓冲区，因此只有在控制乌托邦接口 64 的装置，也就是 ATM 核心 60 发出信元请求时，才产生信元。ATM 核心 60 收到一个信号，说明已经有一个信元准备好了，然后才立刻发出信元请求，该信号实际上指的是即将产生一个信元。当一个信元产生时，就将 47 个八位字节的 PCM 数据插入 SAR-PDU 有效负荷，同时将一个序列号放入 SAR-PDU 报头中。

6.5.2 分段：64 kbps 载体信元（22 个八位字节）

采用部分填充了 22 个八位字节的 64 kbps 载体信元时，缓冲器被设置成每个都有 22 个八位字节的 4 个区域（图 9B）。如上所述，每一个区域（例如区域 1~4）都表示一个信元：第一个区域（八位字节 0~21）代表序列号为 0 和 4 的信元，第二个区域（八位字节 22~43）表示序列号为 1 和 5 的信元，第三个区域（八位字节 44~65）表示序列号是 2 和 6 的信元，第四个区域（八位字节 66~87）表示序列号是 3 和 7 的信元。甚至在这里，CPU 都允许写入缓冲器。当一个信元区域填满数据（22 个八位字节）时，就给出信号说明能够产生一个信元。信元的产生方法跟前面介绍的方法一样，但只将 22 个八位字节放入 SAR-PDU 有效负荷。剩下的 25 个八位字节是假的八位字节。

6.5.3 分段：64 kbps 载体信元（11 个八位字节）

采用部分填充 11 个八位字节的 64 kbps 载体信元时，缓冲器被设置成有 8 个区域，每一个都有 11 个八位字节。跟前面的模式里一样，每一个区域都代表一个信元。但在这种情况下，每一个序列号都有一个唯一的区域。CPU 启动对缓冲器的写入，即使因为组合延迟较小使得这一点并不关键。信元的产生跟前面介绍的第二种模式一样，差别仅仅在于假八位字节（36）的个数不一样。

6.5.4 同时处理几个 64 kbps 载体

为了同时处理两个 64 kbps 信道，94 个八位字节缓冲器必须加倍，如图 9D 所示。对于往缓冲器里写入的部分而言，这不会带来任何差别。每一个信道都独立于其它的信道储存。对于产生信元的部分，每一个缓冲器都被看成一个不同的信元流，它们就是信元流。当信道 1 即将产生一个信元时，就专门给业务接口 64 一个信号。对于信道 2，根据另一个

信号来给出信号。业务接口 64 向两个信道单元分别发出信元请求。对于有 4 个信道的情形，需要一个 4 x 94 个八位字节的缓冲器，如图 9D 所示。

6.5.5 无结构 E1/T1 帧的打包

- 5 在 ATM 里，E1 (2048 kbps) 和 T1 (1544 kbps) 的传送都是无结构的，也就是说，将数据打包产生全填充信元时没有处理单个的 64 kbps 信道。由于使用的是串行接口，因此没有任何必要对齐数据。在这一模式里，缓冲器是按照图 9E 来设置的。在这一模式里，缓冲器里填充的数据比 n x 64 kbps 模式要快得多，这意味着对写入的初始化没有那么
- 10 严格的要求。

6.6 重组

为重组功能规定了跟分段功能一样的缓冲器。更进一步，这些缓冲器跟分段功能的配置相同。

6.6.1 拆包

- 15 对不同的信元进行的拆包跟打包过程一样。从业务接口 64 收到信元。如果使用了几个单个的 64 kbps 载体，就用不同的启动信号来隔离这些信道。收到一个信元时，就在缓冲器的专用区域理储存用户数据（用序列号来确定）。同时通知读数据的一端，这样它就能知道上次被读取以后，数据是否写进了这一区域。该缓冲器是用一种连续的顺序来连续
- 20 读出的，而且数据流被发送给 PCM 接口 32——每一个缓冲器一个信道。图 10 说明对两个部分填充的单个 64 kbps 载体信元进行拆包。

6.6.2 处理丢失的和误插入的信元

- 电路仿真 (CE) 装置 72 收到一个信元时，就检查 SAR-PDU 报头里的序列号。如果信元 C_t 跟前面的信元 C_{t-1} 不连续，就将它丢弃掉，不储
- 25 存用户数据。如果下一个信元 C_{t+1} 跟信元 C_{t-1} 连续，就认为信元 C_t 是误插入的，重组过程继续下去。如果信元 C_{t+1} 跟信元 C_{t-1} 不连续，就认为在信元 C_{t-1} 和 C_{t+1} 之间丢失了许多信元。即使是在这种情况下，仍然继续重组过程，结果是即使它是一个正确的信元，也将信元 C_t 丢弃掉。然而，不能储存一个可能是误插入了的信元，因为使用全填充信元时，缓冲器
- 30 的空间只能放下两个信元。图 11 说明了一个实例。

- 如果正在读取信元编号为奇数的缓冲器区域，而这时序列号为奇数的误插入信元到达 C_t ，就不能将有效负载存入正被读的区域。由于缓冲器空间很小，不可能用前面建议的假数据代替丢失的信元插入缓冲器。但每次将一个信元有效负载写入缓冲器时，都通知读数据的那一端。
- 35 如果上次读过以后没有任何迹象表明已经将一个信元写入一个区域，读数

据的一侧就产生必要数量的假数据位 (1 s)。

6.6.3 处理信元延迟变化 (CDV)

如果网络里所有的信元都有相同的延迟, 它们就应当以精确的周期出现。此时, 读数据部分可以在第一个信元已经到达的时候开始读缓冲器。实际上, 有些信元延迟更长 (见图 8), 这意味着一个缓冲器区域还没有填满数据时就要读其中的数据 (数据下溢出)。因此, 当第一个信元到达时, 读数据的部分应当在一开始就延迟, 这样当信元被写入以后可以在一段时间里连续地读每一个信元。在此情形, 不会因为一个信元突然迟到一会儿而出现缓冲器下溢出。可以由 CPU 对额外的 CDV 延迟进行编程。

6.7 同步

为了同步电话信号流, 多种业务电路 20 需要一个 8 kHz 的网络参考时钟。利用锁相环 (DPLL) 产生一个时钟信号, 来锁定参考时钟信号, 就象图 12 中为了同步下行数据率一样。当接口 32 是一个 PCM 接口时, 振荡器频率 (f_{osc}) 必须是 32.768 MHz, 产生的时钟频率 (f_{ck}) 必须是 2.048 MHz 的。E1 接口也一样。但如果要用 T₁, f_{osc} 就是 24.704 MHz, f_{ck} 就是 1.544 MHz。该振荡器是一个外部振荡器。

6.8 接口和时钟信号

电路仿真 (CE) 装置 72 跟接口 32、业务接口 64 和内部 CPU 总线 73 20 相连。电路仿真 (CE) 装置 72 分成两个时钟区域。ATM 信元的处理 (在缓冲器和业务接口 64 之间分配数据) 是用业务接口 64 分配的时钟信号 (系统时钟) 来同步的。另一方面, 电话数据的处理 (在 PCM/E1/T1 接口和缓冲器之间分配数据) 是用 DPLL 时钟来同步的 (见图 12)。

7.0 乌托邦缓冲器

图 13 所示的乌托邦缓冲器 74 基本上是 ATM 核心 60 控制的业务接口 (内部分支乌托邦接口) 和跟它相连的外部装置控制的外部业务乌托邦接口 34 之间的一个缓冲器。乌托邦缓冲器 74 里的内部缓冲器每一个方向都可以储存 2 个信元。外部乌托邦接口 34 可以工作在第二层或者第一层模式, 模式的选择由 CPU 40 在开始的时候设置。在第二层模式 30 里, 还必须设置物理地址。乌托邦缓冲器 74 有三个不同的时钟。在乌托邦缓冲器 74 和内部分支接口 (业务接口 64) 之间分配数据是由系统时钟同步的。在乌托邦缓冲器 74 和外部乌托邦接口 34 之间分配数据是由用于收发的两个独立时钟来同步的, 这两个时钟信号都由外部乌托邦接口 34 提供。

35 8.0 ATMF 25.6 收发信机

ATMF 收发信机 70_1 、 70_2 每一个都是用于双绞线的点到点物理 ATM 接口，象 ATM 论坛规定的那样。ATMF 25.6 Mbps 收发信机 70_1 、 70_2 是一个公用的标准接口，是由计算机插卡和机顶盒 (STB) 来提供的。典型的 ATMF 收发信机 70_1 、 70_2 见图 14 中的收发信机 70。

5 在下行方向 (图 2 中的箭头 150 所示)，ATMF 收发信机 70_1 、 70_2 收到的信元是来自业务接口 64 的，临时储存在两个信元深的 FIFO (先入先出) 里以后，都用 25.6 MHz 线 30_1 、 30_2 来传输。在上行方向 (图 2 中的箭头 152 所示)，信元是从 25.6 Mbps 线 30_1 、 30_2 上收到的，并储存在两个信元深的 FIFO 里，ATM 核心 60 可以从中将它们读出来。ATM
10 论坛的 ATM 25.6 Mbps 物理接口规定了依赖于物理媒体 (PMD) 的子层和传输会聚 (TC) 子层的功能。为下行部分提供了一个 8 kHz 的参考信号 (跟电路仿真 (CE) 装置 72 的 DPLL 一样的信号)，这样，就可以在 ATMF 接口 30 上发送定时信息。下行时钟信号是从一个外部 32 MHz 振荡器分出来的。上行时钟信号时从上行数据流里提取出来的。用乌托邦
15 时钟信号 (系统时钟信号) 来同步 FIFO 和业务接口 64 之间的处理。

9.0 CPU 模块

CPU 模块 71 在 (1) 外部 CPU 总线 42 和 (2) ATM 核心 60 和业务装置 70_1 、 70_2 、72 和 74 之间分配数据。CPU 模块 71 还处理中断。CPU 模块 71 完成的所有功能，包括对内部接口 (例如总线 73) 的事务处理，
20 都是用系统时钟信号同步的。

9.1 读写处理

在图 15 里说明 CPU 模块 71 进行的读写处理。当 CPU 40 对多种业务电路 20 进行读写操作时，CPU 模块 71 在片选信号 (CS) 上检测到一个低事务处理 (low transaction)。然后 CPU 模块 71 对地址总线上的
25 信号进行解码，并将地址总线信号的最低几位和块选信号发送给被选中的模块 (例如 ATM 核心 60 或业务装置 70_1 、 70_2 、72 和 74 中的一个)。总线 73 是一种双向数据总线，需要读和写时，将它变成两条独立的总线。当内部读写操作结束时，就用数据传送确认信号来说明。既然读写操作是由系统时钟来同步的，因此这种定时方式依赖于频率。

30 9.2 中断处理

多种业务电路 20 的每一个模块都向 CPU 模块 71 给出至少一个标志。标志储存在 CPU 模块 71 管理的状态寄存器里。状态寄存器里的一位从低事务处理变到高事务处理会导致向 CPU 40 发出一个终端请求 (IREQ)。CPU 40 读取状态寄存器里的数据或者收到中断确认信号 (IACK)
35 时，就放弃这一中断请求。通过将屏蔽寄存器中的某一位置位，可以拒

绝中断请求。图 16 说明中断处理的结构。

1. 多种业务电路 20 里的以下事件跟相应的中断处理标志有关：
在 ATM 核心 60 的下行 CPU 读缓冲器 106 里储存一个完整的信元（见图 2A）。
- 5 2. 在 ATM 核心 60 的上行 CPU 读缓冲器组 136 中的缓冲器 136_1 里
储存一个完整的信元（见图 2B）。
3. 在 ATM 核心 60 的上行 CPU 读缓冲器组 136 中的缓冲器 136_2 里
储存一个完整的信元（见图 2B）。
4. 在 ATM 核心 60 的上行 CPU 读缓冲器组 136 中的缓冲器 136_3 里
10 储存一个完整的信元（见图 2B）。
5. 电路仿真（CE）装置 72 的信道 1 中的失去同步事件，或者缓冲
器溢出或下溢出。
6. 电路仿真（CE）装置 72 的信道 2 中的失去同步事件，或者缓冲
器溢出或下溢出。
- 15 7. 电路仿真（CE）装置 72 的信道 3 中的失去同步事件，或者缓冲
器溢出或下溢出。
8. 电路仿真（CE）装置 72 的信道 4 中的失去同步事件，或者缓冲
器溢出或下溢出。

10.0 ATM 核心的 VPI/VCI 表

- 20 如上所述，分用器和译码器 102 以及下行复用器 104 采用 CPU 40
设置的 VPI/VCI 表。图 17A 说明的是分用器和译码器 102 的 VPI/VCI 表；
图 17B 说明的是下行复用器 104 的 VPI/VCI 表。

10.1 下行分用器和译码器的 VPI/VCI

- 25 如图 17A 所示，分用器和译码器 102 有一个集合 VPI/VCI 识别表
102-10 和一个分支 VPI/VCI 译码和目的地表 102-20。输入分用器和译
码器 102 的信元有某种报头信息，用于有效性比较器 102-30 和一个
VPI/VCI 寄存器 102-40。这某种报头信息包括报头的 VPI 的低四位（LSB）
和 VCI 的高八位。在有效性比较器 102-30 里，将报头跟储存在 VPI 寄
存器 102-32 的一个滤波器值进行比较。如果报头信息有效，就将有效
30 信号发往控制器 102-50。

- 分用器和译码器 102 试图在集合 VPI/VCI 识别表 102-10 中找出跟
储存在 VPI/VCI 寄存器 102-4 中的报头信息匹配的值。如图 17A 所示，
VPI/VCI 识别表 102-10 实际上包括四个 RAMS 102-10(1) ~ (4)。RAMS
102-10(1) ~ (4) 中每一个的读出口跟比较器 102-60(1) ~ (4)
35 中某一个的第一个输入口相连。比较器 102-60(1) ~ (4) 中每一个的

第二个输入口连起来以接收 VPI/VCI 寄存器 102-40 中储存的值。RAMS 102-10(1) ~ (4) 中的每一个都有 32 个位置 (32 x 4 = 128 字节 RAM)。分用器和译码器 102 收到一个新的信元时, 将报头信息存入 VPI/VCI 寄存器 102-40, 此时同时检查所有四个 RAM 的第一个位置。也就是说将四个 RAM 第一个位置的值输出给相应的比较器 102-60, 从而判断第一个位置的值是否跟收到的报头信息一致。如果不相同, 就用同样的方式同时检查所有四个 RAM 的第二个位置, 这样进行下去直到找到相同值。这样搜索所有四个 RAM 所需要的最长时间是 32 个时钟周期。

找到匹配的值时, 确定一个索引值, 用于访问分支 VPI/VCI 译码和目的地表 102-20, 并据此为该信元离开分用器和译码器 102 准备一个新报头。如图 17A 所示, 索引值有两个部分。第一部分用于从四个 RAM 中的一个获得相同值的地址或值。第二部分是在找到相同值时从四个比较器 102-60 的四个输出信号转换获得的一个 2 位宽地址。这一 2 位宽的转换地址是从变换器 102-70 获得的, 它的输入端跟比较器 102-60 中每一个的输出端相连。索引值指向分支 VPI/VCI 译码和目的地表 102-20 的位置, 可以利用它获得新报头和目的地值。分支路由选择的新报头有 VPI 的四位和 VCI 的八位。目的地值是说明分支乌托邦装置或 CPU 40 或下行回送缓冲器 108 的一个四位值。

因此, 如上所述, 分用器和译码器 102 有一组集成 RAM 表, 这些表由 CPU 40 来设置。从 CPU 40 的透视图可知, VPI/VCI 表的每一个位置都有一个唯一的地址。找到 VPI/VCI 表的一个地址有一个跟输入信元报头的 VPI/VCI 相同的 VPI/VCI 时, VPI/VCI 表中的相应位置就有新的目的地(例如 CPU 40 之一、回送缓冲器 108 或者业务装置)和新的 VPI/VCI。

如果回送表里的 VCI 域被复位 (VCI=0), 就认为该 ATM 连接是一种 VPC, 这意味着信元报头的 VCI 可以是任意值。此时, 只有 VPI 必须相同, 不译码 VCI。利用 VCI=0 来定义 VPC 的方法非常灵活, 因为 VCI=0 时 ATM 连接的一个未定义值, 也就是说在多种业务电路 20 里不会出现 VCI=0 的信元。当空闲信元和物理 OAM 有 VCI=0 时, 就在调制解调器里将这些信元挑选出来。除了用 VCI=0 来指明 VPC 以外, VPC 的状态还可以用回送表中的另外一位来说明。

10.2 上行复用器的 VPI/VCI 表

如图 17B 所示, 上行分用器和译码器 122 有一个分支 VPI/VCI 识别表 122-10, 一个集合 VPI/VCI 译码表和目的地表 122-20, 以及一个 EPD 状态表 122-25。进入上行分用器和译码器 122 的信元有特定的报头信息和对应的乌托邦地址值 (4 位), 储存在一个 VPI/VCI 寄存器 122-40

里。这一特定的报头信息包括报头 VPI 的低四位 (LSB) 和 VCI 的高八位。

如分用器和译码器 102 那样, 上行分用器和译码器 122 尝试寻找 VPI/VCI 寄存器 122-40 里储存的报头信息和乌托邦地址跟分支 VPI/VCI 识别表 122-10 里匹配的值。如图 17B 所示, VPI/VCI 识别表 122-10 包括四个 RAMS 122-10 (1) ~ (4)。RAMS 122-10 (1) ~ (4) 中的每一个都有一个读出口跟比较器 122-60 (1) ~ (4) 中对应的一个的第一个输入口相连。比较器 122-60 (1) ~ (4) 中每一个的第二个输入口连接起来以接收储存在 VPI/VCI 寄存器 122-40 里的值。上行分用器和译码器 122 收到一个新的信元时, 将报头信息和乌托邦地址存入 VPI/VCI 寄存器 122-40, 在这种情况下, 在 RAMS 122-10 里用前面的 RAMS 102-10 所用的相同方法来搜索相匹配的值。

找到相匹配值时, 确定一个索引值用来访问集合 VPI/VCI 译码和目的地表 122-20 以及 (当它被使用时) EPD 表 122-125。从编制了索引的集合 VPI/VCI 译码和目的地表 122-20 里, 为要离开上行分用器和译码器 122 的信元获得一个新报头。跟分用器和译码器 102 一样, 如图 17 所示, 这一索引值有两个部分。第一部分是用来从四个 RAMS 122-10 中的一个获得相同值的地址或值。第二部分是获得这一相匹配值时, 从四个比较器 122-60 四个输出信号的变换获得的 2 位宽的地址。这 2 位宽的地址是从变换器 122-70 获得的, 该变换器由一些输入端跟比较器 122-60 中的每一个的输出端相连。该索引值指向集合 VPI/VCI 译码和目的地表 122-20 中的某个位置, 从这里获得新的报头。该分支路由选择的新报头有四位 VPI 和八位 VCI。

VPI 滤波器寄存器 102-32 用来在信元发送到集合接口之前, 将一个新的 VPI 值 (高八位) 插入信元的报头, 在下行数据流里, 只接受 (除非是广播信元) 其 VPI 对应于寄存器 102-32 的集合信元, 将它们译码以后, 将 VPI 的高八位复位。在上行数据流里, 只接受 VPI (高八位) 等于零的分支信元, 将它们译码以后, 将寄存器 102-32 的值插入 VPI (高八位)。

对于集合 VPI/VCI 译码和目的地表 122-20 中的每一个位置 (地址), 在 EPD 表 122-25 中都有一个对应的位置。该 EPD 表 122-25 包括每一个 ATM 连接需要用来处理 EPD 的信息。只有在 VPI/VCI 识别表 122-10 中的一个 EPD 的选择位被置位时才使用该 EPD 表 122-25。

11.0 后记

因此 ATM 核心 60 的中心部分是多种业务电路 20。ATM 核心 60 有集

成回送缓冲器、CPU缓冲器和业务质量缓冲器。更有利的是，ATM核心60非常灵活，其结构和工作过程不依赖于分支乌托邦接口，例如业务接口64，所连接的业务装置的类型。

ATM核心60是一种集成电路，在前面的实施方案里，该集成电路支持业务接口64上的八个信道。为其中三个信道的每一个方向提供了一个CPU缓冲器。这意味着，可以在CPU40和这三个业务装置中的每一个之间传送信元。

业务接口64提供的多个物理乌托邦接口还可以集成未来的业务（例如，AAL5 SAR和以太网）。此外，虽然在实施方案里有八条信道，但是ATM核心60可以扩展到更多的信道（例如16条信道）。

本发明的多种业务电路20是一种基本上全部是基于硬件的集成电路。因此，这一多种业务电路20跟基于处理器的电路相比有其它的许多优点，例如更高的数据率和更低的功率消耗。

虽然介绍本发明时，采用了目前认为最为实用的最佳实施方案，很显然，本发明并不局限于公布的实施方案，相反，它还覆盖了所附权利要求所包括的各种修改和等价方案。例如，可以在多种业务电路20里加上CPU40。

表1 通过NT的ATM连接的实例

ATM连接	VPI/VCI 集合侧	VPI/VCI 分支侧	QoS 级
CE信道1 (VCC)	VPI ₁ /43	CE1_fix	QoS1
CE信道2 (VCC)	VPI ₁ /44	CE2_fix	QoS1
业务乌托邦 (VCC)	VPI ₁ /48	0/35	QoS2
ATM#1 (VCC)	VPI ₁ /55	0/35	QoS3
ATM#1 (VPC)	VPI ₁ /-	1/-	QoS3
ATM#2 (VCC)	VPI ₁ /58	0/35	QoS3
CAM CPU<->ATMF#1, F4分段	-	0/3	-
CAM CPU<->ATMF#1, F5分段	-	0/35	-
CAM CPU<->集合, F4端到端	VPI ₁ /4	-	QoS4
CPU<->集合, 信令 (VCC)	VPI ₁ /33	-	QoS4
CPU<->集合, 信令 (VPC)	VPI ₂ /-	-	QoS4

表 2 到调制解调器/收发信机的乌托邦第二层接口

信号名称	类型	宽度	说明
TX_DATA	输出	8	发送数据。
TX_CLK	输出	1	发送时钟信号。
TX_ENB	输出	1	启动发送八位字节
TX_SOC	输出	1	发送信元的开头
TX_CLAV	输入	1	发送信元可用空间信号。
TX_ADDR	输出	5	发送地址。
RX_DATA	输入	8	接收数据。
RX_CLK	输出	1	接收时钟信号。
RX_ENB	输出	1	启动接收八位字节
RX_SOC	输入	1	接收信元的开头
RX_CLAV	输入	1	接收信元。
RX_ADDR	输出	5	接收地址。

表 3 PCM 和 E1/T1 接口

信号名称	类型	宽度	说明
TxCLK	输出	1	发送时钟信号, 2048 或者 1544 kHz。
RxCLK	输入	1	接收时钟信号, 2048 或者 1544 kHz。 只为 E1/T1。
TxD	输出	1	发送数据。
RxD	输入	1	接收数据。
FS	输出	1	开始 PCM 帧。
DV	输出	4	每一信道的数据都有效。

表4 ATMF接口

信号名称	类型	宽度	说明
TxD	输出	2	差分发送数据。
RxD	输入	2	差分接收数据。
EQ	Bidir	2	均衡器的外部滤波器。
PLL	Bidir	2	PLL 的外部滤波器。
AVCC	Bidir	4	模拟信号电源。
AGND	Bidir	4	模拟信号地。

表5 外部业务乌托邦接口

信号名称	类型	宽度	说明
TX_DATA	输入	8	发送数据。
TX_CLK	输入	1	发送时钟信号。
TX_ENB	输入	1	启动发送八位字节
TX_SOC	输入	1	发送信元的开头
TX_CLAV	输出	1	发送信元可用空间信号。
TX_ADDR	输入	5	发送地址。
RX_DATA	输出	8	接收数据。
RX_CLK	输入	1	接收时钟信号。
RX_ENB	输入	1	启动接收八位字节
RX_SOC	输出	1	接收信元的开头
RX_CLAV	输出	1	接收信元。
RX_ADDR	输入	5	接收地址。

表 6 外部 CPU 接口

信号名称	类型	宽度	说明
CS	输入	1	片选
R/W	输入	1	启动读/写
ADDR	输入	12	地址总线
DATA	Bidir	16	数据总线
IREQ	输出	1	中断请求
DTACK*	输出	1	数据传输确认
IACK*	输出	1	中断确认
OE*	输入	1	启动输出
BMODE*	输入	1	总线模式 (16/8 位)
BSEL*	输入	1	字节选取

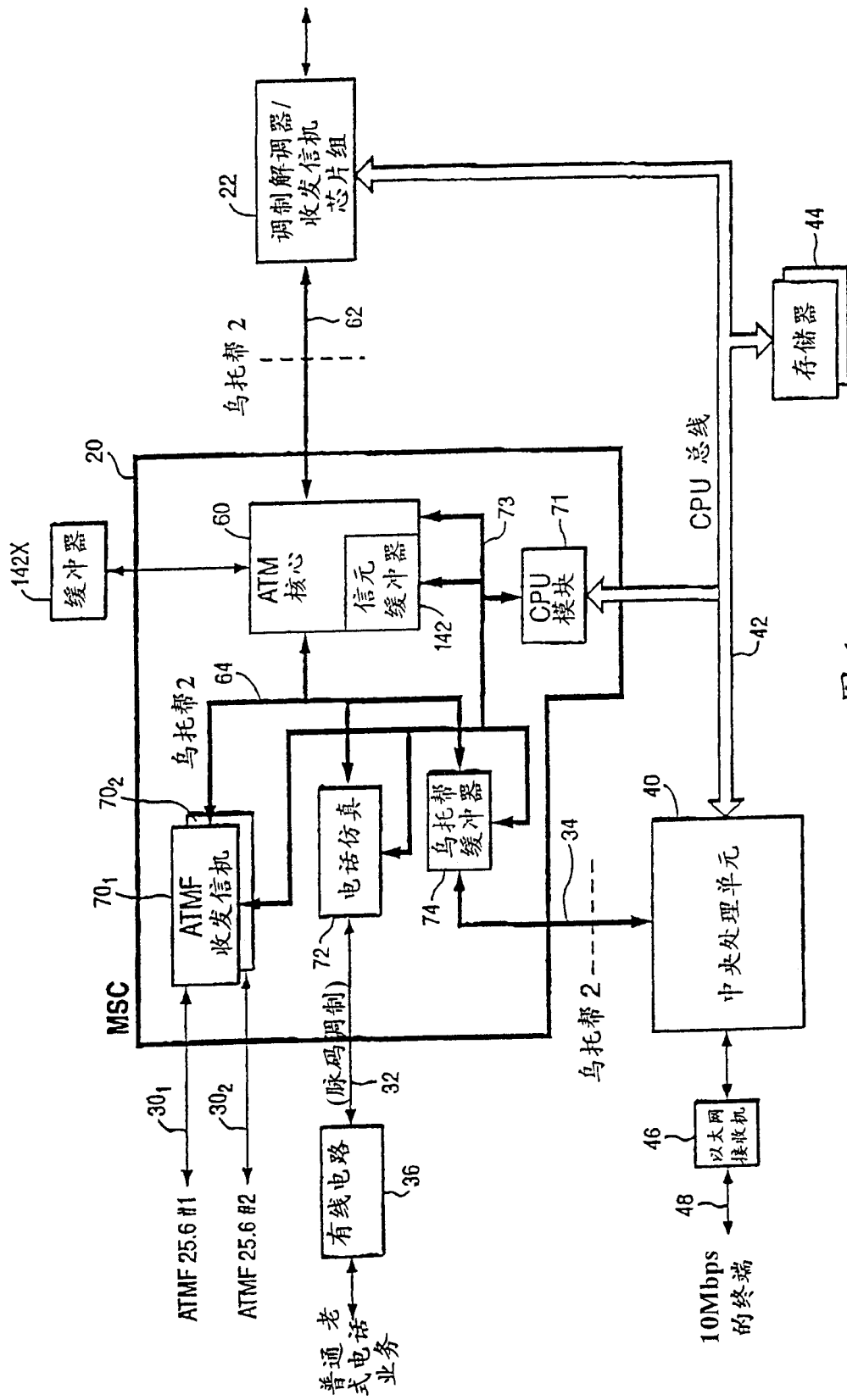


图 1

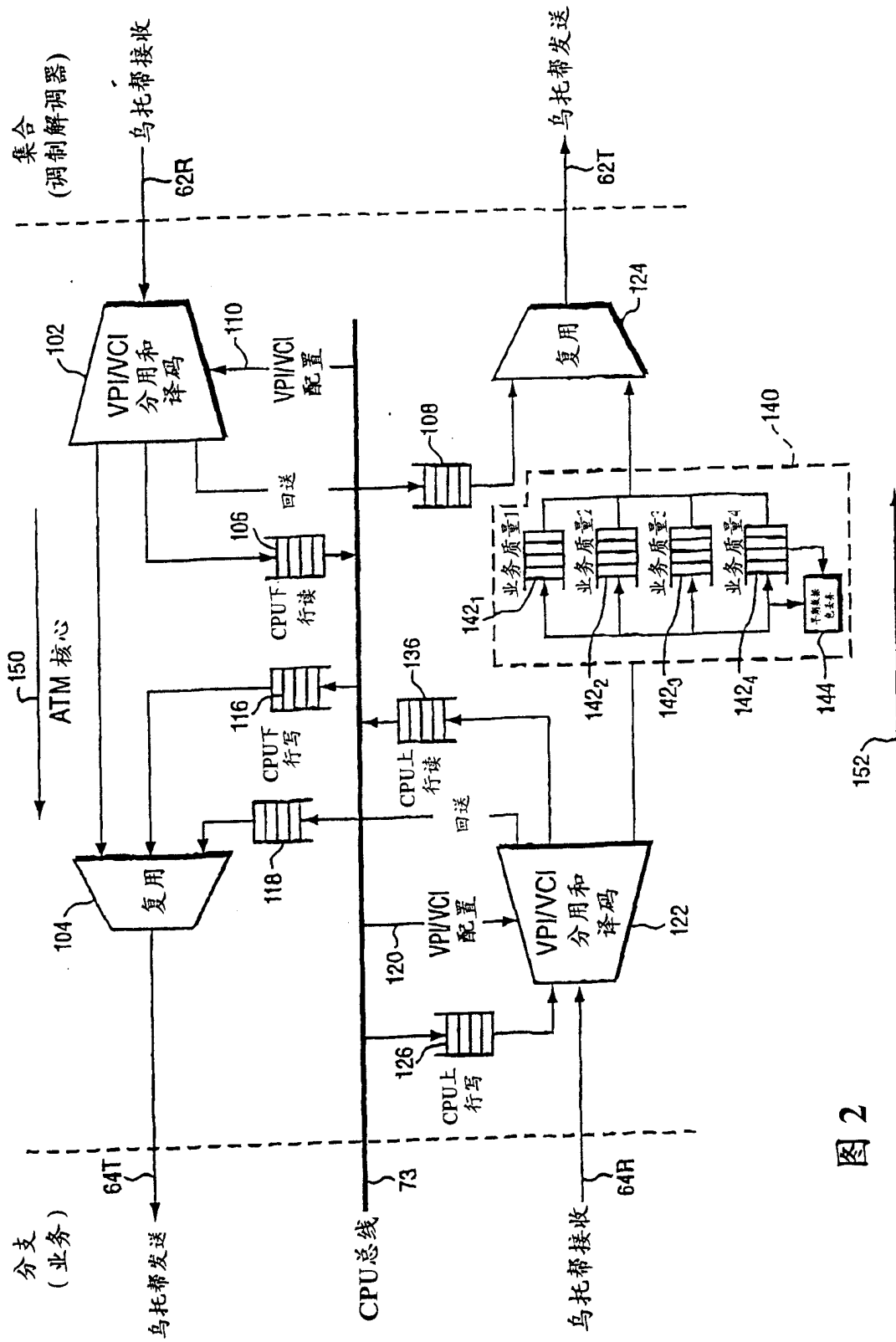


图 2

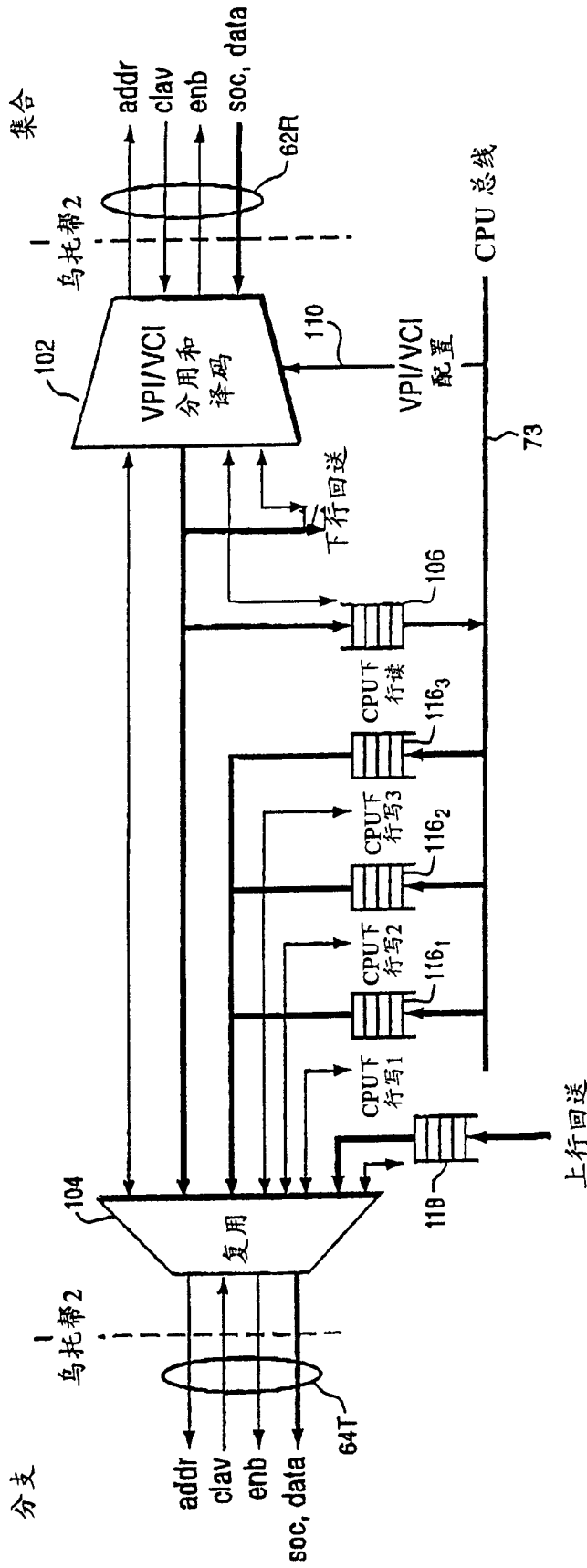


图 2A

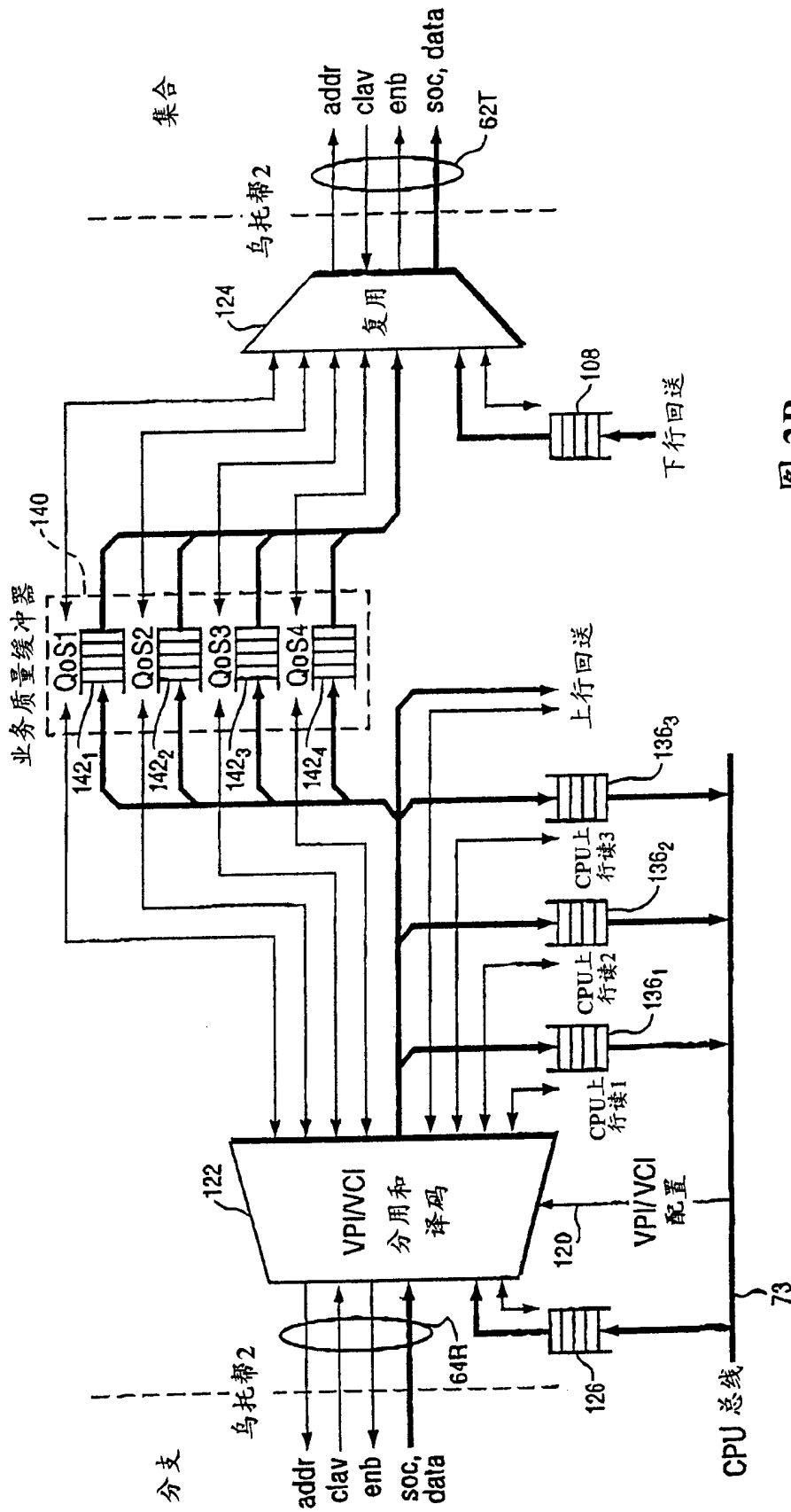


图 2B

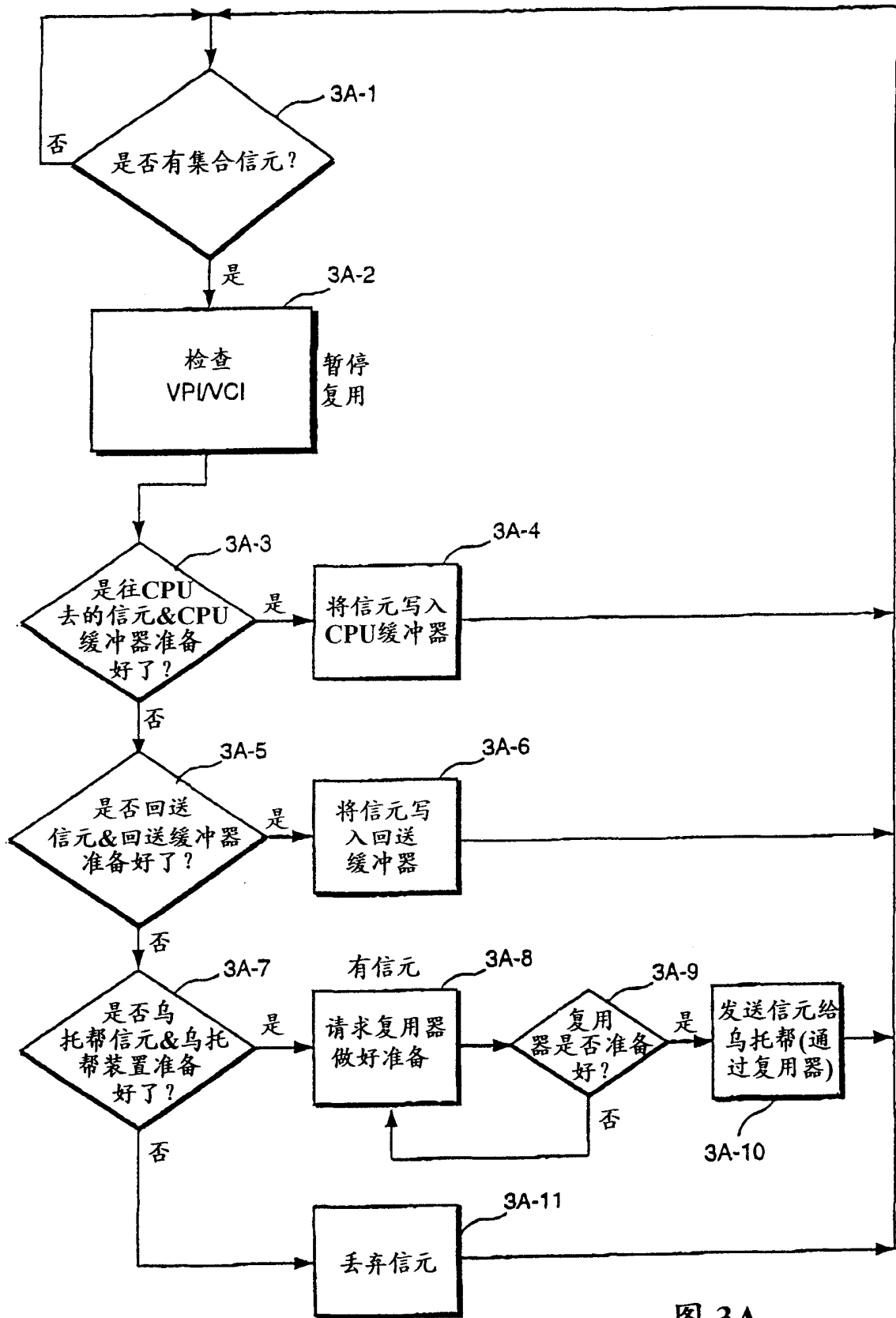


图 3A

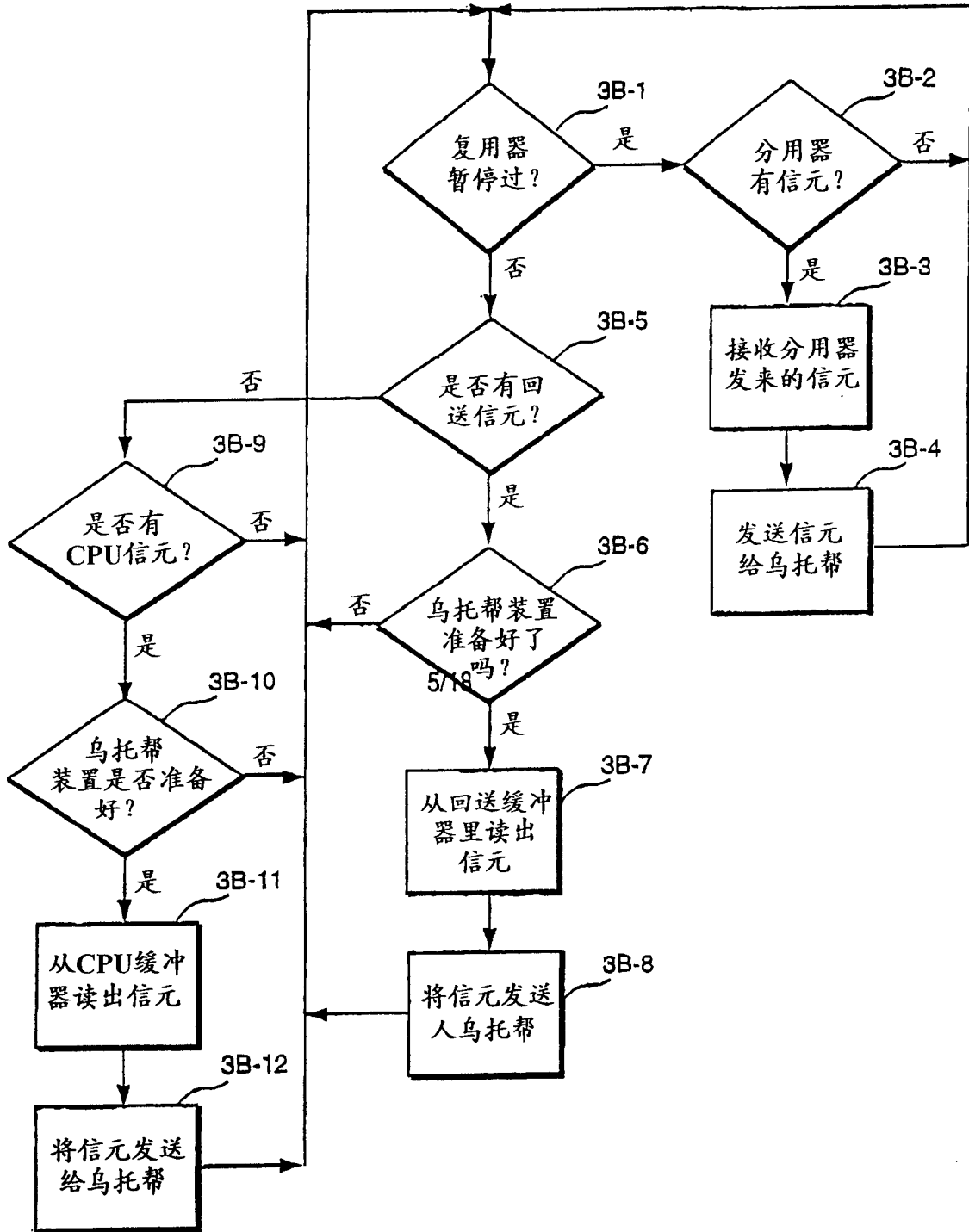
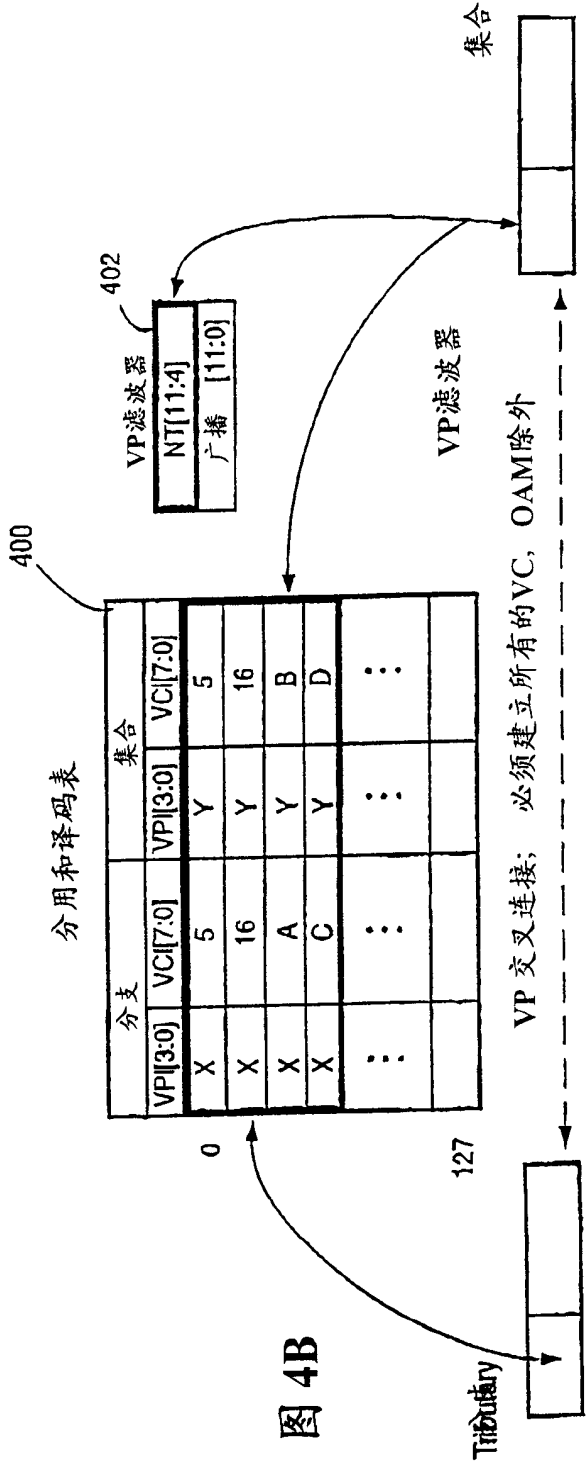
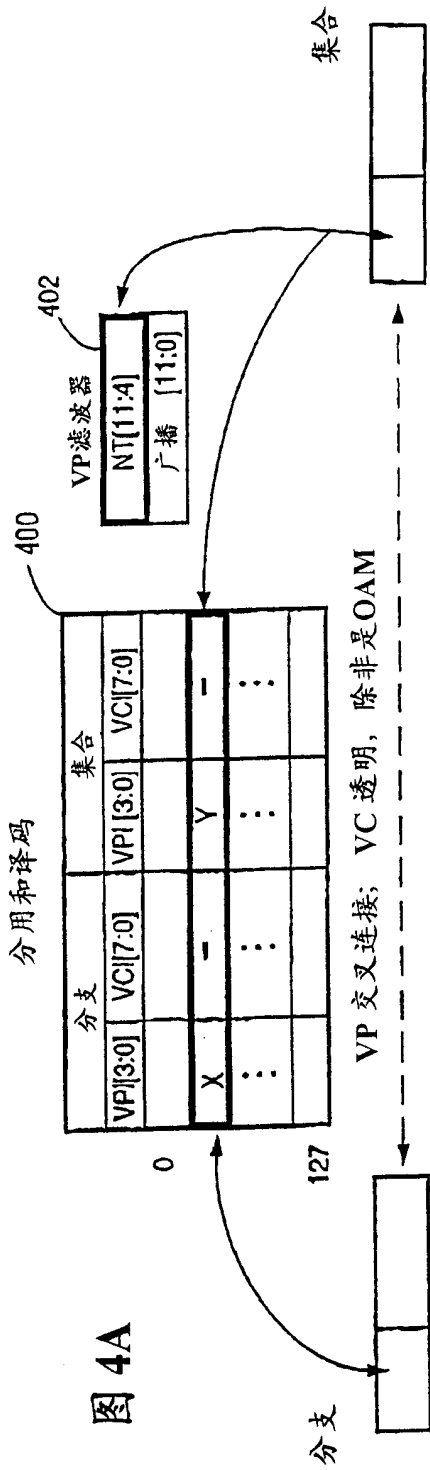


图 3B



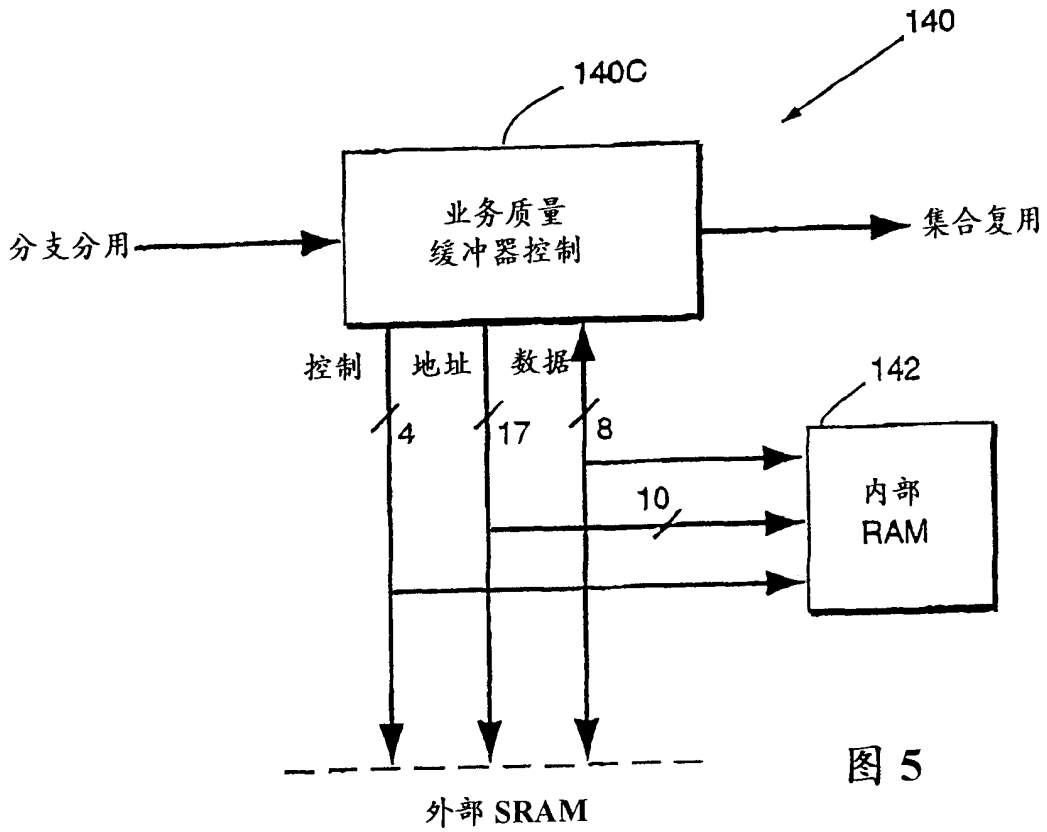


图 5

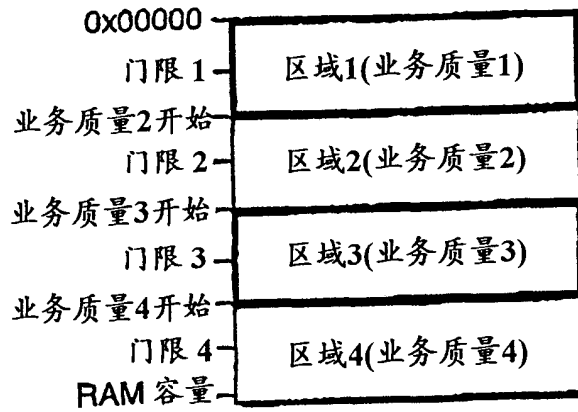


图 5A

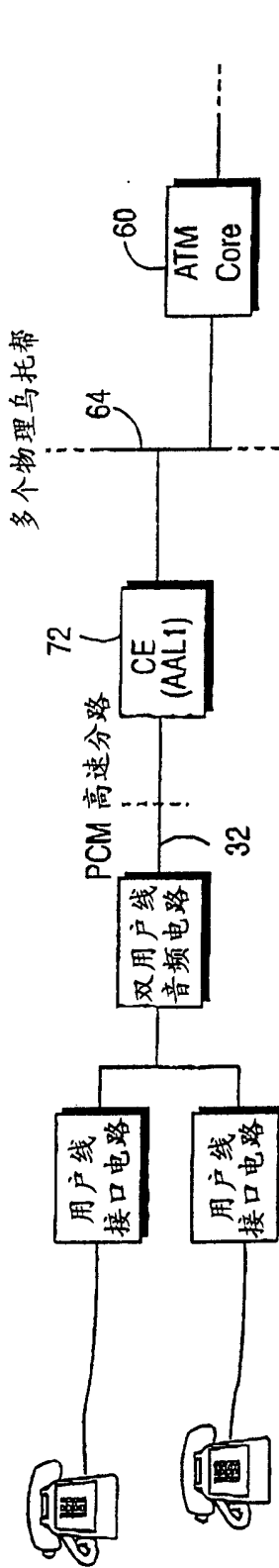
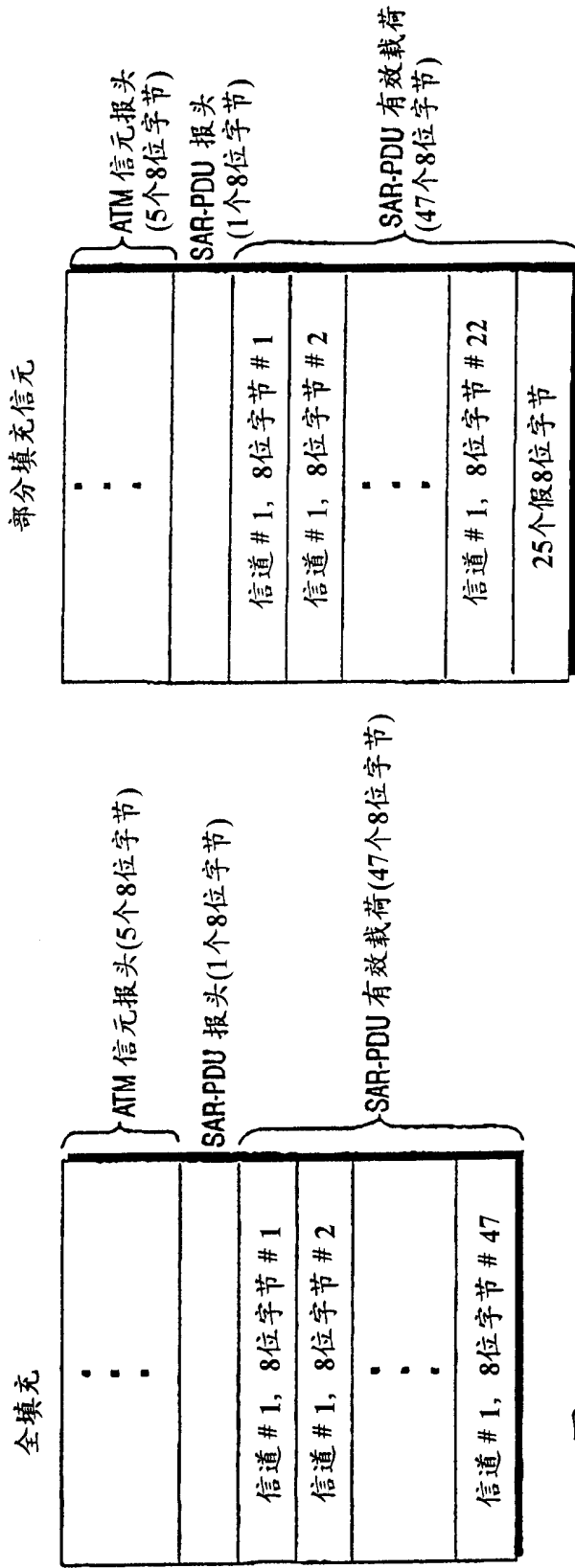


图 6



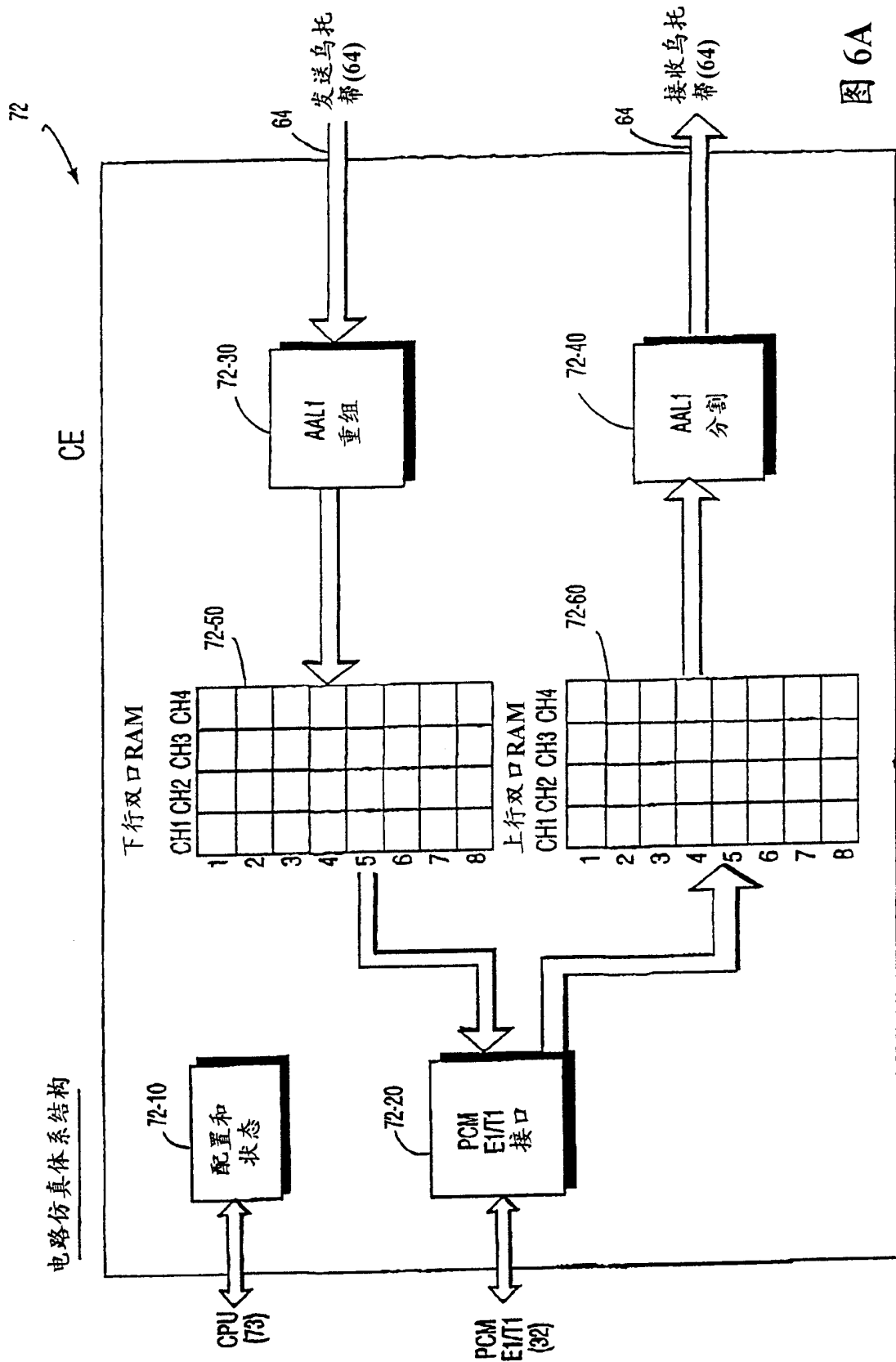


图 6A

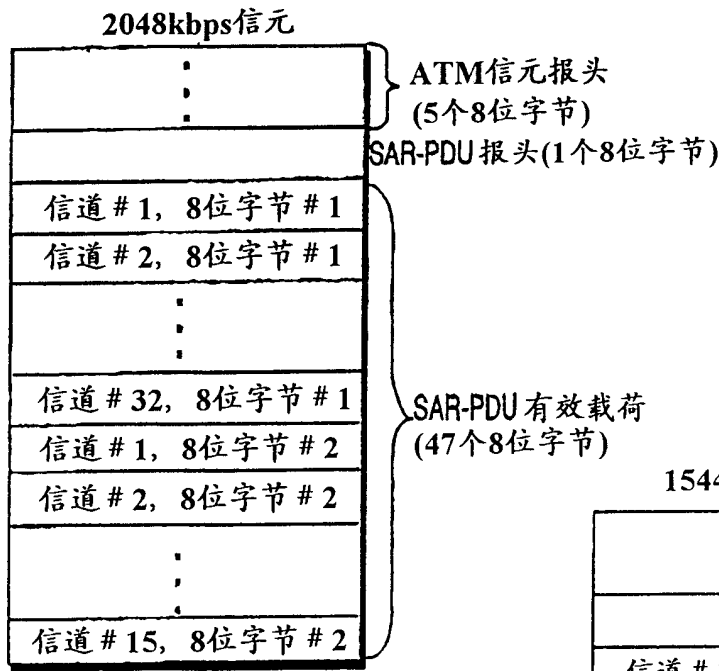


图 7A

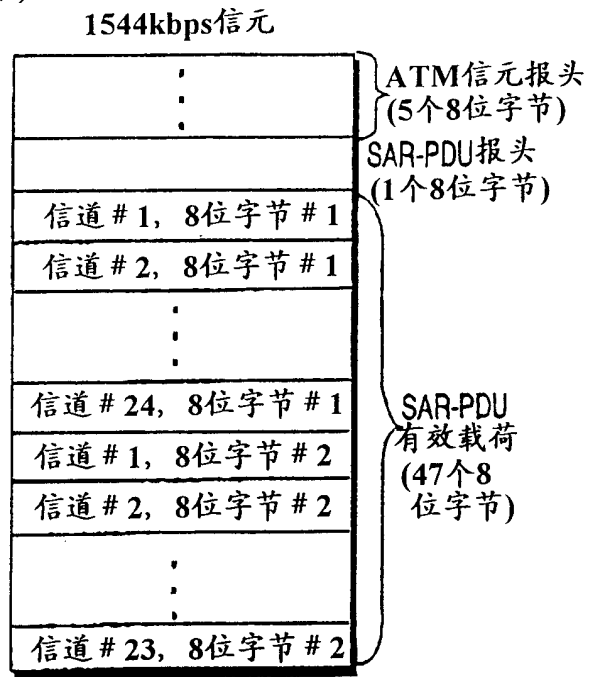


图 7B

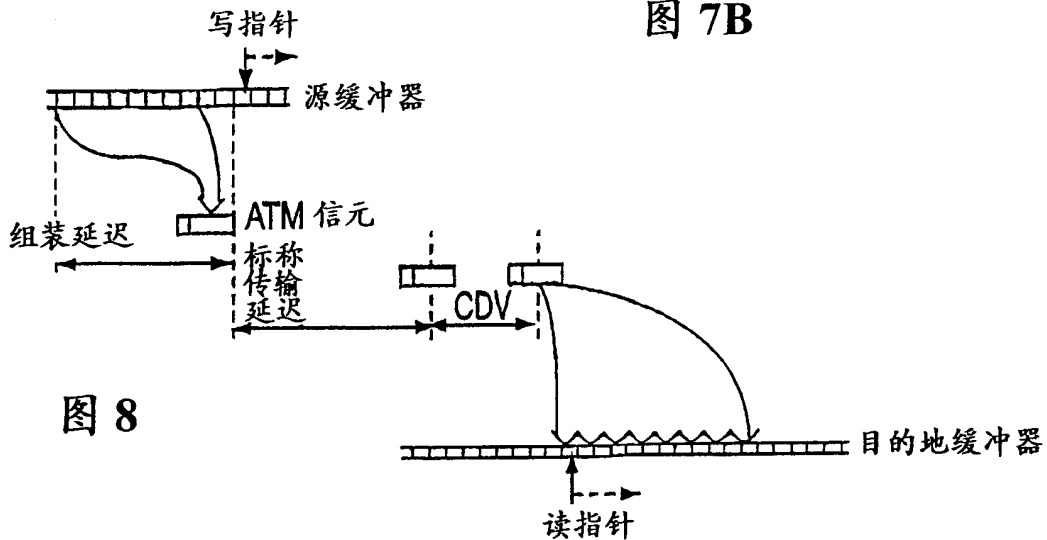


图 8

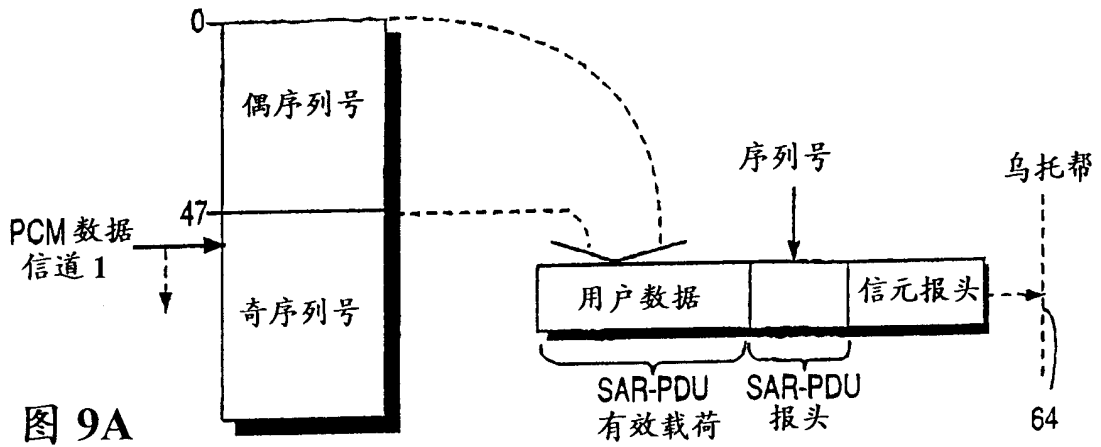


图 9A

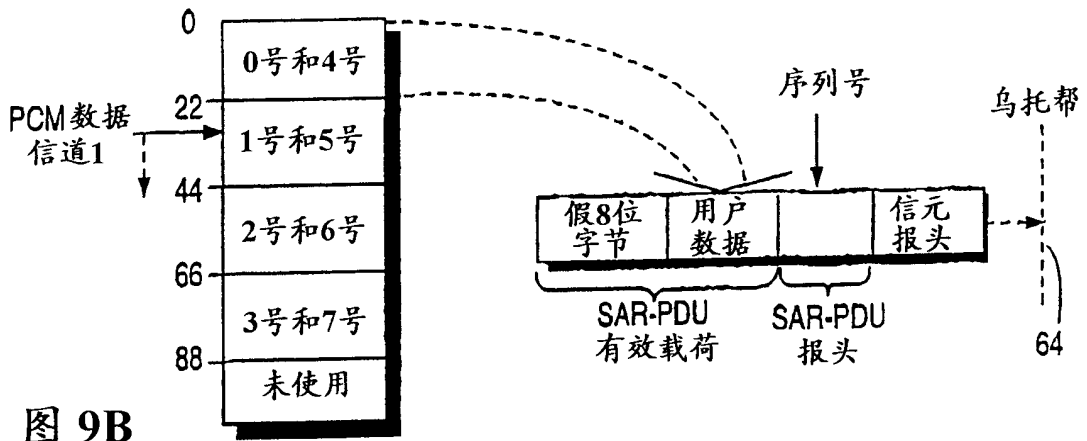


图 9B

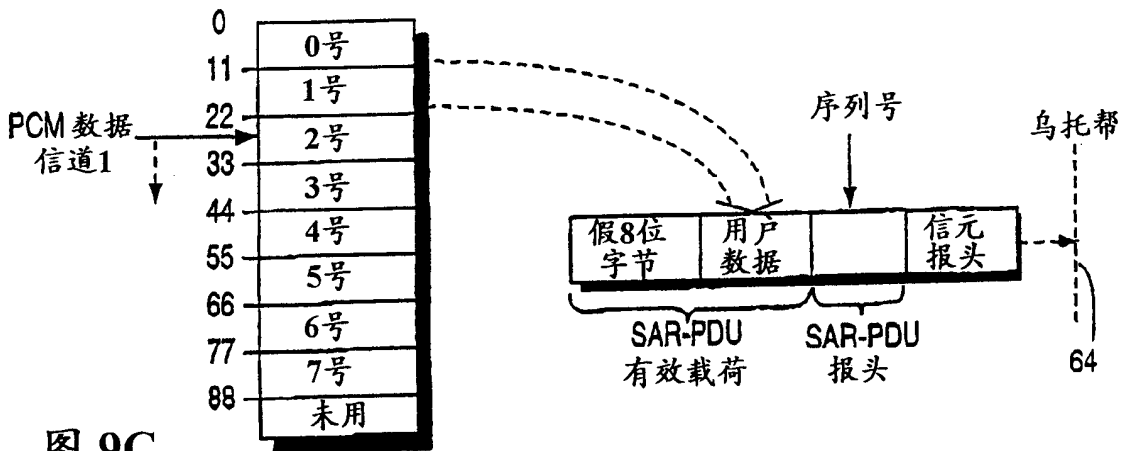


图 9C

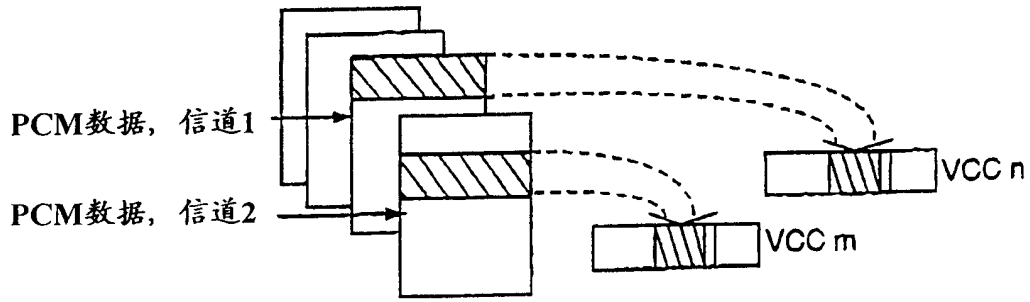


图 9D

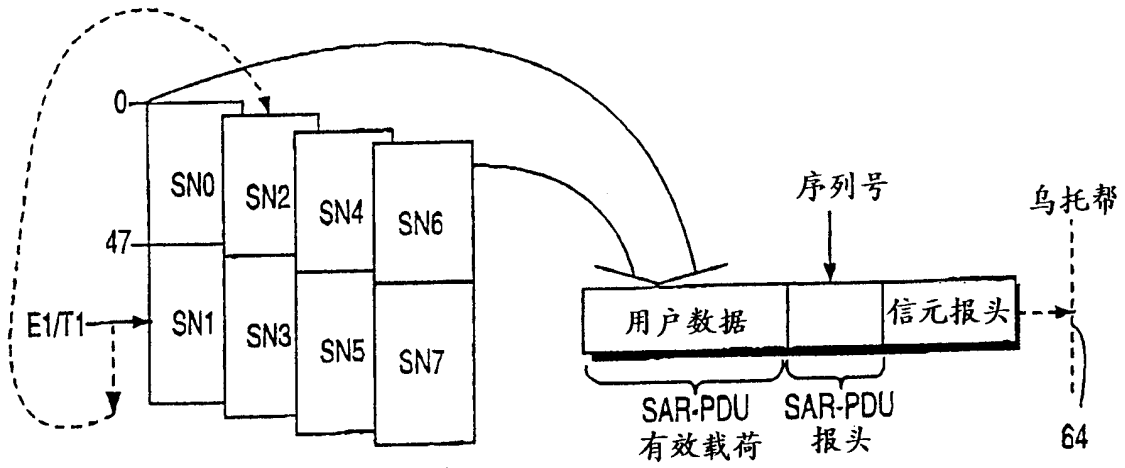


图 9E

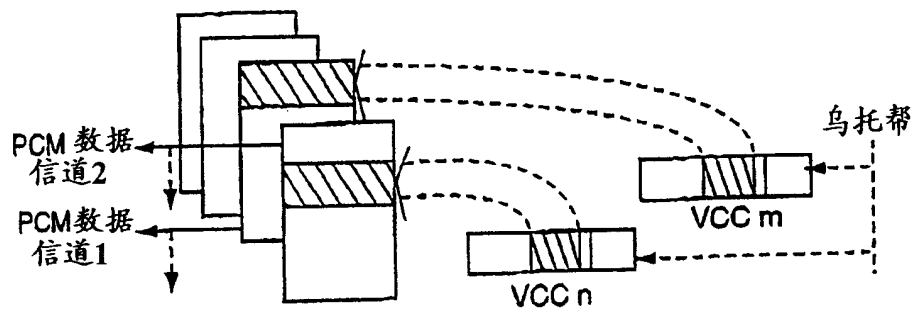


图 9F

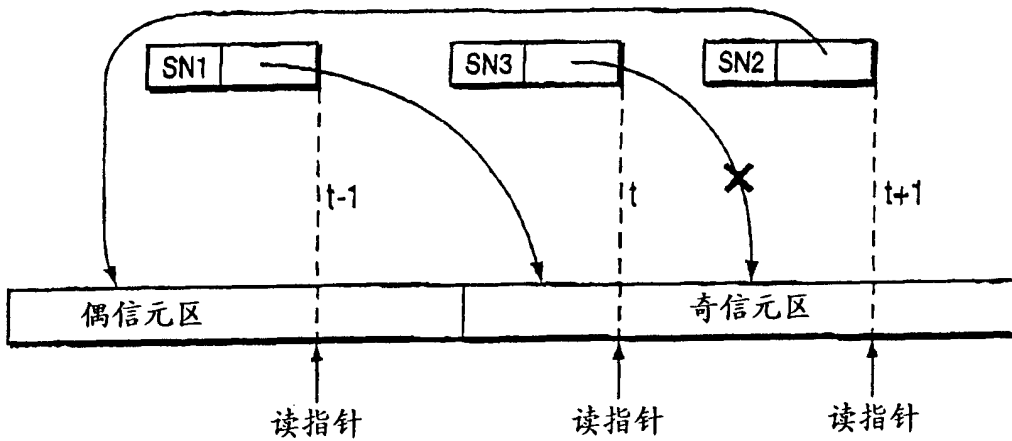


图 11

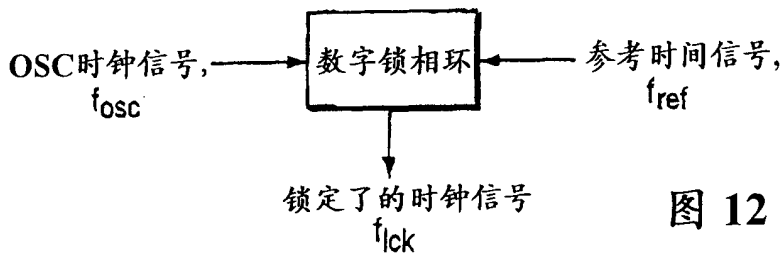


图 12

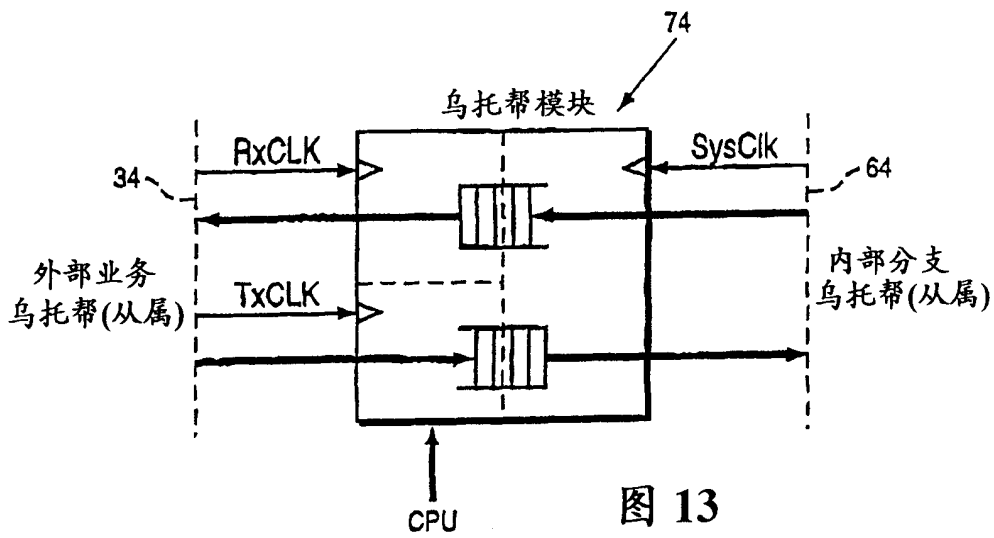


图 13

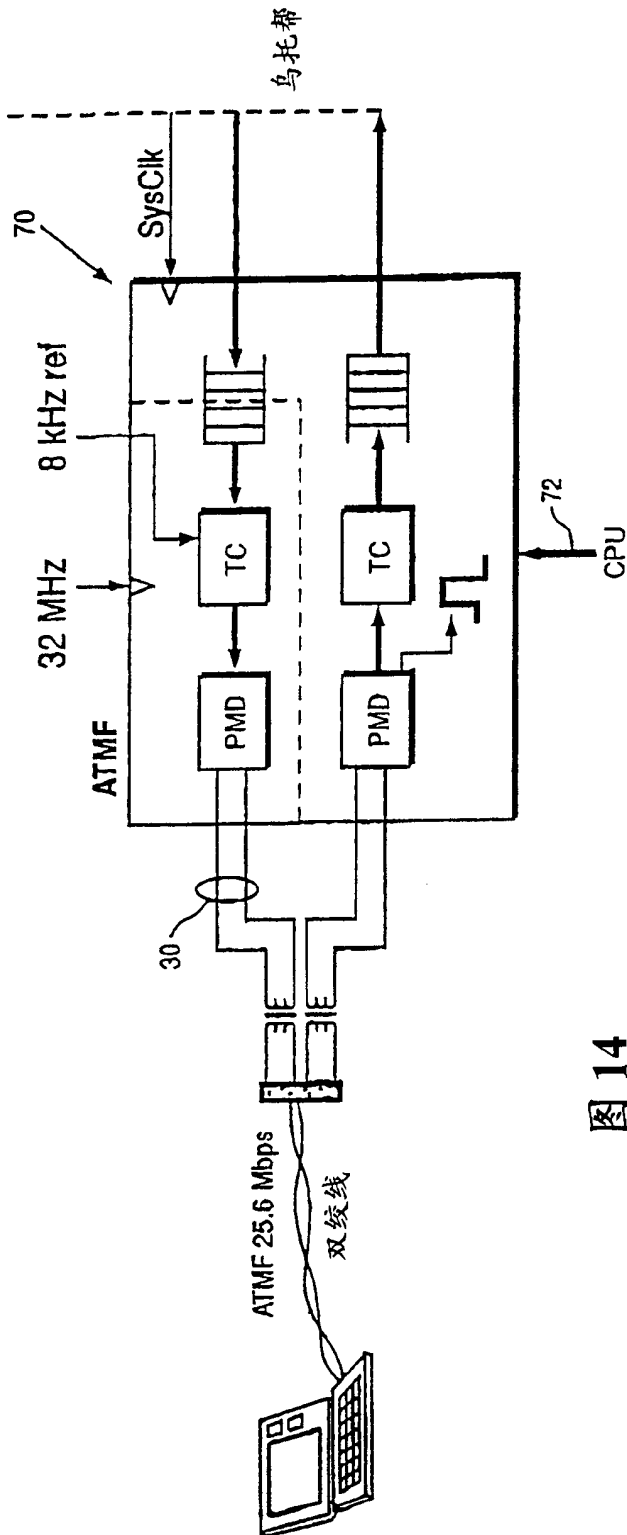


图 14

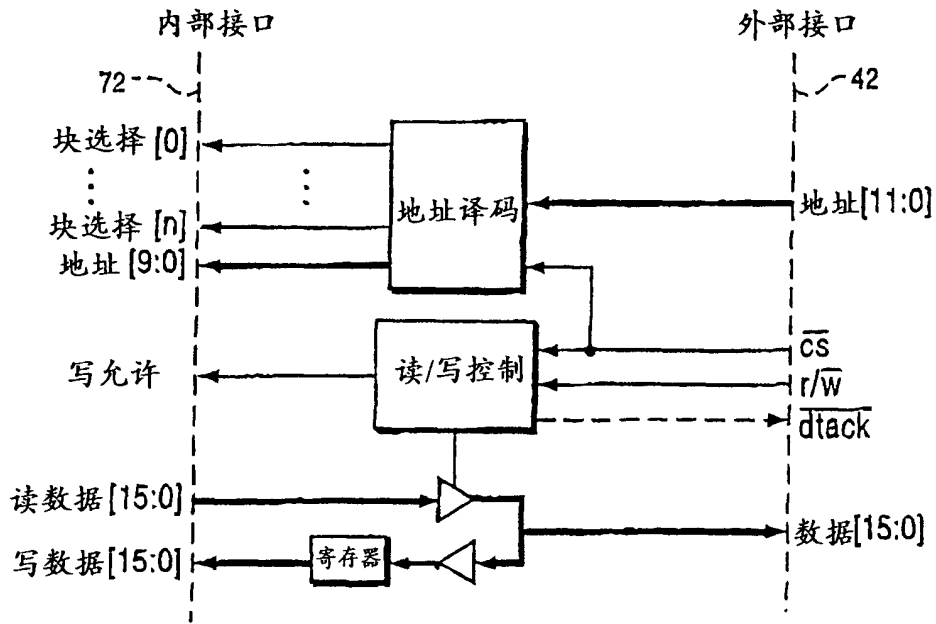


图 15

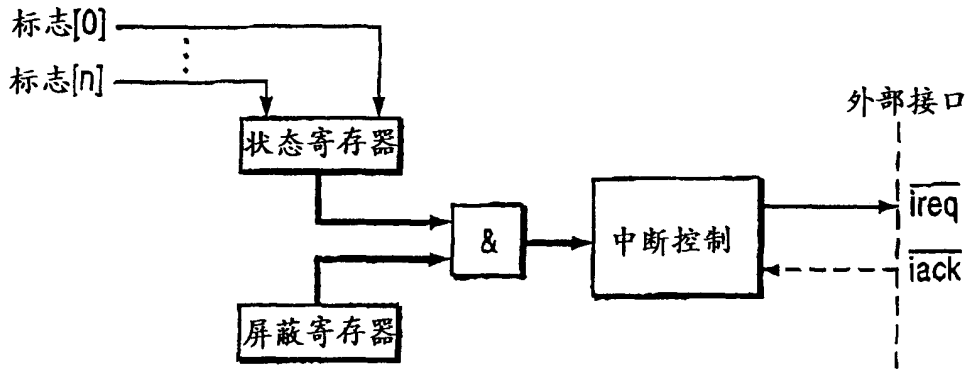


图 16

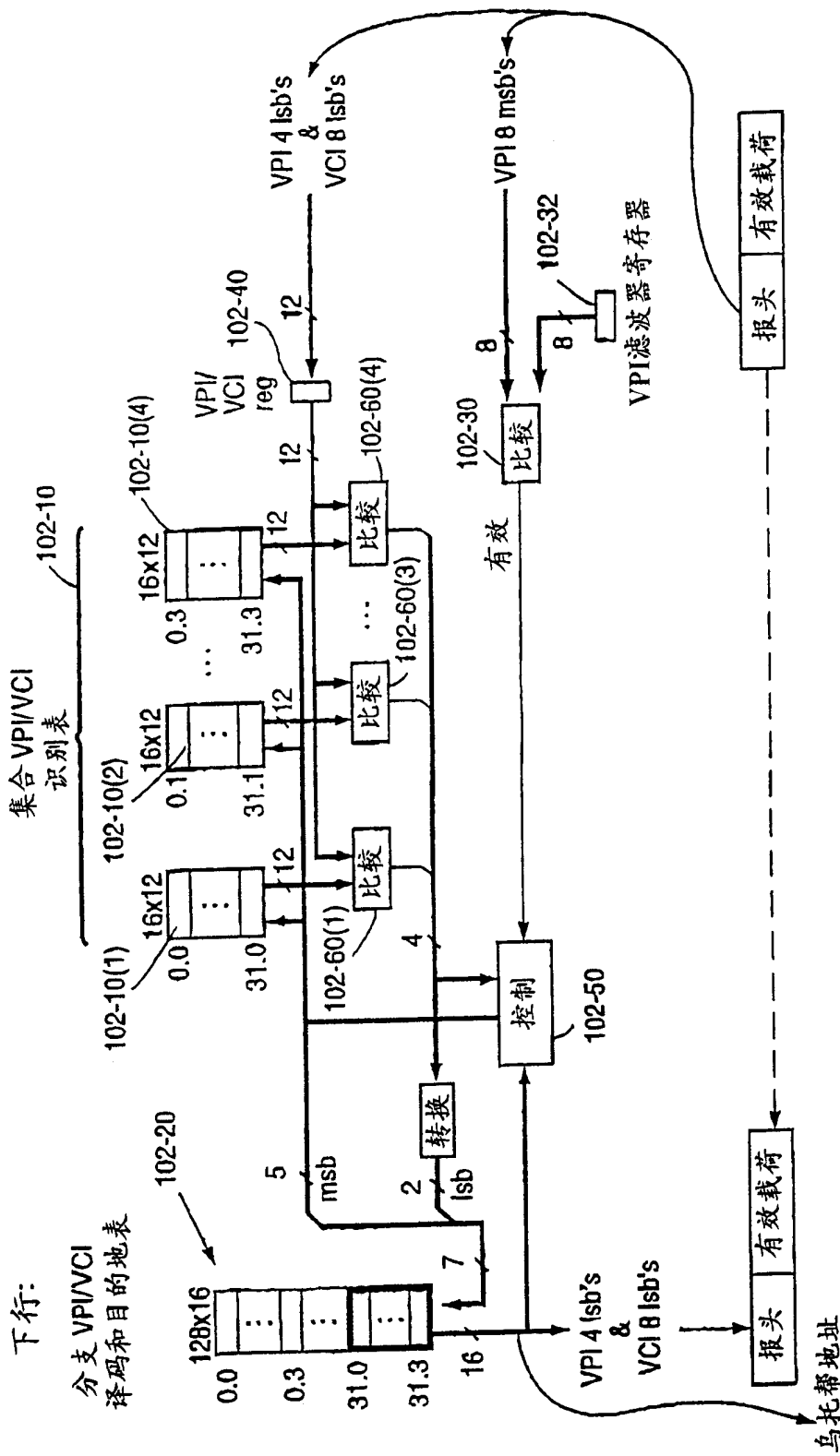


图 17A

